Note
Before using this information and the product it supports, read the information in "Notices" on page 983.

Product Information
This document applies to IBM Cognos Business Intelligence Version 10.2.1 and may also apply to subsequent releases.
Licensed Materials - Property of IBM
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
# Contents

**Introduction** ........................................ iv

**Chapter 1. What’s New in Cognos Report Studio?** ........................... 1

- New features in version 10.2.1 ........................................ 1
  - Improved IBM Cognos Active Report performance ....................... 1
  - Extensible visualizations in IBM Cognos Active Report ................. 1
  - Option to group or ungroup repeating cells in Excel 2007 reports .... 2
  - Crosstab values as percentages ................................ 2
  - Share sets between reports ........................................ 2
  - Improved text-based relational filters ................................ 2
  - Improved data container and query naming support .................... 3
  - Search parameter Ends with any of these keywords .................... 3
- Removed Features in Version 10.2.1 .................................. 3
  - IBM Cognos Statistics ............................................... 3
- New features in version 10.2.0 ........................................ 4
  - Updated world and Africa maps .................................... 4
  - Interactive repeater tables ........................................ 4
  - Improved IBM Cognos Active Report integration with IBM Cognos Workspace .... 4
  - Improved support for bidirectional languages ....................... 5
  - Macro tab in expression editor .................................... 5
  - Inherited table styles in lists and crosstabs .......................... 5
  - Enhanced crosstab headers ......................................... 6
  - Prompt API ......................................................... 6
  - Excel 2007 Data report output format ................................ 6
  - Global classes for accessible reports ................................ 6
- Changed features in version 10.2.0 .................................... 7
  - Support for inline prompts in reports viewed on mobile devices ...... 7
  - Cognos Business Insight and Cognos Business Insight Advanced are renamed .... 7
- Removed Features in Version 10.2.0 .................................. 7
  - Saving and opening reports on your computer .......................... 7
- New features in version 10.1.1 ........................................ 8
  - New active report controls ....................................... 8
  - Conversion of controls to other types of controls .................... 8
  - Improved user experience when authoring active reports ............ 8
  - Active report access code ........................................ 8
  - New startup options ............................................... 8
  - Keyboard shortcuts for accessibility ................................ 9
  - Updated maps ....................................................... 9
  - Custom groups ..................................................... 9
  - Larger worksheet sizes for Microsoft Excel 2007 report outputs .......... 10
  - Microsoft Excel 2007 supported as an external data source .......... 10
- Changed features in version 10.1.1 .................................... 10
  - Specify how to show no data contents ................................ 10
- What’s new in version 10.1.0 ........................................ 11
- New Features in Version 10.1.0 ...................................... 11
- Changed Features in Version 10.1.0 .................................. 19
- Removed Features in Version 10.1.0 .................................. 19

**Chapter 2. Getting started with Report Studio** ............................ 21

- Building IBM Cognos Business Intelligence Applications .................. 21
- Relational and dimensional reporting styles ................................ 22
- The User Interface .................................................... 25
- Work in design or structure view ...................................... 28
- Report Layout and Queries ........................................... 29
  - Layout ............................................................. 29

© Copyright IBM Corp. 2005, 2013
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries</td>
<td>30</td>
</tr>
<tr>
<td>Report Objects</td>
<td>31</td>
</tr>
<tr>
<td>Find Objects in a Report</td>
<td>32</td>
</tr>
<tr>
<td>Options</td>
<td>32</td>
</tr>
<tr>
<td>Web Browser Settings</td>
<td>36</td>
</tr>
<tr>
<td>The IBM Cognos Software Development Kit</td>
<td>37</td>
</tr>
<tr>
<td>Chapter 3. Creating a report</td>
<td>39</td>
</tr>
<tr>
<td>Specify the Data Package</td>
<td>39</td>
</tr>
<tr>
<td> Refresh the package</td>
<td>40</td>
</tr>
<tr>
<td>Choose a Basic Report Layout</td>
<td>40</td>
</tr>
<tr>
<td> Inserting a data container from the toolbox tab</td>
<td>41</td>
</tr>
<tr>
<td>Add Data</td>
<td>41</td>
</tr>
<tr>
<td> Data Source Icons</td>
<td>41</td>
</tr>
<tr>
<td> Insert a Single Data Item</td>
<td>42</td>
</tr>
<tr>
<td>Validate a Report</td>
<td>43</td>
</tr>
<tr>
<td>Save a Report</td>
<td>44</td>
</tr>
<tr>
<td>Run a Report</td>
<td>45</td>
</tr>
<tr>
<td> Specify Not to Render a Page If It Does Not Contain Data.</td>
<td>47</td>
</tr>
<tr>
<td> Controlling the Rows Per Page for Multiple Containers in HTML and PDF</td>
<td>48</td>
</tr>
<tr>
<td> Producing a Report in CSV Format</td>
<td>48</td>
</tr>
<tr>
<td> Producing a Report in Microsoft Excel Format</td>
<td>49</td>
</tr>
<tr>
<td> Producing a Report in XML Format</td>
<td>52</td>
</tr>
<tr>
<td> Set PDF Page Options</td>
<td>52</td>
</tr>
<tr>
<td> View Lineage Information for a Data Item</td>
<td>53</td>
</tr>
<tr>
<td> The IBM Cognos Business Intelligence Lineage Tool</td>
<td>54</td>
</tr>
<tr>
<td> Access the IBM WebSphere Business Glossary</td>
<td>55</td>
</tr>
<tr>
<td> Support for bidirectional languages</td>
<td>56</td>
</tr>
<tr>
<td> Reports for workspaces in Cognos Workspace</td>
<td>57</td>
</tr>
<tr>
<td> Prompts in workspaces in Cognos Workspace</td>
<td>58</td>
</tr>
<tr>
<td> Enable filters in workspaces in Cognos Workspace</td>
<td>59</td>
</tr>
<tr>
<td> Creating Reports for IBM Cognos for Microsoft Office</td>
<td>59</td>
</tr>
<tr>
<td> Creating Reports for Mobile Devices</td>
<td>61</td>
</tr>
<tr>
<td> Creating location-aware reports</td>
<td>61</td>
</tr>
<tr>
<td>Chapter 4. Lists</td>
<td>63</td>
</tr>
<tr>
<td>Set List Properties</td>
<td>63</td>
</tr>
<tr>
<td>Hide Columns in List Reports</td>
<td>64</td>
</tr>
<tr>
<td>Using Repeaters</td>
<td>64</td>
</tr>
<tr>
<td> Convert a List into a Repeater</td>
<td>65</td>
</tr>
<tr>
<td> Example - Create Mailing Labels</td>
<td>65</td>
</tr>
<tr>
<td>Chapter 5. Crosstabs</td>
<td>67</td>
</tr>
<tr>
<td>Crosstab Nodes and Crosstab Node Members</td>
<td>67</td>
</tr>
<tr>
<td>Set Crosstab Properties</td>
<td>69</td>
</tr>
<tr>
<td>Create a Single-Edge Crosstab</td>
<td>70</td>
</tr>
<tr>
<td>Create a Nested Crosstab</td>
<td>71</td>
</tr>
<tr>
<td>Creating headers automatically for sets in crosstabs</td>
<td>72</td>
</tr>
<tr>
<td>Showing crosstab values as percentages</td>
<td>72</td>
</tr>
<tr>
<td>Specify the Default Measure</td>
<td>73</td>
</tr>
<tr>
<td>Swap Columns and Rows</td>
<td>74</td>
</tr>
<tr>
<td>Indent Data</td>
<td>74</td>
</tr>
<tr>
<td>Change a List into a Crosstab</td>
<td>75</td>
</tr>
<tr>
<td>Example - Add Aggregate Data to a Crosstab</td>
<td>75</td>
</tr>
<tr>
<td>Example - Create a Discontinuous Crosstab</td>
<td>76</td>
</tr>
<tr>
<td>Chapter 6. Charts</td>
<td>79</td>
</tr>
<tr>
<td>Current Default Charts and Legacy Charts</td>
<td>79</td>
</tr>
<tr>
<td>Creating Charts</td>
<td>79</td>
</tr>
<tr>
<td> Chart Objects</td>
<td>80</td>
</tr>
</tbody>
</table>
Example - Create an Active Report Showing Revenue Data for Core Products .................................................. 199
  Create the active report .................................................. 199
  Add data to the data drop-down list controls .................. 200
  Add data to the list .................................................. 201
  Add interactivity to the report ........................................ 202
  Specifying default values for the data drop-down list controls .... 202
  Revenue data for core products report output ............ 202
Example - Sales analysis active report .................................................. 203
  Creating a new active report ........................................ 203
  Adding data to the check box group control .................. 204
  Adding a default card to the deck and data to the data discrete values slider .... 204
  Defining a connection between the check box group and the deck .......... 205
  Adding data to the list object ....................................... 205
  Defining a connection between the slider and the list object .......... 206
  Adding a title to the list ........................................... 206
  Adding data to the data deck and chart .......................... 207
  Defining a connection between the chart and the data deck .......... 207
  Adding a title to the chart ......................................... 208
  Sales analysis report output ........................................ 208
Example - Network diagram visualization .......................................... 209
  Adding a new visualization in IBM Cognos Active Report ........ 209
  Adding data to the nodes data container ....................... 210
  Adding data to the links data container ....................... 210
  Defining a connection between the data drop-down list and the visualization object ........................................ 210
  Network diagram visualization output .......................... 211

Chapter 9. Relational Reporting Style ................................................. 213
Add Relational Data to a Report ........................................ 213
Group Relational Data .................................................. 213
  Perform Advanced Grouping ......................................... 214
  Creating relational custom groups ................................ 215
  Set the Group Span for a Column ................................ 216
  Divide Data into Sections ........................................... 218
Summarizing Data Relationally .............................................. 219
  Add a Simple Summary ............................................... 220
  Set the Auto Group & Summarize Property ...................... 221
  Specify Detail or Summary Aggregation in the Model .......... 222
  Specify Aggregation Properties for a Data Item .............. 222
  Use Summary Functions in Data Item Expressions ............ 222
  Summary Functions .................................................. 224
Focusing Relational Data .................................................. 228
  Create a Detail or Summary Filter ................................ 229
  Edit or remove a filter ............................................. 233
  Browse or Search the Values of a Data Item ................ 233
  Ignoring the Time Component in Date Columns ................ 234
Sorting Relational Data .................................................... 235
  Perform Advanced Layout Sorting ................................ 235
Working with Relational Queries .......................................... 236
  Specify a List of Data Items for an Object .................... 237
  Relate a Query to a Layout ........................................ 237
  Connecting Queries Between Different Data Sources .......... 237
  Add a Query to a Relational Report ............................. 238
  Create a Union Query ............................................... 239
  Create a Join Relationship ........................................ 242
  Reference a Package Item in a Child Query ................... 243
  Create a Master Detail Relationship ............................. 244
  Working with Queries in SQL ...................................... 246
Using Relational Calculations .............................................. 248
  Create a Simple Calculation ...................................... 249
  Insert a Query Calculation ....................................... 250
  Create a Layout Calculation ...................................... 252
### Chapter 10. Dimensional Reporting Style

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Dimensional Data to a Report</td>
<td>253</td>
</tr>
<tr>
<td>Customize the Source Tree</td>
<td>255</td>
</tr>
<tr>
<td>Insert a Member</td>
<td>255</td>
</tr>
<tr>
<td>Search for a Member</td>
<td>256</td>
</tr>
<tr>
<td>Nest Members</td>
<td>257</td>
</tr>
<tr>
<td>Insert a Hierarchy</td>
<td>257</td>
</tr>
<tr>
<td>Insert a Member Property</td>
<td>258</td>
</tr>
<tr>
<td>Create a Set of Members</td>
<td>258</td>
</tr>
<tr>
<td>Sharing sets between reports</td>
<td>259</td>
</tr>
<tr>
<td>Edit a Set</td>
<td>261</td>
</tr>
<tr>
<td>Extended Data Items</td>
<td>263</td>
</tr>
<tr>
<td>Tips for Working with Ragged or Unbalanced Hierarchies</td>
<td>264</td>
</tr>
<tr>
<td>Creating dimensional custom groups</td>
<td>265</td>
</tr>
<tr>
<td>Summarizing Data Dimensionally</td>
<td>266</td>
</tr>
<tr>
<td>Add a Simple Summary</td>
<td>268</td>
</tr>
<tr>
<td>Summarizing Values in Crosstabs</td>
<td>269</td>
</tr>
<tr>
<td>Summarizing Member Sets</td>
<td>271</td>
</tr>
<tr>
<td>Rolling and Moving Averages</td>
<td>272</td>
</tr>
<tr>
<td>Show Data for Specific Time Periods</td>
<td>273</td>
</tr>
<tr>
<td>Summary Functions</td>
<td>273</td>
</tr>
<tr>
<td>Focusing Dimensioned Data</td>
<td>278</td>
</tr>
<tr>
<td>Create a Context Filter</td>
<td>279</td>
</tr>
<tr>
<td>Sorting Dimensioned Data</td>
<td>281</td>
</tr>
<tr>
<td>Sort Sets by Label</td>
<td>282</td>
</tr>
<tr>
<td>Sort Sets in the Opposite Axis by Value</td>
<td>282</td>
</tr>
<tr>
<td>Perform Advanced Set Sorting</td>
<td>283</td>
</tr>
<tr>
<td>Working with Dimensioned Queries</td>
<td>283</td>
</tr>
<tr>
<td>Specify a List of Data Items for an Object</td>
<td>284</td>
</tr>
<tr>
<td>Relate a Query to a Layout</td>
<td>284</td>
</tr>
<tr>
<td>Using Dimensional Data Sources with Queries</td>
<td>284</td>
</tr>
<tr>
<td>Connecting Queries Between Different Data Sources</td>
<td>285</td>
</tr>
<tr>
<td>Multiple-fact Queries</td>
<td>285</td>
</tr>
<tr>
<td>Add a Query to a Dimensional Report</td>
<td>289</td>
</tr>
<tr>
<td>Add Dimension Information to a Query</td>
<td>290</td>
</tr>
<tr>
<td>Create a Master Detail Relationship</td>
<td>292</td>
</tr>
<tr>
<td>Working with Queries in SQL or MDX</td>
<td>295</td>
</tr>
<tr>
<td>Example - Create a Dynamic Report That Shows Period-to-date Data</td>
<td>300</td>
</tr>
<tr>
<td>Using Dimensional Calculations</td>
<td>302</td>
</tr>
<tr>
<td>Create a Simple Member Calculation</td>
<td>302</td>
</tr>
<tr>
<td>Insert a Query Calculation</td>
<td>303</td>
</tr>
<tr>
<td>Create a Layout Calculation</td>
<td>305</td>
</tr>
<tr>
<td>Create an Intersection (Tuple)</td>
<td>306</td>
</tr>
<tr>
<td>Assigning a Hierarchy or Dimension</td>
<td>306</td>
</tr>
<tr>
<td>Drilling Up and Drilling Down in Dimensional Reports</td>
<td>307</td>
</tr>
<tr>
<td>Member Sets</td>
<td>307</td>
</tr>
<tr>
<td>Create a Drill-up and Drill-down Report</td>
<td>308</td>
</tr>
<tr>
<td>Example - Create a Dashboard Report</td>
<td>310</td>
</tr>
</tbody>
</table>

### Chapter 11. Adding Prompts to Filter Data

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the Build Prompt Page Tool</td>
<td>314</td>
</tr>
<tr>
<td>Build Your Own Prompt and Prompt Page</td>
<td>315</td>
</tr>
<tr>
<td>Example - Create a Report Showing Products Shipped for a Specific Time Interval</td>
<td>318</td>
</tr>
<tr>
<td>Create a Parameter to Produce a Prompt</td>
<td>319</td>
</tr>
<tr>
<td>Create a Prompt Directly in a Report Page</td>
<td>320</td>
</tr>
<tr>
<td>Modifying Prompts</td>
<td>321</td>
</tr>
<tr>
<td>Change the Prompt Control Interface</td>
<td>321</td>
</tr>
<tr>
<td>Specify That a Prompt Requires User Input</td>
<td>322</td>
</tr>
<tr>
<td>Allow Users to Select Multiple Values in a Prompt</td>
<td>322</td>
</tr>
<tr>
<td>Show or Hide Prompt Status</td>
<td>323</td>
</tr>
<tr>
<td>Specify a Default Selection for a Prompt</td>
<td>323</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Undocumented and Unsupported Features</td>
<td>464</td>
</tr>
<tr>
<td>No Sort Order</td>
<td>464</td>
</tr>
<tr>
<td>Layout Errors</td>
<td>464</td>
</tr>
<tr>
<td>Report Format</td>
<td>464</td>
</tr>
<tr>
<td>Customized Style Sheets</td>
<td>464</td>
</tr>
<tr>
<td>PDF Reports</td>
<td>464</td>
</tr>
<tr>
<td>IF-THEN-ELSE Statements</td>
<td>464</td>
</tr>
<tr>
<td>Solve Order</td>
<td>465</td>
</tr>
<tr>
<td>Chart Behavior</td>
<td>465</td>
</tr>
<tr>
<td>Database Only Processing of Queries</td>
<td>465</td>
</tr>
<tr>
<td>Changes in the Behavior of Functions Between ReportNet 1.1 and IBM Cognos BI</td>
<td>465</td>
</tr>
</tbody>
</table>

| Appendix A. Accessibility Features                                    | 467  |
| Accessibility features in IBM Cognos Report Studio                    | 467  |
| Keyboard Shortcuts                                                    | 467  |
| Considerations to Improve Report Accessibility                         | 468  |
| Enable Accessible Report Outputs                                      | 469  |
| Alternate text and summary text                                       | 469  |
| Designate Cells Headers in Tables                                     | 471  |
| Headings and emphasis                                                  | 471  |
| Prompt control labels                                                  | 472  |
| Example - Conditionally Show a List Below a Chart for an Accessible Report | 473  |
| IBM and accessibility                                                  | 475  |

| Appendix B. Troubleshooting                                           | 477  |
| Problems Creating Reports                                             | 477  |
| Division by Zero Operation Appears Differently in Lists and Crosstabs | 477  |
| Application Error Appears When Upgrading a Report                     | 477  |
| Nested List Report Containing a Data Item That is Grouped More Than Once Does Not Run After Upgrade | 477  |
| Background Color in Template Does not Appear                          | 478  |
| Subtotals in Grouped Lists                                            | 478  |
| Chart Labels Overwrite One Another                                    | 478  |
| Chart Shows Only Every Second Label                                   | 478  |
| Chart Gradient Backgrounds Appear Gray in Internet Explorer           | 479  |
| Metadata Change in Oracle Essbase Not Reflected in Reports and in the Studios | 479  |
| Relationships Not Maintained in a Report With Overlapping Set Levels  | 479  |
| Summaries in Query Calculations Include Nulls with SAP BW Data Sources | 480  |
| Creating Sections on Reports That Access SAP BW Data Sources          | 480  |
| Error Characters (--) Appear in Reports                               | 480  |
| Columns, Rows, or Data Disappear With SSAS 2005 Cubes                 | 481  |
| Function Unreliable with Sets                                         | 482  |
| Searching for values might return unexpected results                  | 482  |
| Report Differences Between TM1 Executive Viewer and IBM Cognos Business Intelligence with TM1 Data Sources             | 482  |
| Order of Metadata Tree Diffs for TM1 Data Sources                      | 483  |
| MSR-PD-0012 error when importing external data                        | 483  |
| MSR-PD-0013 error when importing external data                        | 483  |
| Problems Calculating Data                                             | 483  |
| Unexpected Summary Values in Nested Sets                              | 483  |
| Null Results for Calculations Using SAP BW Data Sources               | 484  |
| Incorrect Results with IBM Cognos PowerCubes and Time Measures         | 484  |
| Incorrect Results in Summaries When Using OLAP Data Sources           | 485  |
| Problems Filtering Data                                               | 486  |
| Unexplained Discrepancies in Number Calculations                      | 486  |
| HRESULT= DB_E_CANTCONVERTVALUE Error When Filtering on a _make_timestamp Column | 488  |
| Problems Running Reports                                              | 488  |
| Report Runs Slowly                                                     | 488  |
| Summaries in a report do not correspond to the visible members         | 490  |
| Cannot Find the Database in the Content Store (Error QE-DEF-0288)      | 493  |
| Parse Errors When Opening or Running an Upgraded Report               | 493  |
Overflow Error Occurs When a Value in a Crosstab Is More Than 19 Characters.

The ORA-00907 Error Appears When Running a Report.

A Report or Analysis Does Not Run Because of Missing Items.

Cannot View Burst Report.

A report upgraded from ReportNet does not retain its original look.

Measure Format Disappears in SSAS 2005.

Drill-through Links are Not Active in the Safari Browser.

Data Does Not Appear in a Target Report or the Wrong Data Appears.

A Running Total in Grouped Reports Gives Unexpected Results.

PCA-ERR-0057 Recursive Evaluation Error.

Arithmetic Overflow Error When Running a Report in PDF Format.

RQP-DEF-0177 An error occurred while performing operation ‘sqlPrepareWithOptions’ status=’-69’

UDA-SQL-0043 Error.

Unable to View Active Reports in Mozilla Firefox.

Problems with viewing saved Active Report output in Mozilla Firefox 8 or later versions.

Problems with large SAP BW queries.

Master Detail or Burst Reports with Charts or Crosstabs May Result in Denial of Service.

PCA-ERR-0087 error when you run a large report.

Differences in the appearance of charts that are run in different formats or on different operating systems.

Out-of-memory errors with reports that are run in interactive HTML format.

Charts in PDF Output Show Unexpected Results.

Result set of a multi-fact query contains blanks.

Cognos Statistics object is not displayed in a report.

Problems When Drilling Through.

Cannot Drill Through Between PowerCubes Because MUNs Do Not Match.

Unexpected or Empty Results When Drilling Through.

Cannot Drill Through From a Relational Source to a Cube.

Calculations Do Not Appear in the Target Report.

Nested Crosstab Only Filters on Some Items.

Data is Not Filtered in the Target Report After Drill-Through.

Drill-through links in active reports do not work.

Appendix C. Sample Reports and Packages.

The Sample Outdoors Company.

The Sample Outdoors Group of Companies.

Employees.

Sales and marketing.

Sample Outdoors database, models, and packages.

Samples in the Sample Outdoors Sales (cube) package.

Consecutive Periods Comparison.

Samples in the Sales and Marketing (Cube) Package.

Actual vs. Planned Revenue.

Historical Revenue.

Revenue by Date Range.

Revenue by Product Brand (2011).

Same Month Prior Year.

Selected Retailer Country or Region.

Top Retailers by Country or Region.

Samples in the GO Data Warehouse (analysis) Package.

Budget vs. Actual.

Core products results.

Customer Returns and Satisfaction.

Employee Satisfaction 2012.

Employee Training by Year.

Eyewear Revenue by Brand and Size.

Global Bonus Report.

GO Balance Sheet as at Dec 31 2012.

Sample Outdoors Company Balance Sheet as at Dec 31 2012.

Manager Profile.

New order methods.

Planned Headcount.
Appendix D. Limitations when using dimensional data sources. ............................. 531
Running a Report Against a Dimensional Data Source ........................................... 531
Limitations When Using Set Expressions in List Reports ........................................ 532
Limitations When Using Clauses in Summary Functions ......................................... 532
Limited Support for Relational Functions When Used with OLAP Data Sources ........... 533
Limitations When Summarizing Measures in DMR Data Sources .............................. 534
Limitations When Filtering Dimensional Data Sources ........................................... 534
Error Characters (--) When Filtering ......................................................................... 535
Limitations When Specifying the Scope of Summary Filters with Dimensional Data Sources .................................................. 535
Limitations When Filtering Data Using an SAP BW Data Source ...................... 536
Considerations when Creating Calculations .............................................................. 537
Calculation Solve Order ............................................................................................ 537
Length of Expressions .............................................................................................. 537
Quality of Service Indicators .................................................................................... 537
Using Quotation Marks in Literal Strings ................................................................ 538
Limitations of Calculations ....................................................................................... 538
Creating Expressions Using SAP BW Data Sources ............................................... 538
Using Microsoft Excel Functions with SSAS 2005 Data Sources ...................... 539
Concatenating Strings ............................................................................................... 539
Intersecting Calculations in Crosstabs and Charts .................................................... 539
Null (Missing) Values in Calculations and Summaries ............................................. 542
Dimensional Coercion Rules .................................................................................... 544
Recommendation - Use Member Unique Name (MUN) Aliases ............................... 547
Limitations When Formatting Empty Cells in SAP BW Data Sources ................. 548

Appendix E. Limitations When Producing Reports in Microsoft Excel Format. ..................... 549
Unable to Load Images from the IBM Cognos Business Intelligence Content Store in a Report ................................................. 549
Blank Worksheet Appears ....................................................................................... 549
Warning Message Appears When Excel Opens an IBM Cognos Business Intelligence Report .................................................. 549
Spreadsheet Content Not Saved for Reports Saved in XLS Format ...................... 549
Unable to Load Excel Report in Netscape 7.01 ....................................................... 550
Unable to Nest Labels in Charts ............................................................................. 550
Data Series Are Truncated ....................................................................................... 550
Colors Different From Those in HTML or PDF ...................................................... 550
Repeating Pie Charts ............................................................................................... 550
Unable to Skip Discrete Axis Labels ....................................................................... 550
Unsupported IBM Cognos BI Formatting ................................................................ 550
Cells Contain Series of # ....................................................................................... 551
Table and Column Widths ...................................................................................... 551
Secure Socket Layer (SSL) Is Not Supported in Some Excel Formats and Versions .................................................. 551
Number Formats Become Currency Formats in Japanese Excel ............................ 552
Reports Show Data in Wrong Columns ................................................................. 552
Unable to Access Reports on Remote Servers ....................................................... 552
Drill-through Reports Are Not Supported in Excel .............................................. 552
Map Reports Are Not Supported in Excel ............................................................. 552
Unsupported Excel Formatting ................................................................................ 552
Hyperlink Buttons Are Not Supported in Excel ..................................................... 553
Unable to View Reports in Excel Format Sent as Email Attachments .................... 553
Unsupported Chart Properties in Excel ................................................................. 553
Unsupported Chart Types in Excel ......................................................................... 553
Cell Height and Width Are Incorrect ...................................................................... 554

Appendix F. Using the expression editor. ................................................................ 555
Operators .................................................................................................................. 555
ascii .................................................. 640
chr ..................................................... 640
decode .................................................. 641
initcap .................................................. 641
instr .................................................... 641
lpad ..................................................... 641
ltrim ..................................................... 641
months_between ........................................ 641
next_day .................................................. 641
nvl ....................................................... 642
round ..................................................... 642
rpad ..................................................... 642
rtrim ..................................................... 642
substr .................................................... 643
{current_db} ............................................. 643
{current_user} .......................................... 643
{session_user} .......................................... 643
to_char .................................................. 643
to_date ................................................... 643
to_number ............................................... 643
translate ............................................... 643
date_trunc .............................................. 644
trunc ..................................................... 644
version ................................................... 644
Oracle ................................................... 644
Oracle Math .............................................. 644
Oracle Trigonometry ..................................... 644
add_months ............................................. 646
ascii ..................................................... 646
ceil ...................................................... 646
char_length ............................................. 646
chr ....................................................... 646
concat .................................................... 646
decode .................................................... 647
dump ....................................................... 647
greatest .................................................. 647
initcap ................................................... 647
instr ..................................................... 647
Instrb .................................................... 647
least ...................................................... 648
length ...................................................... 648
lengthb ................................................... 648
lpad ..................................................... 648
ltrim ..................................................... 648
months_between ........................................ 648
new_time .................................................. 649
next_day .................................................. 649
nls_initcap .............................................. 649
nls_lower ............................................... 649
nls_upper ............................................... 649
nvl ....................................................... 649
replace .................................................... 650
round ..................................................... 650
rpad ..................................................... 650
rtrim ..................................................... 650
sign ....................................................... 651
soundex ................................................... 651
substr .................................................... 651
Substrb .................................................... 651
{sysdate} .................................................. 652
to_char .................................................. 652

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>to_date</td>
<td>652</td>
</tr>
<tr>
<td>to_number</td>
<td>652</td>
</tr>
<tr>
<td>translate</td>
<td>652</td>
</tr>
<tr>
<td>trunc</td>
<td>652</td>
</tr>
<tr>
<td>trunc</td>
<td>653</td>
</tr>
<tr>
<td>{user}</td>
<td>653</td>
</tr>
<tr>
<td>vsizes</td>
<td>653</td>
</tr>
<tr>
<td>Paraccel</td>
<td>653</td>
</tr>
<tr>
<td>Paraccel String</td>
<td>653</td>
</tr>
<tr>
<td>Paraccel Math</td>
<td>654</td>
</tr>
<tr>
<td>current_database</td>
<td>654</td>
</tr>
<tr>
<td>current_schema</td>
<td>655</td>
</tr>
<tr>
<td>{current_user}</td>
<td>655</td>
</tr>
<tr>
<td>{session_user}</td>
<td>655</td>
</tr>
<tr>
<td>translate</td>
<td>655</td>
</tr>
<tr>
<td>version</td>
<td>655</td>
</tr>
<tr>
<td>Postgres</td>
<td>655</td>
</tr>
<tr>
<td>Postgres String</td>
<td>655</td>
</tr>
<tr>
<td>Postgres Math</td>
<td>657</td>
</tr>
<tr>
<td>Postgres Trigonometry</td>
<td>658</td>
</tr>
<tr>
<td>ascii</td>
<td>659</td>
</tr>
<tr>
<td>chr</td>
<td>659</td>
</tr>
<tr>
<td>{current_db}</td>
<td>660</td>
</tr>
<tr>
<td>{current_catalog}</td>
<td>660</td>
</tr>
<tr>
<td>{current_schema}</td>
<td>660</td>
</tr>
<tr>
<td>{current_user}</td>
<td>660</td>
</tr>
<tr>
<td>{session_user}</td>
<td>660</td>
</tr>
<tr>
<td>translate</td>
<td>660</td>
</tr>
<tr>
<td>date_trunc</td>
<td>660</td>
</tr>
<tr>
<td>version</td>
<td>660</td>
</tr>
<tr>
<td>Red Brick</td>
<td>660</td>
</tr>
<tr>
<td>ceil</td>
<td>660</td>
</tr>
<tr>
<td>concat</td>
<td>661</td>
</tr>
<tr>
<td>{current_user}</td>
<td>661</td>
</tr>
<tr>
<td>date</td>
<td>661</td>
</tr>
<tr>
<td>dateadd</td>
<td>661</td>
</tr>
<tr>
<td>datediff</td>
<td>661</td>
</tr>
<tr>
<td>datename</td>
<td>661</td>
</tr>
<tr>
<td>dec</td>
<td>662</td>
</tr>
<tr>
<td>decimal</td>
<td>662</td>
</tr>
<tr>
<td>decode</td>
<td>662</td>
</tr>
<tr>
<td>float</td>
<td>662</td>
</tr>
<tr>
<td>ifnull</td>
<td>662</td>
</tr>
<tr>
<td>int</td>
<td>662</td>
</tr>
<tr>
<td>integer</td>
<td>662</td>
</tr>
<tr>
<td>length</td>
<td>662</td>
</tr>
<tr>
<td>lengthb</td>
<td>663</td>
</tr>
<tr>
<td>ltrim</td>
<td>663</td>
</tr>
<tr>
<td>nullif</td>
<td>663</td>
</tr>
<tr>
<td>positionb</td>
<td>663</td>
</tr>
<tr>
<td>real</td>
<td>664</td>
</tr>
<tr>
<td>round</td>
<td>664</td>
</tr>
<tr>
<td>rtrim</td>
<td>664</td>
</tr>
<tr>
<td>sign</td>
<td>664</td>
</tr>
<tr>
<td>string</td>
<td>664</td>
</tr>
<tr>
<td>substr</td>
<td>665</td>
</tr>
<tr>
<td>substrb</td>
<td>665</td>
</tr>
<tr>
<td>time</td>
<td>665</td>
</tr>
<tr>
<td>timestamp</td>
<td>665</td>
</tr>
</tbody>
</table>
rootMembers ........................................... 749
subset ............................................. 749
unique ........................................... 750
value .............................................. 750
Report Functions .................................... 751
Today ............................................. 751
Now ................................................ 751
AsOfDate ........................................... 751
AsOfTime ........................................... 751
ReportDate ........................................ 751
ReportName ........................................ 751
ReportPath ........................................ 751
ReportDescription ................................ 752
ReportLocale ....................................... 752
GetLocale ........................................... 752
Locale ............................................... 752
ReportProductLocale ............................... 752
ReportAuthorLocale ................................ 752
ReportSaveDate ................................... 752
ReportCreateDate ................................... 752
ReportID .......................................... 753
ReportOutput ...................................... 753
ReportOption ...................................... 753
ServerName ......................................... 753
ServerLocale ....................................... 753
ModelPath ......................................... 753
BurstKey ............................................ 753
BurstRecipients ..................................... 754
IsBursting ......................................... 754
ParamNames ........................................ 754
ParamName .......................................... 754
ParamDisplayValue ................................ 754
ParamValue ......................................... 754
ParamCount ......................................... 754
RowNumber ......................................... 754
PageNumber ........................................ 755
PageCount .......................................... 755
IsPageCountAvailable .............................. 755
HorizontalPageNumber ............................. 755
HorizontalPageCount .............................. 755
PageName ........................................... 755
URLEncode ......................................... 755
TOCHeadingCount ................................... 755
IsAccessible ........................................ 756
ColumnNumber ...................................... 756
IsCrossstabRowNodeMember ......................... 756
IsCrossstabColumnNameMember ....................... 756
IsInnerMostCrossstabRowNodeMember ................ 756
IsInnerMostCrossstabColumnNameMember .............. 756
IsOuterMostCrossstabRowNodeMember ................ 756
IsOuterMostCrossstabColumnNameMember ................ 756
IsFirstColumn ...................................... 757
IsLastColumn ...................................... 757
IsLastInnerMostCrossstabColumnNameMember ............ 757
IsLastInnerMostCrossstabRowNodeMember ............. 757
CubeName .......................................... 757
CubeDescription .................................... 757
CubeCreatedOn ..................................... 757
CubeDataUpdatedOn .................................. 758
CubeSchemaUpdatedOn ................................ 758
CubeIsOptimized ................................... 758
CubeDefaultMeasure .............................................................. 758
CubeCurrentPeriod ............................................................... 758
CellValue . ........................................................................... 758
InScope .................................................................................. 758

Data Type Casting Functions ....................................................... 759
_add_days ............................................................................... 759
_add_months .......................................................................... 759
_add_years ............................................................................. 759
_age ......................................................................................... 759
_day_of_week ......................................................................... 759
_day_of_year .......................................................................... 760
_days_between .......................................................................... 760
_days_to_end_of_month ............................................................ 760
_first_of_month . ......................................................................... 760
_last_of_month ......................................................................... 761
_make_timestamp ....................................................................... 761
_months_between ....................................................................... 761
_week_of_year ........................................................................... 761
_years_between ......................................................................... 761
_ymdint_between ...................................................................... 761
abs . .......................................................................................... 762
ceiling ...................................................................................... 762
character_length ...................................................................... 762
date2string ............................................................................. 762
date2timestamp ...................................................................... 762
date2timestampTZ ................................................................. 762
DTinterval2string ..................................................................... 762
DTinterval2stringAsTime .......................................................... 763
exponent .................................................................................... 763
extract ....................................................................................... 763
floor ........................................................................................ 763
int2DTinterval ......................................................................... 764
int2YMinterval ........................................................................ 764
ln ............................................................................................ 764
lower ......................................................................................... 764
mapNumberToLetter ................................................................... 764
mod .......................................................................................... 765
nullif ......................................................................................... 765
number2string ......................................................................... 765
octet_length ............................................................................ 765
position .................................................................................... 765
power ....................................................................................... 765
round ........................................................................................ 766
sqrt .......................................................................................... 766
status ....................................................................................... 766
string2date ............................................................................ 766
string2double ......................................................................... 766
string2DTinterval .................................................................... 766
string2int32 ............................................................................. 766
string2int64 ............................................................................. 767
string2time ............................................................................. 767
string2timestamp .................................................................... 767
string2timestampTZ ............................................................... 767
string2YMinterval .................................................................... 767
substring .................................................................................. 767
time2string .............................................................................. 768
timestamp2date ......................................................................... 768
timestamp2string ...................................................................... 768
timestamp2timestampTZ .......................................................... 768
timestampTZ2date ...................................................................... 768
timestampTZ2string ................................................................... 768
Appendix G. Report Studio Object and Property Reference ............... 771

Report Studio Objects 771
3-D Area ........................................ 771
3-D Bar ........................................ 771
3-D Combination Chart ......................... 772
3-D Line ......................................... 772
3-D Scatter Chart ................................ 772
Angular Axis ..................................... 773
Angular Measure ................................ 773
Area ................................................ 773
As of Time Expression .............................. 773
Axis Labels ........................................ 774
Axis Title .......................................... 774
Bar .................................................. 774
Baseline ........................................... 775
Baseline ........................................... 775
Baseline ........................................... 775
Baseline ........................................... 775
Block ............................................... 776
Bookmark .......................................... 776
Bubble Chart ...................................... 776
Bubble Chart ...................................... 777
Bubble Measure ................................... 777
Bubble Measure ................................... 777
Bullet Chart ....................................... 777
Bullet Measure ................................... 777
Bullet Measure ................................... 777
Button ............................................... 778
Button ............................................... 778
Button Bar ......................................... 778
Button Extra Item ................................. 779
Button Icon ........................................ 779
Button Label ....................................... 779
Calculated Measure ............................... 779
Calculated Member ............................... 779
Calculated Member ............................... 779
Caption ............................................. 779
Card ................................................ 780
Card Extra Item .................................... 780
Card Label ......................................... 780
Category .......................................... 780
Category Axis ...................................... 780
Category axis ...................................... 780
Category Baseline .................................. 781
Chart Body ......................................... 781
Chart Body ......................................... 781
Chart Footer ........................................ 781
Chart Footer ........................................ 781
Chart Measure ...................................... 782
Chart Node Member ................................ 782
Chart Subtitle ..................................... 782
Chart Subtitle ..................................... 782
Chart Text Item .................................... 782
Query Operation ................................ 8 2 8
Query Reference ................................ 8 2 8
Radar Chart .................................. 8 2 8
Radial Axis .................................. 8 2 9
Radial Measure ................................. 8 2 9
Radio Button .................................. 8 2 9
Radio Button Extra Item.............................. 8 2 9
Radio Button Group ............................... 8 2 9
Radio Button Icon ................................ 8 3 0
Radio Button Label ............................... 8 3 0
Region Color Legend Title............................. 8 3 0
Region Layer .................................. 8 3 0
Region Measure ................................ 8 3 0
Regression Line ................................ 8 3 1
Regression Line Label .............................. 8 3 1
Repeater ................................... 8 3 1
Repeater Table ................................. 8 3 1
Repeater Table Cell ............................... 8 3 1
Repeater Table Cell ............................... 8 3 2
Rich Text Item................................. 8 3 2
Row Number .................................. 8 3 2
Row Number .................................. 8 3 2
Scatter Chart ................................. 8 3 3
Scatter Marker ................................. 8 3 4
Secondary Axis................................. 8 3 4
Secondary Bottom Axis.............................. 8 3 4
Select & Search Prompt.............................. 8 3 4
Set Expression ................................ 8 3 5
Singleton ................................... 8 3 5
Slicer Member Set................................ 8 3 5
SQL .................................... 8 3 5
Static Repeater Table............................... 8 3 5
Summary Filter. ................................ 8 3 6
Tab ..................................... 8 3 6
Tab Control .................................. 8 3 6
Tab Extra Item ................................. 8 3 6
Tab Icon ................................... 8 3 7
Tab Label ................................... 8 3 7
Table .................................... 8 3 7
Table Cell .................................. 8 3 7
Table of Contents ................................ 8 3 7
Table of Contents Entry.............................. 8 3 8
Table Row .................................. 8 3 8
Target Measure................................. 8 3 8
Target Measure................................. 8 3 8
Text Box Prompt ................................ 8 3 8
Text Item ................................... 8 3 9
Time .................................... 8 3 9
Time Prompt.................................. 8 3 9
Toggle Button Bar................................ 8 4 0
Tolerance Measure................................ 8 4 0
Total Column ................................ 8 4 0
Total Column Label ............................... 8 4 0
Tree Prompt .................................. 8 4 0
Trendline ................................... 8 4 1
Trendline Label ................................ 8 4 1
Value .................................... 8 4 1
Value .................................... 8 4 1
Value .................................... 8 4 1
Value Prompt .................................. 8 4 1
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Titles</td>
<td>867</td>
</tr>
<tr>
<td>Column Visibility</td>
<td>867</td>
</tr>
<tr>
<td>Combinations</td>
<td>868</td>
</tr>
<tr>
<td>Component Reference</td>
<td>868</td>
</tr>
<tr>
<td>Conditional Palette</td>
<td>868</td>
</tr>
<tr>
<td>Conditional Styles</td>
<td>868</td>
</tr>
<tr>
<td>Connecting Lines</td>
<td>869</td>
</tr>
<tr>
<td>Contained Text Direction</td>
<td>870</td>
</tr>
<tr>
<td>Container Filter</td>
<td>870</td>
</tr>
<tr>
<td>Container Select</td>
<td>870</td>
</tr>
<tr>
<td>Contents Height</td>
<td>871</td>
</tr>
<tr>
<td>Control Data Item Value</td>
<td>871</td>
</tr>
<tr>
<td>Control Data Item Value</td>
<td>871</td>
</tr>
<tr>
<td>Control Data Item Value</td>
<td>871</td>
</tr>
<tr>
<td>Control Enable</td>
<td>872</td>
</tr>
<tr>
<td>Cross Product Allowed</td>
<td>872</td>
</tr>
<tr>
<td>Cumulation Axis</td>
<td>872</td>
</tr>
<tr>
<td>Cumulation Label</td>
<td>872</td>
</tr>
<tr>
<td>Cumulative Line</td>
<td>872</td>
</tr>
<tr>
<td>Cumulative Line</td>
<td>873</td>
</tr>
<tr>
<td>Current Block</td>
<td>873</td>
</tr>
<tr>
<td>Current Card</td>
<td>873</td>
</tr>
<tr>
<td>Current Tab</td>
<td>873</td>
</tr>
<tr>
<td>Custom Label</td>
<td>873</td>
</tr>
<tr>
<td>Custom Label</td>
<td>873</td>
</tr>
<tr>
<td>Data Format</td>
<td>874</td>
</tr>
<tr>
<td>Data Format</td>
<td>874</td>
</tr>
<tr>
<td>Data Item</td>
<td>874</td>
</tr>
<tr>
<td>Data Item</td>
<td>875</td>
</tr>
<tr>
<td>Data Item Label</td>
<td>875</td>
</tr>
<tr>
<td>Data Item Label</td>
<td>875</td>
</tr>
<tr>
<td>Data Item Label</td>
<td>875</td>
</tr>
<tr>
<td>Data Item Label</td>
<td>875</td>
</tr>
<tr>
<td>Data Item Value</td>
<td>875</td>
</tr>
<tr>
<td>Data Item Value</td>
<td>875</td>
</tr>
<tr>
<td>Data Item Value</td>
<td>876</td>
</tr>
<tr>
<td>Data Item Value</td>
<td>876</td>
</tr>
<tr>
<td>Data Item Value</td>
<td>876</td>
</tr>
<tr>
<td>Data Item Value</td>
<td>876</td>
</tr>
<tr>
<td>Data Item Value</td>
<td>876</td>
</tr>
<tr>
<td>Data Item Value</td>
<td>876</td>
</tr>
<tr>
<td>Data Language</td>
<td>876</td>
</tr>
<tr>
<td>Data Points</td>
<td>877</td>
</tr>
<tr>
<td>Data Source</td>
<td>877</td>
</tr>
<tr>
<td>Days Text</td>
<td>877</td>
</tr>
<tr>
<td>Deck Cards Definition</td>
<td>877</td>
</tr>
<tr>
<td>Default Card</td>
<td>877</td>
</tr>
<tr>
<td>Default Card</td>
<td>877</td>
</tr>
<tr>
<td>Default Card</td>
<td>877</td>
</tr>
<tr>
<td>Default Measure</td>
<td>878</td>
</tr>
<tr>
<td>Default Selections</td>
<td>878</td>
</tr>
<tr>
<td>Default Title</td>
<td>878</td>
</tr>
<tr>
<td>Default Title</td>
<td>878</td>
</tr>
<tr>
<td>Define Contents</td>
<td>878</td>
</tr>
<tr>
<td>Define Custom Grouping</td>
<td>878</td>
</tr>
<tr>
<td>Define Member Sets</td>
<td>878</td>
</tr>
<tr>
<td>Definition</td>
<td>879</td>
</tr>
<tr>
<td>Depth</td>
<td>879</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Position in List</td>
<td>925</td>
</tr>
<tr>
<td>Position type</td>
<td>925</td>
</tr>
<tr>
<td>Position type</td>
<td>925</td>
</tr>
<tr>
<td>Position type</td>
<td>925</td>
</tr>
<tr>
<td>Positive Column Color</td>
<td>926</td>
</tr>
<tr>
<td>Pre-populate</td>
<td>926</td>
</tr>
<tr>
<td>Pre-populate Levels</td>
<td>926</td>
</tr>
<tr>
<td>Pre-Sort</td>
<td>926</td>
</tr>
<tr>
<td>Processing</td>
<td>926</td>
</tr>
<tr>
<td>Progressive Axis</td>
<td>927</td>
</tr>
<tr>
<td>Progressive Palette</td>
<td>927</td>
</tr>
<tr>
<td>Projection List</td>
<td>927</td>
</tr>
<tr>
<td>Properties</td>
<td>927</td>
</tr>
<tr>
<td>Property</td>
<td>928</td>
</tr>
<tr>
<td>Property Unique Name</td>
<td>928</td>
</tr>
<tr>
<td>Push To Bottom</td>
<td>928</td>
</tr>
<tr>
<td>Query</td>
<td>928</td>
</tr>
<tr>
<td>Query</td>
<td>929</td>
</tr>
<tr>
<td>Radar Type</td>
<td>929</td>
</tr>
<tr>
<td>Radial Axis</td>
<td>929</td>
</tr>
<tr>
<td>Radio Buttons Definition</td>
<td>929</td>
</tr>
<tr>
<td>Range</td>
<td>929</td>
</tr>
<tr>
<td>Range Label</td>
<td>930</td>
</tr>
<tr>
<td>Range Type</td>
<td>930</td>
</tr>
<tr>
<td>Regression Line</td>
<td>930</td>
</tr>
<tr>
<td>Regression Type</td>
<td>930</td>
</tr>
<tr>
<td>Relative Alignment</td>
<td>930</td>
</tr>
<tr>
<td>Remove Text</td>
<td>931</td>
</tr>
<tr>
<td>Render</td>
<td>931</td>
</tr>
<tr>
<td>Render Fact Cells</td>
<td>931</td>
</tr>
<tr>
<td>Render Page when Empty</td>
<td>931</td>
</tr>
<tr>
<td>Render Variable</td>
<td>932</td>
</tr>
<tr>
<td>Repeater Direction</td>
<td>932</td>
</tr>
<tr>
<td>Repeater Table Values Definition</td>
<td>933</td>
</tr>
<tr>
<td>Report Expression</td>
<td>933</td>
</tr>
<tr>
<td>Report Expression</td>
<td>933</td>
</tr>
<tr>
<td>Report Expression</td>
<td>933</td>
</tr>
<tr>
<td>Report Expression</td>
<td>933</td>
</tr>
<tr>
<td>Report Expression</td>
<td>933</td>
</tr>
<tr>
<td>Report Expression</td>
<td>933</td>
</tr>
<tr>
<td>Report Expression</td>
<td>934</td>
</tr>
<tr>
<td>Report Expression</td>
<td>934</td>
</tr>
<tr>
<td>Report Expression</td>
<td>934</td>
</tr>
<tr>
<td>Report Expression</td>
<td>934</td>
</tr>
<tr>
<td>Report Expression</td>
<td>934</td>
</tr>
<tr>
<td>Report Expression</td>
<td>934</td>
</tr>
<tr>
<td>Report Expression</td>
<td>934</td>
</tr>
<tr>
<td>Required</td>
<td>934</td>
</tr>
<tr>
<td>Results Deselect All Text</td>
<td>935</td>
</tr>
<tr>
<td>Results Select All Text</td>
<td>935</td>
</tr>
<tr>
<td>Results Text</td>
<td>935</td>
</tr>
<tr>
<td>Reverse Category Order</td>
<td>935</td>
</tr>
<tr>
<td>Right Position</td>
<td>935</td>
</tr>
<tr>
<td>Right Position (px)</td>
<td>936</td>
</tr>
<tr>
<td>Rollup Aggregate Function</td>
<td>936</td>
</tr>
<tr>
<td>Rollup Method</td>
<td>936</td>
</tr>
<tr>
<td>Rollup Processing</td>
<td>936</td>
</tr>
<tr>
<td>Root Members Only</td>
<td>936</td>
</tr>
<tr>
<td>Rotate Values</td>
<td>937</td>
</tr>
<tr>
<td>Row Coordinate</td>
<td>937</td>
</tr>
<tr>
<td>Rows Per Page</td>
<td>937</td>
</tr>
<tr>
<td>Scale</td>
<td>937</td>
</tr>
<tr>
<td>Scale Interval</td>
<td>938</td>
</tr>
<tr>
<td>Scope</td>
<td>938</td>
</tr>
</tbody>
</table>
Introduction

This document is intended for use with IBM® Cognos® Report Studio. It provides step-by-step procedures and background information to help you create standard and complex reports. Report Studio is a Web product for creating reports that analyze corporate data according to specific information needs.

Finding information

To find IBM Cognos product documentation on the web, including all translated documentation, access one of the IBM Cognos Information Centers (http://pic.dhe.ibm.com/infocenter/cogic/v1r0m0/index.jsp). Release Notes are published directly to Information Centers, and include links to the latest technotes and APARs.

You can also read PDF versions of the product release notes and installation guides directly from IBM Cognos product disks.

Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products. Report Studio has accessibility features. For more information, see Appendix A, “Accessibility Features,” on page 467.

IBM Cognos HTML documentation has accessibility features. PDF documents are supplemental and, as such, include no added accessibility features.

Forward-looking statements

This documentation describes the current functionality of the product. References to items that are not currently available may be included. No implication of any future availability should be inferred. Any such references are not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of features or functionality remain at the sole discretion of IBM.

Samples disclaimer

The Sample Outdoors Company, Great Outdoors Company, GO Sales, any variation of the Sample Outdoors or Great Outdoors names, and Planning Sample depict fictitious business operations with sample data used to develop sample applications for IBM and IBM customers. These fictitious records include sample data for sales transactions, product distribution, finance, and human resources. Any resemblance to actual names, addresses, contact numbers, or transaction values is coincidental. Other sample files may contain fictional data manually or machine generated, factual data compiled from academic or public sources, or data used with permission of the copyright holder, for use as sample data to develop sample applications. Product names referenced may be the trademarks of their respective owners. Unauthorized duplication is prohibited.
Chapter 1. What's New in Cognos Report Studio?

This section contains a list of new, changed, and removed features for this release. It will help you plan your upgrade and application deployment strategies and the training requirements for your users.

For information about upgrading, see the IBM Cognos Business Intelligence Installation and Configuration Guide for your product.

For information about other new features for this release, see the IBM Cognos Business Intelligence New Features Guide.

To view What's New information for past releases, including version 8.4, access one of the IBM Cognos Information Centers (http://pic.dhe.ibm.com/infocenter/cogic/v1r0m0/index.jsp).

To review an up-to-date list of the environments that are supported by IBM Cognos products, including information about operating systems, patches, browsers, web servers, directory servers, database servers, and application servers, visit the IBM Cognos Customer Center (http://www.ibm.com/software/data/cognos/customercenter).

New features in version 10.2.1

The following topics describe the new features in this release.

**Improved IBM Cognos Active Report performance**

Active Report performance is improved in this release.

You can now compress Active Report output. The compression feature compresses Active Report controls within the output and produces files that are smaller and open faster. The compression feature significantly improves the performance of Active Reports that include large amounts data versus graphical content. The option to compress Active Report output is set in the Active Report Properties window.

In addition, improvements to the Active Report engine streamline performance, and result in improved memory consumption characteristics. With these improvements, large active reports perform better, particularly when they are consumed within the IBM Cognos Business Intelligence Mobile application.

**Related tasks:**

“Specify Active Report Properties” on page 173

You can specify properties for active reports, such as the text that appears in the browser window title bar, the maximum number of rows that a report can contain, and an access code that users must enter to view a report.

**Extensible visualizations in IBM Cognos Active Report**

With extensible visualizations, you can use innovative new visualizations in active reports.
IBM Cognos Business Intelligence introduces new visualizations to Report Studio, such as treemaps and packed bubble visualizations. Your administrator can download sample visualizations to a gallery that is accessible from the Active Report authoring environment. Sample visualizations can be used or modified to meet your requirements.

**Related concepts:**
- “Extensible visualizations” on page 173

You can use extensible client-side visualizations in IBM Cognos Active Report to visualize data.

---

**Option to group or ungroup repeating cells in Excel 2007 reports**

When you produce reports in Excel 2007 format, you can choose to group repeating cells into a single merged cell.

Grouped or ungrouped cells are produced by various report objects, such as grouped data items, headers and footers, and nested objects. The option to group or ungroup repeating cells is controlled by the **Group repeating cells when exporting to Excel** report property.

**Related concepts:**
- “Option to group repeating cells in reports produced in Excel 2007 format” on page 50

When you produce reports in Excel 2007 format, you can specify whether repeating cells are grouped, or merged, into a single cell.

---

**Crosstabs values as percentages**

In crosstabs, you can now show values as a percentage of a summary instead of the actual values. For example, you can show the revenue that was generated by each product line as a percentage of the total revenue.

**Related tasks:**
- “Showing crosstab values as percentages” on page 72

When you work with dimensional data sources, you can show crosstabs values as a percentage of a summary instead of the actual values. When you show values as a percentage, you can compare the contribution of an item to the whole.

---

**Share sets between reports**

When you work with dimensional data, you can now share a set. Shared sets appear in the model tree for inclusion in other reports.

When you include a shared set, you can designate it to be either **Run Time** or **Design Time**. If a set is designated as **Run Time**, the report always uses whatever is in the shared set as it is defined. If a set is designated as **Design Time**, the report caches the contents of the shared set at the time when it is initially included. The report uses this definition until you explicitly refresh it.

**Related tasks:**
- “Sharing sets between reports” on page 259

When you are working with dimensional data, you can share a set to make it available for inclusion in multiple reports.

---

**Improved text-based relational filters**

You can now define relational filters that are based on a text pattern.
In previous releases, text-based relational filters were defined by selecting actual values, such as Camping Equipment. Performance is impacted when the number of strings in a filter condition increases. This feature automatically generates filters that contain strings or members that match a text pattern. The following text matching options are available:

- Starts with
- Ends with
- Contains
- Matches SQL pattern

**Related tasks:**

- "Create a Detail or Summary Filter" on page 229

Add a filter expression to focus a report and minimize processing time by excluding unwanted data. For example, you can filter data to show only customers who placed purchase orders that were valued at over one thousand dollars during the past year. When you run the report, you see only the filtered data.

**Improved data container and query naming support**

In IBM Cognos Report Studio, when you insert a data container from the toolbox tab into a report, you can immediately specify a name for the container and a name for the query that you want to link to the container. You can also link an existing query to the container.

In previous releases, Cognos Report Studio automatically generated the names for data containers that are inserted in a report and the names of the queries that are linked to the containers. For example, when you insert a list, the generated name of the list is List1, and the name of the query that is linked to the list is Query1. To change either name, you must modify the **Name** property of the object in the **Properties** pane.

**Related tasks:**

- "Inserting a data container from the toolbox tab" on page 41

In addition to choosing a basic report layout when you create a report, you can insert data containers from the toolbox tab into a report in IBM Cognos Report Studio.

**Search parameter Ends with any of these keywords**

A search parameter, **Ends with any of these keywords**, is now available when you perform searches, such as when you search for members when you create a filter expression.

---

**Removed Features in Version 10.2.1**

The following features were removed since the last release.

**IBM Cognos Statistics**

IBM Cognos Statistics was removed from this version of IBM Cognos Business Intelligence.

To ensure that reports that were created in previous releases, and that contain statistical objects, will run in this and future releases, statistical objects are removed when the reports are upgraded. Each removed statistical object is replaced with the following image:
Figure 1. Image that replaces statistical objects in upgraded reports

A warning also appears in the Upgrade Information window for each statistical object that is replaced with the image.

Tip: Queries and their data items that are associated to statistical objects are not removed from upgraded reports.

You can use IBM SPSS® Statistics to perform statistical reporting and analysis.

New features in version 10.2.0

This section describes new features in this release.

Updated world and Africa maps

The world and Africa maps were updated to include the region of South Sudan in IBM Cognos Report Studio version 10.2.0.

Related tasks:
[“Use maps from previous versions” on page 9]

The old versions of the World, Europe, Africa, and India maps that were included with IBM Cognos Business Intelligence prior to version 10.1.1 are still available. To continue using the previous maps, your IBM Cognos BI administrator must enable them.

Interactive repeater tables

You can now add interactive behavior to repeater tables in IBM Cognos Active Report.

Adding interactive behavior to repeater tables is useful for creating button bars and grids of custom content. For example, you can insert a data item in a repeater table that shows a different chart for each data value. You can customize the repeater table so that the data values appear as buttons in a two-column grid.

Related concepts:
[“Adding Controls to a Report” on page 185]

IBM Cognos Active Report offers several active report controls. For many controls, two different versions are available, a static version and a data-driven version.

Improved IBM Cognos Active Report integration with IBM Cognos Workspace

Check boxes and sliders in workspaces in IBM Cognos Workspace can now drive IBM Cognos Active Report.

You can define public variables and range variables in Cognos Active Report that allow Cognos Workspace check boxes and sliders to drive active reports in a workspace.
Related tasks:

“Create and Manage Active Report Variables” on page 184
Create active report variables to allow controls in a report to interact with each other. Active report variables pass information from one control to another control.

**Improved support for bidirectional languages**

IBM Cognos Report Studio offers improved support for bidirectional languages.

The improved support for bidirectional languages includes the following capabilities:
- Base text direction for text in more report objects, such as charts, maps, and prompts
- Control of object direction for charts, maps, and prompts
- Digit shaping

In addition, the improved support for bidirectional languages includes the report formats PDF, Microsoft Excel, and IBM Cognos Active Report.

**Related concepts:**

“Support for bidirectional languages” on page 56
You can author reports that support bidirectional languages. You can specify base text direction, digit shaping, and container direction.

**Macro tab in expression editor**

A macro tab is now available in the IBM Cognos Report Studio expression editor. With the macro tab, you can add parameter maps, session parameters, and macro functions to expressions.

**Related tasks:**

“Insert a Query Calculation” on page 250
Insert a query calculation into your report to add a new row or column with values that are based on a calculation.

“Insert a Query Calculation” on page 303
When working with dimensional data, insert a query calculation into your report to add a new row or column with values that are based on a calculation. For example, you create a query calculation named Euros that converts dollars to euros by multiplying an existing dollar measure by a conversion rate. Euros can then be displayed to end users in a separate row or column.

**Inherited table styles in lists and crosstabs**

When an item is inserted in a list or crosstab, the item inherits the table style of one of the items in the data container.

By default, items inherit the style of an item of the same type. For example, if you insert a measure in a list, the measure inherits the style of a measure that is in the list, if one exists. Inheritance rules control which style is inherited.
Related tasks:

"Apply a Table Style" on page 340
Apply a table style to quickly format tables. You can also apply a table style to lists, crosstabs, and repeater tables.

Enhanced crosstab headers
In crosstabs, you can display headers that indicate the parent categories of the data items in the columns and rows. The headers help users to understand where the data is located in the hierarchy.

Related tasks:

"Creating headers automatically for sets in crosstabs" on page 72
You can automatically add header labels on columns and rows when working with sets in a crosstab that uses a dimensional data source. The headers help consumers of the report to understand where the data is in the hierarchy.

Prompt API
A JavaScript Prompt API is available to provide report authors with a method of customizing prompt interaction in the reports that they author.

With the prompt API, JavaScript applications can interact with IBM Cognos Business Intelligence report prompts for the purposes of validation or custom interaction. The prompt API can query and set user selections, validate typed-in values using patterns, set default values using expressions or query data, and more. Sample reports exist to demonstrate various usage scenarios of the prompt API.

Related reference:

Appendix H, “Prompt API for IBM Cognos BI,” on page 975
The JavaScript Prompt API provides report authors with a method of customizing prompt interaction in the reports they author.

Excel 2007 Data report output format

This format is similar to a comma-separated values file (.csv). It has no formatting such as headers, footers, styling, or data formatting.

Related concepts:

"Producing a Report in Microsoft Excel Format" on page 49
You can export your report output to several different Microsoft Excel spreadsheet software formats.

Global classes for accessible reports
New global classes that you can apply to text objects were added to IBM Cognos Report Studio. The new classes make text easier to read for visually impaired users.

The new global classes are as follows:

- Heading 1 to heading 6
- Emphasize text
- Strong text
You can apply certain global classes to text objects to make the text easier to read for visually impaired users.

**Changed features in version 10.2.0**

This section describes changed features since the last release.

**Support for inline prompts in reports viewed on mobile devices**

IBM Cognos Mobile now supports several inline prompt types. Previously, all inline prompts in reports viewed on a mobile device appeared in a prompt page.

The prompt types that Cognos Mobile supports as inline prompts are as follows:

- Text Box
- Value
- Date
- Time
- Date & Time
- Interval

**Cognos Business Insight and Cognos Business Insight Advanced are renamed**

Two product components are renamed in this release to avoid confusion with the names of other components. IBM Cognos Business Insight is now IBM Cognos Workspace. IBM Cognos Business Insight Advanced is now IBM Cognos Workspace Advanced.

**Removed Features in Version 10.2.0**

This section describes features that were removed since the last release.

**Saving and opening reports on your computer**

The ability to save and open reports locally on your computer by enabling the Allow local file access option was removed in this release.

To save and open reports on your computer, use the Copy Report to Clipboard and Open Report from Clipboard commands in the Tools menu instead.

**New features in version 10.1.1**

Listed below are new features since the last release.
New active report controls
IBM Cognos Active Report offers new controls, including iterators, sliders, and row numbers.

Related concepts:
“Adding Controls to a Report” on page 185
IBM Cognos Active Report offers several active report controls. For many controls, two different versions are available, a static version and a data-driven version.

Conversion of controls to other types of controls
You can now convert a control in an active report to another type of control that is compatible to the original control. For example, you can convert a radio button group control to a check box group control.

Related concepts:
“Adding Controls to a Report” on page 185
IBM Cognos Active Report offers several active report controls. For many controls, two different versions are available, a static version and a data-driven version.

Improved user experience when authoring active reports
Improvements were made to the IBM Cognos Report Studio user interface for authoring active reports. For example, there are two new tabs that respectively show the controls and variables defined in a report.

Controls are shown in a hierarchical structure, allowing you to see their relationships to one another as well as the variables that drive them. For each variable in a report, you can see the default values defined for each one, as well as the controls to which they are linked.

Related tasks:
“Create and Manage Active Report Variables” on page 184
Create active report variables to allow controls in a report to interact with each other. Active report variables pass information from one control to another control.
“Define a Connection Between Controls” on page 192
Define a connection between controls to link them together. When you link controls, an action performed on one control affects the behavior of the other control.

Active report access code
You can now specify an access code that users will have to enter before they can view an active report.

Related tasks:
“Specify Active Report Properties” on page 173
You can specify properties for active reports, such as the text that appears in the browser window title bar, the maximum number of rows that a report can contain, and an access code that users must enter to view a report.

New startup options
You can set various options that control the appearance and behavior of IBM Cognos Report Studio ([Tools, Options]).

### Keyboard shortcuts for accessibility

To make the report authoring experience more accessible to people who have a physical disability, such as restricted mobility or limited vision, IBM Cognos Report Studio now includes new keyboard shortcuts to allow navigating the user interface with a keyboard.

This product uses some standard Microsoft Windows and accessibility shortcut keys.

### Updated maps

Some maps included with IBM Cognos Report Studio version 10.1.1 were updated.

- The World and Europe maps were updated to include the countries and regions of Montenegro and Kosovo.
- The India map was updated to include the northern region of Kashmir.

IBM Cognos Report Studio provides a set of maps that you can use to represent tabular data in a spatial context. For example, on a map of the world, countries and regions can be colored to represent the level of revenue.

### Use maps from previous versions

The old versions of the World, Europe, Africa, and India maps that were included with IBM Cognos Business Intelligence prior to version 10.1.1 are still available. To continue using the previous maps, your IBM Cognos BI administrator must enable them.

**Procedure**

1. Go to the `{c10_location}/maps` directory.
2. Rename the files `world.cmf`, `europe.cmf`, `africa.cmf`, and `india.cmf` to remove the `.cmf` extension, as follows: `world_default.cmf`, `europe_default.cmf`, `africa_default.cmf`, and `india_default.cmf`.
3. Rename the files `world8_4.cmf`, `europe8_4.cmf`, `africa8_4.cmf`, and `indiaUN.cmf` to remove `8_4` and add the `.cmf` extension, as follows: `world.cmf`, `europe.cmf`, `africa.cmf`, and `india.cmf`.

### Custom groups

You can create custom groups of existing data items based on elements that you define and that are meaningful to you.

For example, you can create custom groups for product names based on their alphabetical order. You might create groups for products that start with the letters A to C, D to F, G to H, and so on.
Related tasks:

“Creating relational custom groups” on page 215
Create custom groups to classify existing data items into groups that are meaningful to you.

“Creating dimensional custom groups” on page 265
Create custom groups to classify existing data items into groups that are meaningful to you.

Larger worksheet sizes for Microsoft Excel 2007 report outputs

IBM Cognos Business Intelligence now supports a larger worksheet size for report outputs exported in Microsoft Excel 2007 format.

In previous versions of IBM Cognos Business Intelligence, report output in Microsoft Excel format was limited to a maximum of 256 columns by 65,000 rows. Although this remains as the default worksheet size, administrators can now enable larger worksheets and change the maximum number of rows in a worksheet - up to a maximum of 16,384 columns by 1,048,576 rows - by using advanced server properties. This number matches the Microsoft Excel 2007 worksheet size limitations.

For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

For an up-to-date list of environments supported by IBM Cognos products, such as operating systems, patches, browsers, Web servers, directory servers, database servers, and application servers, visit http://www.ibm.com/.

Related concepts:

“Producing a Report in Microsoft Excel Format” on page 49
You can export your report output to several different Microsoft Excel spreadsheet software formats.

Microsoft Excel 2007 supported as an external data source

IBM Cognos Business Intelligence now supports external data sources from Microsoft Excel up to version Microsoft Excel 2007.

For an up-to-date list of environments supported by IBM Cognos products, such as operating systems, patches, browsers, Web servers, directory servers, database servers, and application servers, visit http://www.ibm.com/.

Related concepts:

“Use Your Own External Data” on page 11
You can now supplement your enterprise data with your own external data file.

Changed features in version 10.1.1

Listed below are changed features since the last release.

Specify how to show no data contents

The user interface for specifying what appears for data containers that contain no data was modified.

You can now select between showing an empty data container, alternate content that you specify, or text that you specify.
The new default for data containers that have no data available is to show the text **No Data Available** instead of showing an empty data container.

**Related tasks:**

[“Specify what appears for data containers that contain no data” on page 392](#)

You can specify what appears in a data container when no data is available from the database.

---

**What's new in version 10.1.0**

This section contains a list of new, changed, and removed features for past releases.

Knowing this information will help you plan your upgrade and application deployment strategies and the training requirements for your users.

To review an up-to-date list of environments supported by IBM Cognos products, such as operating systems, patches, browsers, Web servers, directory servers, database servers, and application servers, visit [www.ibm.com](http://www.ibm.com).

For information about upgrading, see the IBM Cognos Business Intelligence *Installation and Configuration Guide* for your product.

For an overview of new features for this release, see the IBM Cognos Business Intelligence *New Features Guide*.

**New Features in Version 10.1.0**

Listed below are new features since the last release.

**Use Your Own External Data**

You can now supplement your enterprise data with your own external data file.

You can import an .xls, .txt, .csv, or .xml file into IBM Cognos and create reports that contain your own data. This allows you to leverage the IBM Cognos platform, including the security, data integrity, and full range of reporting functionality.

You can perform self-service reporting without the need to ask your IT department or administrator to set up a data source connection to your file. You import your own data file and start reporting on it right away.

After importing, your external data file is protected by the same IBM Cognos security as your enterprise data, thus allowing you to report on your data in a secure environment.

**Active Reports**

You can now use IBM Cognos Report Studio to create active reports.

IBM Cognos Active Report is a report output type that provides a highly interactive and easy-to-use managed report. Active reports are built for business users, allowing them to explore their data and derive additional insight.

Active reports make business intelligence easier for the casual user. Report authors build reports targeted at their users’ needs, keeping the user experience simple and engaging. Active reports can be consumed by users who are offline, making them an ideal solution for remote users such as the sales force.
Related concepts:

Chapter 8, “Active Reports,” on page 171

You can use IBM Cognos Report Studio to create active reports. IBM Cognos Active Report is a report output type that provides a highly interactive and easy-to-use managed report. Active reports are built for business users, allowing them to explore their data and derive additional insight.

Statistical Analysis

IBM Cognos Report Studio now offers statistical functionality in an add-on package called IBM Cognos Statistics.

Report Studio integrates statistical analysis and reporting, providing you with the capability to distribute reports that now can include statistical information. Unlike specialized statistical tools, this functionality is easy to use within the IBM Cognos Business Intelligence authoring environment.

This functionality covers three main areas in statistical analysis:

- the distribution of data
- data analysis and testing
- statistical process control

Mozilla Firefox Support

Versions 3.5 and 3.6 of the Mozilla Firefox Web browser are now supported for IBM Cognos Report Studio.

For a full list of supported software environments, see http://www.ibm.com/.

Enhanced charts

Chart enhancements concentrate on improved effectiveness of communication. The new bullet chart and extensions to pie and gauge charts are especially beneficial in dashboard reports. Additional enhancements provide superior presentation through conditional notes and markers and improved legends.

Current Default Charts and Legacy Charts:

IBM Cognos Report Studio has a new default chart technology. Any new charts added to a report use the current default charts.

The current default charts use more properties that allow you to customize more aspects of the chart. In addition, the current default charts offer you a preview of the style changes that you make to your chart.

If you prefer to continue working with the legacy charts, you can select the Use legacy chart authoring option.

You can also convert legacy charts to the current default charts. When you convert a chart to a different chart type, Report Studio maps the chart properties that exist in the new chart type. For more information, see “Upgrading Legacy Charts” on page 462 and “Convert Charts From One Type to Another” on page 102.
Bullet Charts:

You can now create bullet charts to complement your dashboard reports and as an alternative to gauge charts.

A bullet chart features a single primary measure, such as current revenue, compares that measure to one or more other measures to enrich its meaning, such as the target or planned revenue, and displays it in the context of a qualitative range of performance, such as poor, satisfactory, or good.

Related concepts:

“Bullet Charts” on page 93

Bullet charts are a variation of bar charts. They compare a featured measure (the bullet) to a targeted measure (the target). They also relate the compared measures against colored regions in the background that provide additional qualitative measurements, such as good, satisfactory, and poor.

Chart Matrix:

You can now convert a complex nested chart into a matrix configuration that shows multiple small charts arranged in rows and columns.

The numeric scale of all the charts is the same so that you can easily compare and analyze values from each chart.

Related tasks:

“Create a Matrix of Current Default Charts” on page 142

You can view a complex chart that includes nested series or categories into a matrix, or crosstab, that shows multiple small charts arranged in rows and columns.

Customize the Legend:

You can hide or show the legend and change its position relative to the chart object, chart area, or using a specific report expression.

For more information, see “Customize the Legend in a Current Default Chart” on page 124.

If your legend includes items that are too long, you can truncate long legend items at a specific number of characters. You can also show the values of the data items in your legend. For more information, see “Customize the Items in the Legend of a Current Default Chart” on page 125.

Legends can display trend lines and conditional item entries.

Colored Regions:

You can now define colored regions in the body of a chart to make your chart more informative.
For example, you can divide the background of a scatter chart into quadrants and color each quadrant. For more information, see “Add Colored Regions in a Current Default Chart” on page 117.

Enhanced Pie Charts:

You can customize how data labels appear in a pie chart, including their format and placement.

For more information, see “Showing data values in current default pie and donut charts” on page 137.

To make your pie charts easier to read and analyze, you can now:

- pull out pie slices from the rest of the pie to highlight them. For more information, see “Pull Out Pie Slices in a Current Default Pie Chart” on page 149.
- set the position at which the first slice in a pie chart appears. For more information, see “Set the Position of the First Slice in a Current Default Pie Chart” on page 148.
- summarize the smaller slices in a pie chart to avoid having many tiny slices in your pie. For more information, see “Summarize Small Slices, Bars, or Columns in Current Default Charts” on page 139.

Enhanced Chart Styles:

You can customize the palette, drop shadows, gridlines, backgrounds, and bar shapes in charts.

You can create a new chart palette or edit an existing chart palette, including specifying colors, fill types, and patterns. For more information, see “Customizing the Color Palette of a Chart” on page 108.

You can add drop shadows to chart elements, such as lines in line charts. For more information, please see “Add Background Effects to a Chart Object in a Legacy Chart” on page 116.

You can customize the color, style, and weight of chart gridlines or set the background of your chart to display bands of color. For more information, see “Show Gridlines in a Current Default Chart” on page 123.

You can change the bar shape in a bar or column chart to one of the following two dimensional and three dimensional shapes: box, cylinder, cone, pyramid, rectangle, and triangle. You can also change the width of the bars or columns in a bar or column chart to a percentage of the chart body, such as 80% or 25%.

Chart Axes:

To make your charts easier to read, you can now customize the color, style, weight, and transparency of chart axis lines and specify where to display major and minor gridlines on the chart.

For more information, see “Customize the Axes of a Current Default Chart” on page 120.
Color by Value in Scatter and Bubble Charts:

In scatter and bubble charts, you can now specify data to appear in different colors based on the data values.

The color of the points or bubbles adds an additional aspect to the chart to assist you in finding relationships in large amounts of data. An entry appears in the legend to identify the color and its value. For more information, see “Define Color by Value in Current Default Scatter and Bubble Charts” on page 146.

Bubble Size:

In a bubble chart, you can now set the maximum and minimum bubble size and specify the smallest bubble to show in the chart.

For more information, see “Specify the Bubble Size in a Current Default Bubble Chart” on page 148.

Positioning and Formatting Notes:

When you add a note to a chart, you can now set the note's position relative to the sides of the chart area or chart body. You can also use a report expression to position the note next to a data item within the chart.

For example, in a bar chart showing revenue for each product line by country or region, you could type [Country or Region]='Canada' to position the note relative to the Canada bar. You can also customize the background of the note with color and background images. For more information, see “Add a Note to a Current Default Chart” on page 127.

Insert Charts With the Toolbar Button:

You can now insert a specific chart type from any report with the insert chart button on the toolbar. When you insert a chart into a report that already contains a list, the list data is added to the chart automatically.

Enhancements to the User Interface

IBM Cognos Report Studio includes enhancements to the user interface layout, buttons, tooltips, and visual aids.

These enhancements include the following:

• You can now position the content and Properties panes to the right of the work area, to match the user interface of IBM Cognos Business Insight and IBM Cognos Business Insight Advanced. For more information, see “Options” on page 32.

• When working with dimensional data sources, the insert individual members button and the insert member with children button have moved to the Source tab. These buttons have the same function as they did in the previous release. For more information, see “Insert a Member” on page 255.

• When working with dimensional and mixed model data sources, you can now switch between viewing the full data tree and the dimensional-only data tree by
clicking the view package tree button and the view members tree button. For more information, see “Add Dimensional Data to a Report” on page 253.

These same buttons are available in the expression editor.

- You can now remove all of the filters at once from your report with new options in the filters button. For more information, see “Focusing Dimensional Data” on page 278 and “Focusing Relational Data” on page 228.

- Expanded tooltips for toolbar buttons now include a description of what the button does. To see a tooltip, pause the pointer over a button in the toolbar. You can turn off the expanded tooltips (Tools, Options).

- A new **Show Container Selectors** visual aid allows you to select list, crosstab, repeater table, table of contents, active report application objects, and table container objects with one click. When enabled, a small clickable region (three orange dots) appears in the top left corner of these report objects. This selector provides easier access to the container objects’ properties and allows you to easily move containers with one mouse gesture. For more information, see “Visual Aids Button” on page 27.

### Improved Set Operations

You can now perform more operations on a set of members when working with dimensional data sources.

You can do the following by right-clicking a set, and clicking **Edit Set**:

- Exclude members from the initial set or from the set as it is currently shown. For more information, see “Exclude Members from a Set” on page 261.

- Move members to the top or bottom of the set. For more information, see “Move Members Within a Set” on page 261.

- Show only the members with the top or bottom values. For more information, see “Limit Data to Top or Bottom Values” on page 262.

- Filter the members in a set. For more information, see “Filter the Members Within a Set” on page 262.

- Expand and collapse members to add its child members below it as new rows. For more information, see “Expand and Collapse a Member Within a Set” on page 263.

### Preview Reports when Opening and Saving

You can now see a preview of the report when you open or save it (Tools, Options, Display report preview).

For more information, see “Options” on page 32.

### Report Name Used for the Exported Output File Name

When you run a report in an export format such as PDF, delimited text (CSV), Microsoft Excel (XLS), the IBM Cognos report name is now used as the exported file name. This allows you to save the report output using the same name as the original report. For more information, see “Run a Report” on page 45.
Enhancements for SAP BW Data Sources

IBM Cognos Business Intelligence includes prompting for SAP BW variables and time-dependent hierarchies when creating reports with SAP BW data sources.

Prompting for SAP BW Variables

SAP BW variables are parameters in a SAP BW query that your business warehouse administrator sets up. When the queries run, the SAP BW variables are filled with values. SAP BW variables are automatically exposed as prompts when you run a report or when you add data to a report.

Prompting for SAP BW variables in IBM Cognos BI has improved. You are prompted for optional variables when you initially drag and drop data items.

Time-dependent Hierarchies

Time-dependant hierarchies now automatically reflect hierarchy or structure changes. When a structure is imported into IBM Cognos Framework Manager, each SAP BW time hierarchy is depicted as an individual level. IBM Cognos Report Studio users can use these structures to report on and compare levels that are valid for a specific time period.

Column Titles When Drilling up or Down

When you drill down or up, you can now specify whether the column title shows the member caption value or the level label value.

For more information, see “Create a Drill-up and Drill-down Report” on page 308.

Custom Properties for Prompts

You can now customize the text that instructs your report users how to use report prompts.

For example, you can now change the default text that appears above a search text box from Keywords to the text of your choice. For more information, see “Customize Prompt Text” on page 323.

Simplified Calculations

To add a calculation, you now choose to add either a Query Calculation or Layout Calculation object from the toolbox tab.

For more information, see “Using Relational Calculations” on page 248 and “Using Dimensional Calculations” on page 302.

Function Description Improvements:

The functions that you can use to create calculations now include improved descriptions and more examples.

The descriptions and examples appear in the Report Studio user interface and in this user guide. For more information about functions, see Appendix E, “Using the expression editor,” on page 555.
New Vendor-specific Functions:

Functions that are specific to Postgres, Vertica, Netezza, Paraccel, MySQL, and Greenplum now appear in the expression editor in the Vendor Specific Functions folder.

For more information, see Appendix F, “Using the expression editor,” on page 555.

New Report Style

IBM Cognos Business Intelligence includes a new default report style with updated colors and gradients.

If your report uses a custom report template, your report will appear the same in this version of IBM Cognos Report Studio as it did in previous versions. By default, new reports appear in the new report style. For more information, see “Create and Modify Report and Object Styles” on page 350.

If you want to create new reports with the custom template style, you can edit the default template for new reports to match your custom template. For more information about custom templates, see Chapter 20, “Creating Report Templates,” on page 451.

List Enhancements

Report Studio includes enhancements to aggregate summaries, grouping, and hiding columns in lists.

You can now specify whether to automatically include an overall aggregate summary in the list footer. For more information, see Chapter 4, “Lists,” on page 63.

You can also specify whether grouping a column automatically makes it the first column in the list. For more information, see “Group Relational Data” on page 213.

You enable these behaviors in the IBM Cognos Report Studio options, by selecting the Group and automatic summary behavior for lists check box.

A new Render property allows you to hide columns from the output of list reports. For more information, see “Hide Columns in List Reports” on page 64.

Accessible Report Output

IBM Cognos Report Studio now includes features that allow you to create reports that are more accessible to people who have a physical disability, such as restricted mobility or limited vision.

Accessible reports contain features that allow users with disabilities to access report content using assistive technologies, such as screen readers. You can

• add alternative text for non-text objects, such as images, charts, and maps.
• add summary text for crosstabs, lists, repeater tables, and tables.
• specify whether table cells are table headers.
• use command keys, or shortcut keys, to navigate through IBM Cognos Business Insight Advanced.
• use a screen-reader software with a digital speech synthesizer to listen to what is displayed on the screen.
• set your system’s display settings for high-contrast displays.
In addition, the documentation now includes alternate text for all graphics so that screen readers can interpret graphics.

**Braces and Parentheses are Breakable**
When you run a report in PDF format, braces {} and parentheses () no longer stay on the same line as the text before them.

For example, **Products(2012)** may now break to a new line between **Products** and **(2012)**.

### Changed Features in Version 10.1.0
Listed below are changes to features since the last release.

**Report Studio Express Authoring Mode Replaced by Business Insight Advanced**
The two authoring modes in IBM Cognos Report Studio, Professional and Express, no longer exist.

Report Studio is still the tool that professional report authors use to create advanced and managed reports. The Express authoring mode is now replaced by IBM Cognos Business Insight Advanced. Business Insight Advanced is part of a new report consumption experience that provides an integrated business intelligence experience for business users. It offers much greater capability, such as full support for list reports, charts, and relational data sources. It offers an entirely different user experience that is designed for data exploration.

For more information, see the Business Insight Advanced User Guide.

**Order of Drawing Bars in Bar Charts**
In a horizontal bar chart, the new default chart type draws bars from bottom to top. Legacy charts draw the bars or columns in the reverse order.

To control the order of the bars, use the property for the category axis.

**Line break rules in PDFs have changed**
Line break rules for word wrapping in PDFs have changed slightly from the last release. Brackets and parenthesis such as [ ] and ( ) are now treated as breakable. For example, in past releases, the following string would be treated as a single non-breaking string.

\[\text{Boilers(Steam)}\]

In this release, "Boilers" and "(Steam)" are treated as separate strings which might result in the following when the string occurs at the end of a line.

\[\text{Boilers} \]

\[\text{(Steam)}\]

### Removed Features in Version 10.1.0
Listed below are features that are removed since the last release.

**Analyze with Analysis Studio**
The ability to open a report with IBM Cognos Analysis Studio from within IBM Cognos Report Studio (Tools or right-click, Analyze) was removed in this release.
Conformed Dimensions on SAP BW Data Sources
Support for conformed dimensions generated by IBM Cognos Framework Manager for SAP BW data sources was removed in this release.

The Microsoft Excel 2000 spreadsheet software and Excel 2000 Single Sheet spreadsheet software report outputs are no longer supported in this release.
Chapter 2. Getting started with Report Studio

IBM Cognos Report Studio is a Web-based report authoring tool that professional report authors and developers use to build sophisticated, multiple-page, multiple-query reports against multiple databases. With Report Studio, you can create any reports that your organization requires, such as invoices, statements, and weekly sales and inventory reports.

Use Report Studio for reports that
- are intended for a wide audience
- exist long enough to require maintenance for changing requirements and data
- require detailed control over the appearance

Your reports can contain any number of report objects, such as charts, crosstabs, and lists, as well as non-BI components such as images, logos, and live embedded applications that you can link to other information.

Building IBM Cognos Business Intelligence Applications

You use the IBM Cognos Business Intelligence components to build reporting and analysis applications.

The lifetime of an IBM Cognos Business Intelligence application can be months, or even years. During that time, data may change and new requirements appear. As the underlying data changes, authors must modify existing content and develop new content. Administrators must also update models and data sources over time. For more information about using data sources, see the IBM Cognos Business Intelligence Administration and Security Guide and the IBM Cognos Framework Manager User Guide.

In a working application, the technical and security infrastructure and the portal are in place, as well as processes for change management, data control, and so on. For information about the workflow associated with creating IBM Cognos BI content, see the IBM Cognos Architecture and Deployment Guide. For additional information, see the IBM Cognos Solutions Implementation Methodology toolkit, which includes implementation roadmaps and supporting documents. Information about the toolkit is available on the IBM Cognos Customer Center (http://www.ibm.com/software/data/cognos/customercenter/).

The following graphic provides an overview for how to use IBM Cognos BI to build applications across all of your IBM Cognos BI components.
1. Locate and prepare data sources and models

IBM Cognos BI can report from a wide variety of data sources, both relational and dimensional. Database connections are created in the Web administration interface, and are used for modeling, for authoring, and for running the application.

To use data for authoring and viewing, the business intelligence studios need a subset of a model of the metadata (called a package). The metadata may need extensive modeling in Framework Manager.

2. Build and publish the content

Reports, scorecards, analyses, workspaces, and more are created in the business intelligence studios of IBM Cognos BI. Which studio you use depends on the content, lifespan, and audience of the report, and whether the data is modeled dimensionally or relationally. For example, self-service reporting and analysis are done through IBM Cognos Workspace Advanced, IBM Cognos Query Studio, and IBM Cognos Analysis Studio, and scheduled reports are created in IBM Cognos Report Studio. Report Studio reports and scorecards are usually prepared for a wider audience, published to IBM Cognos Connection or another portal, and scheduled there for bursting, distribution, and so on. You can also use Report Studio to prepare templates for self-service reporting.

3. Deliver and view the information

You deliver content from the IBM Cognos portal or other supported portals, and view information that has been saved to portals, or delivered by other mechanisms. You can also run reports, analyses, scorecards, and more from within the business intelligence studio in which they were created.

For information about tuning and performance, see the IBM Cognos Business Intelligence Administration and Security Guide and [http://www.ibm.com](http://www.ibm.com)

---

**Relational and dimensional reporting styles**

You can create reports in IBM Cognos Report Studio using either a relational reporting style or a dimensional reporting style.

The Report Studio tools and query language are the same for both styles. However, it is important to choose a reporting style to ensure that you are making the most of your data and to avoid mixing dimensional and relational concepts.

**How to choose a reporting style**

When authoring a report, first choose your preferred reporting style for working with data: relational or dimensional. You can choose a reporting style from your viewpoint:

- If you think about your data as tables and columns, you have a relational viewpoint and should use a relational reporting style.
- If you think about your data as a number of dimensions intersecting at cells, you have a dimensional viewpoint and should use a dimensional reporting style.

**Relational reporting style**

The relational reporting style consists of lists. You focus the data with filters and summarize with header and footer summaries.

- If your data is purely relational, then only query subjects and query items appear in the Source tab, and you must use the relational reporting style.
If your data is dimensional, then dimensions appear in the Source tab, and you can still use a relational reporting style, but instead of query items (columns) and query subjects (tables), you use measures, levels, and level properties.

To see an example of relational style reporting with dimensional data, see the Manager Profile sample report in the GO Data Warehouse (analysis) package.

The relational reporting style is similar to report authoring in IBM Cognos Query Studio.

**Dimensional reporting style**

The dimensional reporting style consists of measures and members from different hierarchies arranged in a crosstab with cell values at the intersections. You focus the data with set expressions that navigate from specific members in the hierarchy and summarize with set summaries.

To see an example of dimensional style reporting with dimensional data, see the GO Balance Sheet as at Dec 31, 2012 sample report in the GO Data Warehouse (analysis) package.

The dimensional reporting style is similar to report authoring in IBM Cognos Analysis Studio.

**Guidelines for each reporting style**

This user guide is divided into relational and dimensional reporting sections so that you can follow the best practices for using this product with the reporting style that you have chosen. The following table outlines the best practices for both reporting styles.

<table>
<thead>
<tr>
<th>Item</th>
<th>Relational reporting</th>
<th>Dimensional reporting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report type</td>
<td>Lists, Crosstabs pivoted from lists, Charts created from lists, Maps created from scratch or from lists</td>
<td>Crosstabs, Charts created from scratch or from crosstabs, Maps created from scratch or from crosstabs</td>
</tr>
<tr>
<td>Item</td>
<td>Relational reporting</td>
<td>Dimensional reporting</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Package tree</td>
<td>In the <strong>Source</strong> tab, click the view members tree button.</td>
<td>In the <strong>Source</strong> tab, click the view members tree button.</td>
</tr>
<tr>
<td></td>
<td>If you are using dimensional data, hide members in the source tree by right-clicking the tree, clicking <strong>Package Tree Settings</strong>, and clearing the check boxes for <strong>Members</strong> and <strong>Members for each level</strong>.</td>
<td>Ensure that the source tree tool bar is set to <strong>Create sets for members</strong>:</td>
</tr>
<tr>
<td></td>
<td>The view will include the following:</td>
<td>The view will include the following:</td>
</tr>
<tr>
<td></td>
<td>package</td>
<td>package</td>
</tr>
<tr>
<td></td>
<td>folder</td>
<td>folder</td>
</tr>
<tr>
<td></td>
<td>namespace</td>
<td>measure</td>
</tr>
<tr>
<td></td>
<td>query subject</td>
<td>member</td>
</tr>
<tr>
<td></td>
<td>query item</td>
<td>named set</td>
</tr>
<tr>
<td></td>
<td>measure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>level</td>
<td></td>
</tr>
<tr>
<td>Inserting data</td>
<td>From the package tree, add query items or levels to the report.</td>
<td>From the package tree, add sets of members to the report.</td>
</tr>
<tr>
<td></td>
<td>Avoid using hierarchies directly in the report.</td>
<td>You can also choose to insert just the member, just the children of the member, or the member and its children.</td>
</tr>
<tr>
<td></td>
<td>Avoid using named sets.</td>
<td></td>
</tr>
<tr>
<td>Calculating data</td>
<td>Add query calculations using only relational and common constructs.</td>
<td>Add query calculations using only dimensional and common constructs.</td>
</tr>
<tr>
<td></td>
<td>Choose only <strong>Other expression</strong>.</td>
<td>Use <strong>Other expression</strong> only for value calculations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avoid <strong>Common functions</strong> that are marked with an exclamation mark (!) because they have limited support.</td>
</tr>
<tr>
<td>Summarizing data</td>
<td>Headers and footers in lists</td>
<td>Member set summaries; for example, aggregate within set</td>
</tr>
<tr>
<td></td>
<td>Crosstab member summaries; for example, aggregate within detail</td>
<td></td>
</tr>
<tr>
<td>Focusing data</td>
<td>Add detail or summary filters to view only the data you want to see. For example, add the <strong>Quarter</strong> query item to a list and filter by Q3.</td>
<td>Add only the relevant members to an edge of the crosstab or to the context filter. For example, only add the Q3 member to your report. Use a set expression such as <strong>Topcount</strong> or <strong>Filter</strong>.</td>
</tr>
<tr>
<td>Item</td>
<td>Relational reporting</td>
<td>Dimensional reporting</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Drilling</td>
<td>Drilling through by value</td>
<td>Drilling through by member</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drilling up and down</td>
</tr>
<tr>
<td>Page and section breaks</td>
<td>Simple page breaks</td>
<td>Simple page breaks</td>
</tr>
<tr>
<td></td>
<td>Page sets</td>
<td>Page sets</td>
</tr>
<tr>
<td></td>
<td>Sections</td>
<td>Page layers</td>
</tr>
<tr>
<td></td>
<td>Master detail relationships using</td>
<td>Master detail relationships using</td>
</tr>
<tr>
<td></td>
<td>parameterized filters</td>
<td>parameterized edge or slicer expressions</td>
</tr>
</tbody>
</table>

The User Interface

The IBM Cognos Report Studio user interface has two panes, an explorer bar, and a work area to help you create reports.

We recommend that you use a screen resolution of at least 1024 by 768 pixels.

Content Pane

The content pane contains objects that you can add to a report. You add objects to a report by dragging them to the work area. The content pane contains these tabs:
• The **Source** tab contains items from the package selected for the report, such as data items and calculations.

• The **Data Items** tab describes the queries created in the report.

• The **Toolbox** tab contains a variety of objects that you can add to the report, such as text and graphics.

• The **Search** tab contains the results when you perform a search for members. You can insert the members found in a search directly into a report.

• If you are authoring an active report, the **Active Report Controls** tab shows the controls and active report variables inserted in a report and their relationships to one another. You can click a control in this tab to quickly locate the control in the report as well as set default values for variables.

• If you are authoring an active report, the **Active Report Variables** tab shows the active report variables defined in a report. Use this tab to create new variables and set default values.

**Properties Pane**

The **Properties** pane lists the properties that you can set for an object in a report.

You can obtain additional information about a property by selecting it and pressing F1. For example, you can view the list of objects that use each property.

When you specify a value for a property, press Enter, click another property, or save the report to ensure that the value is saved.

**Tip:** To view a description of the currently selected property at the bottom of the pane, from the **View** menu, click **Property Descriptions**.

**Explorer Bar**

Pause the pointer over the following buttons on the Explorer bar to work with different parts of a report:

• the page explorer button
  You use Page Explorer to view or create new report pages and prompt pages or to create and modify classes.

• the query explorer button
  You use Query Explorer to create or modify queries in relational reporting or dimensional reporting and to perform complex tasks, such as defining union joins and writing SQL statements.

• the condition explorer button
  You use Condition Explorer to work with variables to define conditions in a report.
Page Layers Area

Use the Page layers area to create sections, or page breaks, in a report to show values for each member on a separate page. For example, you can drag Northern Europe sales territory from the Source tab to the Page layers area. The report is broken into a separate page for each territory within northern Europe. Each page’s context appears in the report header.

For more information, see “Create Page Layers” on page 449.

Context Filter Area

When working with dimensional data, use the Context filter area to filter your report to show values, or context, for only a specific data item. This technique is also known as a slicer filter. For example, you can drag Sales Territory from the Source tab to the Context filter area. When you click a specific territory from the list, the values in the crosstab change to represent data for that territory.

For more information, see “Create a Context Filter” on page 279.

Visual Aids Button

The visual aids button provides the following options to help you when you are designing reports in the layout.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Boundary Lines</td>
<td>Shows all boundary lines around objects.</td>
</tr>
<tr>
<td>Show Repeating</td>
<td>Repeats objects when you insert them. For example, when you insert a data item in a crosstab, the data item appears in each row or in each column of the crosstab.</td>
</tr>
<tr>
<td>Show Page Header &amp; Footer</td>
<td>Shows the page header and page footer.</td>
</tr>
<tr>
<td>Show Drag &amp; Drop Padding</td>
<td>Shows drag-and-drop zone when the Padding property for an object is set to 0. If the Padding property is set to a value that is greater than the minimum padding that IBM Cognos Report Studio uses to show drag-and-drop zones, only the minimum padding is shown.</td>
</tr>
<tr>
<td>Show Hidden Objects</td>
<td>Shows objects for which the Box Type property was set to None or for which the Visible property was set to No.</td>
</tr>
<tr>
<td>Show Sorting</td>
<td>Shows the sorting icon for data items for which a sort order was specified. For more information about sorting data, see “Sorting Relational Data” on page 235 or “Sorting Dimensional Data” on page 281.</td>
</tr>
<tr>
<td>Show Grouping</td>
<td>Shows the grouping icon for grouped data items.</td>
</tr>
<tr>
<td>Show Source Type</td>
<td>Shows the icon for the source type of objects, such as layout calculation.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Show Data Item Type</td>
<td>Shows the icon for the type of data item, such as query item, member, or measure.</td>
</tr>
<tr>
<td>Show Drill-through Definitions</td>
<td>Shows data items for which the drill-through definition was defined as a hyperlink.</td>
</tr>
<tr>
<td>Show Table of Contents Entries</td>
<td>Shows table of contents entries inserted in the report.</td>
</tr>
<tr>
<td>Show Bookmarks</td>
<td>Shows bookmarks inserted in the report.</td>
</tr>
<tr>
<td>Show Master Detail Relationships</td>
<td>Shows master detail relationships defined in the report.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> Pausing the pointer over the master detail relationship icon shows the relationship.</td>
</tr>
<tr>
<td>Show No Data Contents Tab Control</td>
<td>Shows tabs if the data container's No Data Contents property is set to Yes.</td>
</tr>
<tr>
<td>Show Repeater and Singleton Containers</td>
<td>Shows repeater and singleton containers inserted in the report.</td>
</tr>
<tr>
<td>Show Interactive Object Controls</td>
<td>Shows controls in the top right corner of active report application objects. For more information about active reports, see Chapter 8, “Active Reports,” on page 171.</td>
</tr>
<tr>
<td>Show Container Selectors</td>
<td>Shows a small selector (three orange dots) in the top left corner of the following container objects and allows you to select all the objects within them: list, crosstab, repeater table, table of contents, table, and active report application objects.</td>
</tr>
<tr>
<td>Show Empty Extra Item Drop Zones</td>
<td>When you are creating a visualization, this visual aid shows empty drop zones for extra items that you can insert. The extra drop zones are displayed only when data items are inserted in all required drop zones.</td>
</tr>
</tbody>
</table>

**Work in design or structure view**

IBM Cognos Report Studio has two views in which you can author reports: **Page Design** view and **Page Structure** view. You can choose a report authoring view on the **View** menu.

Different options are available in each view, so you often need to use both views. For example, you must use both views to remove sections in relational reporting.

**Page design view**

**Page Design** view is the default view in IBM Cognos Report Studio. In this view, you can see what your report will look like after you run it.

**Page structure view**

**Page Structure** view displays an overview of all of the report objects in your report in a tree structure, which is organized by page.
When you add objects to a report, you usually work in the layout. An alternative view of the report is available.

---

**Report Layout and Queries**

All reports have two components: a layout component that defines the report appearance and a query component that defines report data. Understanding these components will help you design effective reports.

**Layout**

A layout is a set of pages that defines the appearance and formatting of a report.

When you design the layout of a report, you
- present the data in a meaningful way by using lists, crosstabs, charts, and maps
- add formatting, such as borders, color, images, and page numbers
- specify how the data flows from one page to the next

**Pages**

Pages are containers for the layout objects that you use to build a report. A page is made up of the following mandatory and optional components:
- page header (optional)
- page body (mandatory)
- page footer (optional)

When you run a report, the amount of data queried often exceeds one page. As a result, a page will repeat until all the data is shown. You have control over how data flows from one page to the next. For example, here are alternative representations of a report that contains a chart and a lengthy list.
Objects

You add layout objects to a page when you create a report. Below are objects that you will use often when building reports in IBM Cognos Report Studio:

- **list**
  Add a list to show data in rows and columns.

- **crosstab**
  Add a crosstab to show data in a grid with dimensions along the rows and columns and measures in the cells or intersection points.

- **chart**

- **map**

- **repeater**
  Add a repeater to show each instance of a certain column or data item in a separate frame.

- **text**

- **block**
  Add a block to hold text or other information. Blocks are often used to lay out horizontal bands of information.

- **table**

**Related tasks:**

Insert a Formatting Object in a Report” on page 336

In addition to text and images, the Toolbox tab contains other objects that you can add to the report layout.

Queries

Queries determine what data items appear in the report. Sometimes you want detailed rows of data, which you obtain by using a simple SELECT statement. Other times you must calculate totals or averages using summary functions and grouped columns or must apply filters to show only the data you want.
IBM Cognos Report Studio automatically creates the queries you need as you build reports. However, you can modify these queries or create your own custom queries to get the results you want.

Related concepts:
- "Working with Relational Queries" on page 236
- "Working with Dimensional Queries" on page 283

Queries specify what data appears in the report. In IBM Cognos Report Studio, you create and modify queries using Query Explorer.

Report Objects

You build reports by adding objects and manipulating them to obtain the results you want. To understand how to work with objects in IBM Cognos Report Studio, you must be familiar with the following concepts:
- object types
- objects as containers
- locking and unlocking objects
- hierarchy of objects

Object Types

In IBM Cognos Report Studio, layout objects are either inline or block. You can insert other objects on the same line as an inline object, but not on the same line as a block object. When you insert an object to the left or to the right of a block object, the object appears on the line above or below the block object, respectively. Examples of inline objects include graphics and text items. Examples of block objects include any report type (list, crosstab, chart, map, or repeater) and tables.

You can also use an object’s floating property to define how other objects flow around the object. For example, you can specify how text flows around an image.

Objects as Containers

Objects, such as tables, blocks, and any report frame, are containers in which you can insert other objects. For example, you can insert a list in one cell of a table and a chart in another.

You can also nest objects to create a sophisticated layout. For example, you can insert a table in a cell of another table.

Locking and Unlocking Objects

To manipulate the contents of some objects, you must first unlock the object. For example, you have a list that contains the column Product Name. You want to insert a graphic inside the Product Name column to show an image of each product. Unlocking the list allows you to insert the image object inside a list column.

Tip: From the Structure menu, click Lock Page Objects. Toggling this menu item locks and unlocks all layout objects in a report. However, this setting is not saved with the report.
Hierarchy of Objects

In IBM Cognos Report Studio, objects are organized hierarchically. For example, a list contains list columns, and each list column contains a text item, which is the name of the inserted data item.

The hierarchy of objects is useful to remember when you apply formatting because formatting is applied to the child objects of the object. For example, you can specify that all list column titles in a list have red as the background color. The formatting is automatically applied to any new columns you add to the list because the formatting is applied to the list and is therefore applied to the objects in the list. If you apply formatting to a specific object, it will override the same formatting specified for the parent object.

Related concepts:

This appendix contains definitions of the objects and properties found in IBM Cognos Report Studio. They are available contextually, by pressing F1 when an object or property is active in the Report Studio authoring environment.

Find Objects in a Report

You can quickly locate specific objects in a report by using Find.

Procedure

1. From the Edit menu, click Find.
2. Click Find what and choose the type of object to find.
3. In the Options box, specify the search criteria.
   For example, if you want to find the objects that reference a specific data item, you must specify the query that contains the data item and the data item name.
4. To search for objects that are hidden in the report, in the View box, select the Show Hidden Objects check box.
5. In the View box, under Page view, select whether you want to do the search in the Page Design or Page Structure view.
6. Click Find Next.

Results

The first object that meets the search criteria is selected in the report. Continue clicking Find Next to show all other objects that meet the search criteria.

Note: In some cases, such as searching for a data item that is used to define a conditional style, IBM Cognos Report Studio cannot directly select the object in the report. Instead, Report Studio selects the object that uses the object for which you are searching.

Related concepts:

“Work in design or structure view” on page 28
IBM Cognos Report Studio has two views in which you can author reports: Page Design view and Page Structure view. You can choose a report authoring view on the View menu.

Options

You can set various options that control the appearance and behavior of IBM Cognos Report Studio (Tools, Options).
### View Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Windows skin</td>
<td>Replaces the current appearance of the interface with the display scheme specified by the Windows settings.</td>
</tr>
<tr>
<td>Show startup dialog</td>
<td>Shows the Welcome dialog box at startup.</td>
</tr>
<tr>
<td>Reuse IBM Cognos Viewer window</td>
<td>Reuses the same IBM Cognos Viewer window when you rerun a report without first closing the window.</td>
</tr>
<tr>
<td>Resize IBM Cognos Viewer window</td>
<td>Maximizes the IBM Cognos Viewer window when you run a report.</td>
</tr>
<tr>
<td>Enable animation</td>
<td>Animates the appearance of dialog boxes, menus, and panes.</td>
</tr>
<tr>
<td>Show object and query name dialog on insert</td>
<td>Specifies whether to show the Object and Query Name dialog box when you add a list, crosstab, repeater table, repeater, or singleton from the Toolbox tab. Use this dialog box to quickly specify a name for the data container and a name for the query that is associated with the data container. Or you can link an existing query to the data container. Tip: The Insert Chart and Choose Map dialog boxes also contain fields to specify a name and query when you add a chart or map to a report.</td>
</tr>
<tr>
<td>Window startup size</td>
<td>Specifies the size of the Report Studio window at startup.</td>
</tr>
<tr>
<td>Position pane on the right (requires restart)</td>
<td>Moves the content and Properties panes to the right of the work area. This check box is cleared by default. For the change to take effect, you must close and then restart Report Studio.</td>
</tr>
<tr>
<td>Show rich tooltips (requires restart)</td>
<td>Specifies whether to show descriptive tooltips when you hover over a button in the toolbar. This check box is selected by default. For the change to take effect, you must close and then restart Report Studio.</td>
</tr>
<tr>
<td>Display report preview</td>
<td>Shows a preview of the report when you open or save a report, within the Open, Save, and Save As dialog boxes.</td>
</tr>
<tr>
<td>Start page view</td>
<td>Enables you to start Report Studio in Page Design or Page Structure view. For the change to take effect, you must close and then restart Report Studio.</td>
</tr>
</tbody>
</table>

### Edit Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrap text in editors</td>
<td>Automatically wraps text in all editors where you can define expressions.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Automatically populate values list</td>
<td>When building expressions in the expression editor, automatically shows values when you browse the data of a data item.</td>
</tr>
<tr>
<td>Automatically validate expressions</td>
<td>Automatically validates calculations, such as filters, created in the expression editor. For more information, see “Using Relational Calculations” on page 248 or “Using Dimensional Calculations” on page 302.</td>
</tr>
<tr>
<td>In-place edit</td>
<td>Enables the editing of text in place when double-clicking.</td>
</tr>
<tr>
<td>Use chart flyouts in Design View</td>
<td>When in Page Design view, shows the chart drop zones as flyouts that appear only when you hover your mouse over the chart. When cleared, the chart drop zones are always displayed.</td>
</tr>
<tr>
<td></td>
<td>By default, this option is not enabled.</td>
</tr>
<tr>
<td>Drop replace on crosstab and chart nodes</td>
<td>Specifies what the existing members are replaced with when you drag a new member onto a report.</td>
</tr>
<tr>
<td>Double-click on member action</td>
<td>When working with dimensional data, specifies what happens when you double-click a member data item.</td>
</tr>
<tr>
<td></td>
<td>By default, you drill down or up on the item that you double-click.</td>
</tr>
<tr>
<td>Layout dimensions</td>
<td>Specifies the width and height of the area where you will create reports.</td>
</tr>
</tbody>
</table>

**Report Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias member unique names</td>
<td>When working with a dimensional data source, creates an alias when you add a member to the report or to an expression.</td>
</tr>
<tr>
<td>Delete unreferenced query objects</td>
<td>Automatically deletes query objects linked to another object. For example, if you delete a list, the query linked to the list is deleted as well.</td>
</tr>
<tr>
<td>Delete unreferenced conditional styles and palettes</td>
<td>Automatically deletes conditional styles or palettes when the last data item that refers to the conditional style or palette is also deleted.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Always create extended data items for new reports</strong></td>
<td>When working with dimensional data sources, determines whether Report Studio creates extended data items or expression-based data items for new reports. Expression-based data items allow you to view or edit the expression by double-clicking the Expression property for the item. If you are working with a dimensional data source and this option is not selected, the appropriate dimensional data item is added when you insert items such as members, levels, or calculated members. For example, if you insert a level, a level set is created. This makes it easier to work with dimensional data items because Report Studio knows the data item type of the items that are in the report.</td>
</tr>
<tr>
<td><strong>Table Style inheritance</strong></td>
<td>When a table style is applied to a list or crosstab, specifies whether new objects inserted in the list or crosstab should inherit the style. For more information, see “Apply a Table Style” on page 340.</td>
</tr>
<tr>
<td><strong>Automatic group and summary behavior for lists</strong></td>
<td>When working with lists, automatically adds an overall aggregate summary in the list footer and a summary for any groups in the list. When grouping a column, automatically makes it the first column in the list.</td>
</tr>
<tr>
<td><strong>Automatically create crosstab headers for sets</strong></td>
<td>When adding sets in a crosstab that uses a dimensional data source, automatically adds header labels on new columns and rows. The headers help consumers of the report to understand where the data is in the hierarchy.</td>
</tr>
<tr>
<td><strong>Limit on inserted individual members</strong></td>
<td>When working with a dimensional data source, limits the number of child members that are inserted. For example, you specify 3 for this option and, in the toolbar, you specify the option to insert children when you drag a member to a data container. You then drag the Camping Equipment member to the rows of a crosstab. What you see as rows are the child members Cooking Gear, Tents, and Sleeping Bags and a row named Others (Camping Equipment) for the remaining child members of Camping Equipment.</td>
</tr>
</tbody>
</table>

### Advanced Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use operating system clipboard</strong></td>
<td>Uses the Microsoft Windows (or other operating system) clipboard instead of the internal Report Studio clipboard.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use legacy chart authoring</strong></td>
<td>Enables you to create new reports using the legacy charts instead of the default charts and disables the automatic upgrade of charts in existing reports to the current default charts. Select this check box if you do not want to upgrade the charts in your existing reports.</td>
</tr>
<tr>
<td><strong>Disable previews</strong></td>
<td>When editing properties such as date and number formatting, does not display a sample of the formatting that is applied to data. When this option is not enabled, either sample data or data from your data source is displayed with formatting options applied. Additionally, when in <strong>Page Design</strong> view, displays a static image for a chart instead of updating the chart with a preview of your chart, using simulated data.</td>
</tr>
<tr>
<td><strong>Override 10.x styles with 8.x styles on new reports</strong></td>
<td>Specifies whether to use the Version 8.x report styles by default when creating new reports. For more information about report styles, see <a href="#">&quot;Create and Modify Report and Object Styles&quot; on page 350</a>.</td>
</tr>
<tr>
<td><strong>Active Report validation of visualizations</strong></td>
<td>Specifies whether to validate visualizations in an active report when the report is run from Report Studio.</td>
</tr>
<tr>
<td><strong>Map feature display limit</strong></td>
<td>When working with maps, specifies the maximum number of features that can appear in a map.</td>
</tr>
<tr>
<td><strong>Member display count limit (in source tree)</strong></td>
<td>When working with dimensional data, specifies the maximum number of members that can appear in the <strong>Source</strong> tab before you must perform a search.</td>
</tr>
</tbody>
</table>

**Related concepts:**

["Recommendation - Use Member Unique Name (MUN) Aliases" on page 547](#). If you are working with a dimensional data source, use MUN aliases to simplify building reports and expressions. In addition, you can tie the MUN back to the member in the package.

---

### Web Browser Settings

IBM Cognos Report Studio can be used in the Microsoft Internet Explorer and Mozilla Firefox Web browsers.

To review an up-to-date list of the environments that are supported by IBM Cognos products, including information about operating systems, patches, browsers, web servers, directory servers, database servers, and application servers, visit the [IBM Cognos Customer Center](http://www.ibm.com/software/data/cognos/customercenter/).
IBM Cognos Business Intelligence uses the default browser configurations provided by Internet Explorer and Firefox. Additional required settings are specific to the browser.

For Internet Explorer, the following settings are required:
- Allow Cookies
- Active Scripting
- Allow META REFRESH
- Run ActiveX controls and plug-ins
- Script ActiveX controls marked safe for scripting
- Binary and script behaviors
- Allow programmatic clipboard access
- Userdata persistence
- Enable pop-ups for the IBM Cognos BI server

For Firefox, the following settings are required:
- Allow Cookies
- Enable Java™
- Enable JavaScript
- Load Images
- Enable pop-ups for the IBM Cognos BI server

Report Studio uses the native Microsoft Internet Explorer XML support, which is a component of the browser. ActiveX support must be enabled because Microsoft implements XML using ActiveX. IBM Cognos BI does not provide or download ActiveX controls. Only the ActiveX controls that are installed as part of Internet Explorer are enabled through this configuration.

For more information about the Web browser configuration and cookies used by IBM Cognos BI, see the IBM Cognos Business Intelligence *Installation and Configuration Guide*.

---

**The IBM Cognos Software Development Kit**

When you create a report, you are creating a report specification.

A report specification is an XML file that you can view (Tools, Show Specification). In addition, you can view the specification for a selected object with the Show Specification (Selection) menu option.

You can also programmatically create or modify reports by using an editing tool to work with report specifications. You then use the IBM Cognos Software Development Kit to implement the reports in your IBM Cognos Business Intelligence environment. This is useful if, for example, you must make the same modification in many reports. Rather than opening each report and making the change, you can automate the process using the Software Development Kit, thereby saving you time. For more information about the Software Development Kit, contact your local sales office.
Tips

- When you are viewing the report specification, you cannot modify or copy parts of it.
- You can also modify the XML code in a report specification by saving the report specification on your computer.
Chapter 3. Creating a report

When you create a report, you are actually creating a report specification. The report specification defines the queries and prompts that are used to retrieve data and the layouts and styles used to present the data. For simplicity, the report specification is named by the same name as the report.

Specify the Data Package

Specify the package that will provide items for the report.

The packages that you use to generate reports are based on models that are created in the modeling tool, IBM Cognos Framework Manager. A model is a set of related objects, such as query subjects, dimensions, filters, and calculations. When you open a package in IBM Cognos Business Intelligence, these model objects are visible in the left frame.

Before you begin

The package must be previously created and published to the IBM Cognos Connection portal. For more information, see the IBM Cognos Framework Manager User Guide.

Important: If the package version changed since you opened IBM Cognos Report Studio, close and restart Report Studio to ensure that you are working with the latest version of the package.

Procedure

1. Open IBM Cognos Report Studio with the package you want to use.
2. In the Welcome dialog box, choose whether to open a new or existing report or template:
   - To create a new report or template, click Create a new report or template and choose a basic report layout.
   - To open an existing report or template, click Open an existing report or template and select a report.

Tip: You can specify a language other than the default language for your package by clicking on the ellipsis (...) button to the right of the Package field, clicking the Language ellipsis button in the Report Package dialog box, highlighting the desired language, and then clicking OK. As stated on the language selection dialog box, items such as separators and decimals may need to be manually updated for language-specific syntax rules as a result of your selection.

Results

Objects from the selected package, such as query items, appear on the Source tab.

Tip: You can later change packages.
Refresh the package

If the package that a report is using changed, refresh it to ensure that you are working with the latest content.

About this task

The contents of a package can change without the package version changing. For example, your modeler makes changes to a model and republishes the package with model versioning disabled.

Procedure

Refresh the package.

- To get the latest contents of the current package version being used, in the Source tab, click the refresh icon.
- To get the latest package version, close and restart your component.

Choose a Basic Report Layout

IBM Cognos Report Studio includes several basic report layouts that include report objects, such as lists, crosstabs, maps, repeaters, and headers and footers. You can also choose to start with a blank report or open an existing report.

Tip: The Budget vs. Actual sample report in the GO Data Warehouse (analysis) package is based on a basic report layout. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Procedure

1. From the File menu, click New.
2. If you want to change the package, click the ellipsis (...) button and click a different package.
3. Choose a basic report layout:
   - To create a report from a blank layout, double-click Blank.
   - To create a report using one of Report Studio’s predefined report layouts, double-click the layout.
   - To create a new report template that can be applied to reports created in IBM Cognos Query Studio or IBM Cognos Analysis Studio, double-click Report Template.
     For more information about creating report templates, see Chapter 20, “Creating Report Templates,” on page 451.
   - To create a financial report using the basic financial report layout, double-click Financial.
   - To create a new report using another report, double-click Existing, locate the report, and click Open.

Results

The basic report layout appears in the report page.
Inserting a data container from the toolbox tab

In addition to choosing a basic report layout when you create a report, you can insert data containers from the toolbox tab into a report in IBM Cognos Report Studio.

About this task

When you insert a data container from the Toolbox tab, you can specify a name for the data container and the query that you want to link to the container instead of accepting the default names that are generated by Report Studio.

Procedure

1. From the Toolbox tab, drag the data container that you want to insert in the report.

   For lists, crosstabs, repeaters, repeater tables, and singletons, the Object and Query Name window opens. For charts, the Insert Chart dialog box opens, and for maps, the Choose Map window opens.

   Tip: The Object and Query Name window opens when the Show object and query name dialog on insert option in the Tools menu, View Options tab, is selected.

2. Type a name for the data container and a name for the query to be created for the container.

   Tip: You can link the container to an existing query by clicking the query menu and choosing from the list of available queries.

Add Data

Select the data items you want to appear in your report.

For more information about adding data to a relational style report, see "Add Relational Data to a Report" on page 213. For more information about adding data to a dimensional style report, see "Add Dimensional Data to a Report" on page 253.

Procedure

From the Source tab, drag data items to the report object.

A flashing black bar indicates where you can drop a data item. Data items in the report appear on the Data Items tab.

Data Source Icons

Each object in the data source has a representative icon. You can insert all of the following objects in a report, except for packages and dimensions.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Package Icon]</td>
<td>Package, which contains the objects you can insert in a report.</td>
</tr>
<tr>
<td>![Namespace Icon]</td>
<td>Namespace, used to organize objects.</td>
</tr>
<tr>
<td>Icon</td>
<td>Object</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Query subject, which represents a table in the database.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>In relational data sources, query item, which represents a column of qualitative data in the database, such as product name or country or region.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>In dimensional data sources, level attribute, which represents a property of a level.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>A member is a unique item within a hierarchy. For example, Camping Equipment and 4 Man tent are members of the Products Hierarchy.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Dimension, which represents a broad grouping of descriptive data about a major aspect of a business, such as products, dates, or markets.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Hierarchy, which represents a collection of dimensional members organized into a tree structure.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Level, which is a set of members that have common attributes. For example, a geographical dimension might contain levels for country or region or city. Multiple levels can exist within a level hierarchy, beginning with the root level. The root level is the parent and rollup of all members in the first level. It is used to obtain a rollup of all values across the hierarchy and to provide a convenient point to start drilling. For example, a Years level hierarchy may contain the following levels: • Root level Years • First level Year • Second level Quarter • Third level Month</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Measure or fact, a query item that represents a column of quantitative data in the database, such as revenue or quantity.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Measures folder, which contains hierarchical measures.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Model filter.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Model calculation.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Folder, used to organize data items. You cannot import a folder into your report.</td>
</tr>
</tbody>
</table>

**Insert a Single Data Item**

You can insert a single data item anywhere in your report using the singleton object. The singleton object retrieves only the first row value for that query. Inserting a single data item is useful when you want to show a value that is independent from the rest of the values in the report or when you want to insert some boilerplate text, such as a company name and address. For example, you can add the total revenue value in the header of each page in a report.

You can associate multiple singleton objects with a single query in relational reporting and dimensional reporting to optimize performance, such as when all the data items in the singleton are from the same database table. In addition, two or
more singletons can reference data items from the same query. This is useful when using a single query is more efficient to display a set of single values than using multiple queries.

You can also filter the data item in the singleton. For example, you can show the total revenue for only the year 2012.

Queries that are associated to a singleton object are not supported when producing report output in delimited text (CSV) format.

**Tip:** The Returns by Damage, Failed Orders and Complaints in 2012 sample report in the GO Data Warehouse (analysis) package includes a singleton. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Procedure**

1. From the **Toolbox** tab, drag **Singleton** to the report.
   
   An empty data container is created.

2. From the **Source** tab, drag a data item into the **Singleton** container.

   **Tip:** To create a singleton, you can also drag a data item anywhere in your report layout.

3. To change the query associated to the singleton object, in the **Properties** pane, double-click the **Query** property and make changes.

**Results**

When the report is run, the first row value for the data item is retrieved.

---

**Validate a Report**

Validate your report to ensure that it contains no errors.

When you open a report created in a previous version of IBM Cognos Business Intelligence, it is automatically upgraded and validated. For more information, see Chapter 22, “Upgrading Reports,” on page 461.

**Procedure**

1. From the **Tools** menu, click **Validate Report**.
   
   A message box appears indicating whether any errors were found in the report.

2. If you require more detail from the validation process, from the **Tools** menu, click **Validate Options** and do the following:
   
   • Click one of the following validation levels.

<table>
<thead>
<tr>
<th>Validation level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>Retrieves all errors returned from the query.</td>
</tr>
<tr>
<td>Warning</td>
<td>Retrieves all errors and warnings returned from the query. This is the default validation level.</td>
</tr>
</tbody>
</table>
### Validation level

<table>
<thead>
<tr>
<th>Key Transformation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In addition to errors and warnings, retrieves informational messages describing important transformation steps from the report specification to the native query sent to the data source. These messages can show the cause of errors and warnings returned from the query.</td>
</tr>
<tr>
<td>Information</td>
<td>Retrieves errors, warnings, key transformations, and other information related to query planning and execution.</td>
</tr>
</tbody>
</table>

**Tip:** Your administrator can change the default validation level. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

- Select the **Interactive data** check box to specify that no query optimization is to be used.

  The **Interactive data** check box controls how queries will be processed during validation.

  Clear the **Interactive data** check box to set the **Execution Optimization** property to **All Rows**.

  **Tip:** For more information about the **Execution Optimization** property, see Execution Optimization in Appendix G, “Report Studio Object and Property Reference,” on page 771.

  A message box appears indicating whether any errors were found in the report.

- To view messages that result from the conversion of your report to use the dynamic query mode instead of the compatible query mode, select the **Dynamic Query Migration** check box.

  For more information about using the dynamic query mode, see the IBM Cognos Business Intelligence Dynamic Query Guide. For more information about managing the migration from the compatible query mode to the dynamic query mode, see the IBM Cognos Lifecycle Manager Installation and User Guide.

- Revalidate your report.

3. If you encounter validation errors and want IBM Cognos Report Studio to identify incorrect objects in your report, from the **Tools** menu, click **Auto Correct**.

   Report Studio provides a list of such objects that you can remove individually to successfully run your report.

   In some cases, the information or error message is linked to the location of the issue in your report. To go to the location of the issue, click the message, and then click **Select**. If only warnings and information appear in the dialog box, these will disappear when you click **OK**.

### Save a Report

Save your report to preserve the modifications you made.

Reports are saved to the IBM Cognos Business Intelligence server. You can also save your report on your computer.
Procedure

1. From the File menu, click Save or click Save As to save a copy of the report under a different name.

2. If you are saving the report for the first time, specify where to save the report and type a file name.
   
   For information about setting up folders in IBM Cognos Connection for your reports, see the IBM Cognos Connection User Guide.

3. Click Save.

Run a Report

Run your report to see the data that is retrieved. Save time by validating it first to check for errors.

You can also run a report or a group of reports in IBM Cognos Connection. Furthermore, you can save report outputs to a file system. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

You can specify not to render a report page if it does not contain any data.

When you run a report in an export format such as PDF, delimited text (CSV), Microsoft Excel spreadsheet software (XLS), the IBM Cognos report name is used as the exported file name.

If you run a report that uses functions or features not supported by the data source, an error message appears. We recommend that you periodically test your reports while you author them in IBM Cognos Report Studio to ensure that you do not encounter multiple error messages when you run the report.

Procedure

1. Open a report.

2. If you want to clear parameter values stored on the IBM Cognos Business Intelligence server, from the File menu, click Clear Parameter Values.

   Parameter values stored by the IBM Cognos BI server include signon, validation, and prompt information. For example, if you define two data source connections in IBM Cognos Connection that point to the same data source, you are prompted to choose one when you run a report. This information is stored so that you are not prompted each time you run the report.

3. If you want to view only the tabular data, from the Run menu, click View Tabular Data.

   If the report contains multiple queries, you must first click an object, such as a list or crosstab, that uses the query for which you want to view the tabular data.

   Use this command to ensure that the right results appear. For example, you create a calculation and you want to ensure it is giving you the results you want.

   Tip: You can also view tabular data in Query Explorer, which is useful when you are building queries.

4. If you want to set run options, from the Run menu, click Run Options.
The default value is the value of the selected corresponding run option in IBM Cognos Connection.

**Note:** The run options that you set apply only to the current session. When you close Report Studio, the options return to the default settings.

5. Change any values for the current session.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Specify to change the default format from HTML.</td>
</tr>
<tr>
<td>Paper size</td>
<td>Specify only if the output format is PDF.</td>
</tr>
<tr>
<td>Paper orientation</td>
<td>Specify only if the output format is PDF.</td>
</tr>
<tr>
<td>Data mode</td>
<td>Specify how much data is returned:</td>
</tr>
<tr>
<td></td>
<td>• <strong>All Data</strong> returns all data.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Limited Data</strong> limits the amount of data returned based on design mode filters defined in the package.</td>
</tr>
<tr>
<td></td>
<td>• <strong>No Data</strong> returns artificial data instead of actual data from the data source.</td>
</tr>
<tr>
<td></td>
<td>For more information about design mode filters, see the Framework Manager User Guide.</td>
</tr>
<tr>
<td>Language</td>
<td>The content language sets the preferred language for the data, IBM Cognos Viewer, dates, and so on.</td>
</tr>
<tr>
<td>Rows per page</td>
<td>Specifies the number of rows to appear on each page.</td>
</tr>
<tr>
<td></td>
<td>A <strong>Rows Per Page</strong> property exists in the Properties pane for lists and crosstabs. If you set this property, the setting overrides the same-named run option. This property applies to both HTML and PDF outputs. For more information about this property, see &quot;Controlling the Rows Per Page for Multiple Containers in HTML and PDF&quot; on page 48.</td>
</tr>
<tr>
<td>Prompt</td>
<td>Select to be prompted for each prompt defined unless the prompt is defined in a report page.</td>
</tr>
<tr>
<td></td>
<td>If you clear the <strong>Prompt</strong> check box, you are prompted only if the report cannot run without user intervention. For example, if a report has a single parameterized filter that is optional, you are not prompted when you run the report.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Include accessibility features | Specifies whether accessibility features, such as alternate text on images and charts and summary text in tables, are included in the report output.  
This option is also used for conditional layouts in Report Studio. When the option is selected, you can specify that a crosstab is rendered as an accessible alternate to a chart.  
For more information about how to create accessible reports, see “Considerations to Improve Report Accessibility” on page 468. |
| Enable bidirectional support | Specifies whether to enable bidirectional support in the report output.  
Tip: You can also enable bidirectional support in Cognos Connection by modifying your user preferences (My Preferences link).  
If you enable bidirectional support in Cognos Connection, this run option will be automatically selected. |

6. From the Run menu, click one of the options to produce the report in the format you want.  
You can produce a report in HTML, PDF, CSV, various Microsoft Excel formats, and XML. You cannot produce a report in CSV format if you have more than one query defined in the report unless the additional queries are used for prompts.

Results

The report runs in IBM Cognos Viewer. Once the report has finished running, you can run the report again in the same format or in a different format. If you run the report again in CSV or XLS format, the report will appear in a new browser window.

The options available in IBM Cognos Viewer depend on the capabilities set by the administrator for each user. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

Related concepts:
“Running a Report Against a Dimensional Data Source” on page 53
You can cancel a report that is running against Microsoft SQL Server Analysis Services only during the initial portion of its execution. After this time, the report runs to completion.

Specify Not to Render a Page If It Does Not Contain Data
You can specify not to render a report page if the page does not contain any data when the report is run.

Procedure
1. In the report page, click a data container.
2. In the Properties pane, click the select ancestor button and click the data container type.
For example, if the data container is a list, click List.

**Tip:** You can also click the container selector (three orange dots) of the container to select it.

3. Set the **Render Page when Empty** property to **No**.
4. Repeat steps 1 to 3 for all other data containers in the page and any table of contents objects.

**Results**

When you run the report, if no data is produced in all data containers and table of contents objects in a page, the page is not rendered. The page is not rendered even if the page contains other objects, such as text items or images.

**Controlling the Rows Per Page for Multiple Containers in HTML and PDF**

If you have more than one data container in a report, such as a list and a crosstab, you can control how the report is rendered in HTML and PDF by setting the **Rows Per Page** property for each container.

For HTML output, the report property option **Page break by data container for interactive HTML** controls whether the default number of rows is rendered for each data container on each page.

IBM Cognos Business Intelligence uses the following rules when rendering reports in HTML and PDF:

- If the **Rows Per Page** property is not set for any of the data containers, 20 rows per page are rendered in HTML and each page is completely filled in PDF. The first data container is rendered until there is no more data, followed by the next container, and so on.

  **Tip:** The number of rows that appear on a PDF page depends on the font size set in the report.

- If the **Rows Per Page** property is set for each data container, the specified numbers of rows are rendered in HTML and PDF on each page until there is no more data.

- If the property is set for only some of the containers, the specified numbers of rows are rendered in HTML and PDF on each page until there is no more data. For the remaining containers, 20 rows per page are rendered on each page in HTML and each page is completely filled in PDF.

For example, you have two lists, List1 and List2. You set the **Rows Per Page** property to 5 for List1. When you run the report in HTML, the first page contains the first 5 rows from List1 followed by the first 15 rows of List2.

- If no data is returned for a data container, an empty container is rendered.

**Producing a Report in CSV Format**

IBM Cognos Business Intelligence can produce reports in CSV format so you can open them in other applications, such as Microsoft Excel spreadsheet software.

Reports saved in CSV format

- support Unicode data across many client operating systems
- are UTF-16 Little Endian data encoded
include a BOM (Byte Order Mark) at the beginning of the file
are tab-delimited
do not enclose strings in quotation marks
use a new line character to delimit rows

You can open reports saved in CSV format using a variety of spreadsheet software applications. By default, reports produced in CSV format will appear in the application associated with the .csv file type.

You cannot produce the following in CSV format:
- maps
- charts that do not have at least one category or series
- reports that have more than one query defined in the report, unless the additional queries are used for prompts

In IBM Cognos Connection, you can configure the CSV output to suit your environment. For example, you can specify the character used to delimit fields. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

Producing a Report in Microsoft Excel Format

You can export your report output to several different Microsoft Excel spreadsheet software formats.


The output is similar to other Excel formats, with the following exceptions:
- Charts are rendered as static images.
- Row height can change in the rendered report to achieve greater fidelity.
- Column widths that are explicitly specified in reports are ignored in Microsoft Excel 2007.
- Merged cells are used to improve the appearance of reports.
- The default size of worksheets is 65,536 rows by 256 columns.

Your IBM Cognos administrator can enable larger worksheets and change the maximum number of rows in a worksheet, up to a maximum of 16,384 columns by 1,048,576 rows, by using advanced server properties. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

Excel 2007 Data provides data for use in Microsoft Excel version 2007. These reports only contain minimal formatting. Default data formatting is applied to the data based on data type and assumes that each column has a single data type.

The output is similar to other Excel formats, with the following exceptions:
The generated output includes only the first list query in the report. If a report contains multiple queries and the first query is a multi-dimensional query for a crosstab or for a chart, an error message is displayed when the report runs.

Nested frames and master-detail links are not supported.

Cells in the Microsoft Excel file have a default width and height. You must adjust the column width and height if the data is larger than the default size.

Style specifications are not rendered, including color, background color, and fonts.

Borders are not rendered.

User-specified data formatting in the report specification are not applied, including exception highlighting and color rules for negative numbers.

**Excel 2002** provides fully formatted reports for use in Microsoft Excel versions earlier than 2007. Excel 2002 format also offers the following benefits:

- Spreadsheets are contained in a single file for reliable spreadsheet navigation.
- The maximum size of worksheets is 65,536 rows by 256 columns.

**Related concepts:**

- “Larger worksheet sizes for Microsoft Excel 2007 report outputs” on page 10
- IBM Cognos Business Intelligence now supports a larger worksheet size for report outputs exported in Microsoft Excel 2007 format.

There are limitations when producing reports in Microsoft Excel format.

**Option to group repeating cells in reports produced in Excel 2007 format**

When you produce reports in Excel 2007 format, you can specify whether repeating cells are grouped, or merged, into a single cell.

Report outputs to Excel 2007 format are easier to read and look more like other output formats when repeating values are grouped. If further analysis of the data is required within Excel or if the outputs are used to provide data to another tool, it is often preferable for repeating values to be populated in each row or column to which they apply.

By default, repeating cells are merged in Excel 2007 output. For example, Product line is a grouped column in a list. The values for Product line, such as Camping Equipment and Golf Equipment, appear once in a merged cell in Excel output. When repeating cells are not grouped, the values for Product line appear in each repeating cell. The option to merge repeating cells in Excel output is controlled by selecting or clearing the **Group repeating cells when exporting to Excel** check box in the **Report Properties** dialog box.

**Crosstabs**

In general, grouped data item values appear in each repeating cell. For example, a crosstab contains Product line and Product type as rows. When repeating cells are ungrouped, the label for each product line value is rendered in each repeating row cell.

If a cell on a column edge spans multiple worksheet rows, the cell label is not repeated in all rows. If a cell on a row edge spans multiple columns, the cell label value is not repeated in multiple worksheet columns. For example, a crosstab contains Product line, Product type, and Product as rows. A summary is added for
Product line. The product line summary cell spans Product line, Product type, and Product. In Excel, repeating cells are produced for the summary row, but the label for the summary appears only in the first cell.

Repeating cells that are produced from crosstab headers follow this behavior. For example, Product line and Product type are inserted as rows in a crosstab. Headers named Product line and Product type are created, and each header spans two columns. In Excel, the header labels appear only in the first cell. The repeating cells are empty.

If a layout object, such as a table or image, is inserted in the crosstab corner, the size of the object might cause the crosstab corner column edge to span multiple rows and row edges to span multiple columns. When this situation occurs, column labels repeat only in the column span and row labels repeat only in the row span.

The following figure shows a crosstab that contains Year as columns and Order method type as rows. A three by three table is inserted in the crosstab corner. In Excel output, the size of the table produces repeating cells in the columns and in the rows. Year labels, such as 2010, repeat only in the columns and not in the rows. Order method labels, such as E-mail, repeat only in the rows.

Lists

In lists, grouped data values are repeated in ungrouped cells. For example, a list contains Product line and Product type, and Product line is grouped. When repeating cells are ungrouped in Excel, the label for each product line value is rendered in the repeating cells.

Group header and list page header labels are not repeated. These labels appear only in the first cell, and repeating cells are empty.

Nested data containers, images, and charts

Nested data containers, images, and charts in a report can produce merged cells. For example, when a crosstab is inserted in a list, other columns in the list appear as merged cells in Excel 2007 output. The following rules are applied when merged cells are split.

1. If a cell is merged as a result of a data item grouping, then data values are repeated in the split cells.
2. If a cell is merged as a result of a nested data container, image, or chart, then data values are not repeated in split cells.
For example, a list contains Product line, Product type, Quantity, and an image as columns. The Product line column is grouped. The image produces merged cells in Excel output because its size spans more than one row. When the report is run with the option to ungroup repeating cells, Product line values are repeated, but Product type and Quantity values are not.

**Bookmarks**

When you produce Excel 2007 output with ungrouped repeating cells, clicking an entry in a table of contents brings you to the first row that contains that entry. For example, a report contains a list with a table of contents. Product line values are repeated in the list in Excel output. When you click Camping Equipment in the table of contents, you are brought to the first cell in the list that contains Camping Equipment.

**Related tasks:**
“Specify Report Properties” on page 350
You can change the default report property settings.

**Producing a Report in XML Format**

XML report outputs save the report data in a format that conforms to an internal schema, xmldata.xsd.

You can find this schema file in c10_location/bin.

This format consists of a dataset element, which contains a metadata element and a data element. The metadata element contains the data item information in item elements. The data element contains all the row and value elements.

You can create models from reports and other data that conform to the xmldata.xsd schema. This is useful if you want to use a report as a data source for another report, or if you use a database that cannot be read by IBM Cognos Framework Manager. In this case, export the data from the data source to an XML file, in conformance with the xmldata schema, and then open the XML file in Framework Manager.

For more information, see the Framework Manager User Guide.

You cannot produce the following in XML format:

• maps
• charts that do not have at least one category or series
• reports that have more than one query defined in the report, unless the additional queries are used for prompts
  If a report contains more than one data container, such as a crosstab and a list, and both containers use the same query, only the output for the list is produced. If a report contains multiple lists, only the output for the first list is produced. If a report contains multiple crosstabs and multiple lists, only the output for the first list is produced.

**Set PDF Page Options**

Set PDF page options, such as page orientation and paper size, to control how report pages appear in PDF. You can set PDF page options for individual report pages or for all report pages in a layout.
Tip: The PDF Page Properties sample report in the GO Sales (analysis) package includes PDF page options. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

You can also set PDF page options in IBM Cognos Connection. For more information, see the IBM Cognos Connection User Guide.

The paper size for PDF output can be a maximum of 200 x 200 inches or 500 x 500 centimeters. A larger paper size produces a blank page.

**Procedure**

1. To set PDF page options for all report pages in a layout, do the following:
   - From the **File** menu, click **PDF Page Setup** and set the page options.
2. To set PDF page options for an individual page, do the following:
   - Pause the pointer over the page explorer button and click the page.
   - In the work area, click anywhere on the page.
   - In the **Properties** pane, click the select ancestor button and click **Page**.
   - Double-click the **PDF Page Setup** property, select the **Override the page setup for this page** check box, and set the page options.

---

**View Lineage Information for a Data Item**

View lineage information of a data item to see what the item represents before you add it to a report.

Lineage information traces the metadata of an item back through the package and the data sources used by the package. Lineage also displays any data item filters that were added by the report author or that were defined in the data model. Viewing lineage information ensures that you add the correct data items to a report. For example, you can view the lineage information of a model calculation to see how it was created.

**Note:** Lineage is available only after your administrator has configured it. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide. In addition, lineage is not supported in reports that are not linked to packages.

You can use the lineage tool that comes with IBM Cognos Business Intelligence, or you can use another lineage tool by specifying the URL to the tool in IBM Cognos Administration. Note that if the URL source is secured, the source must be able to prompt users for a password because IBM Cognos BI does not pass security information. IBM Cognos BI also supports the IBM Metadata Workbench as a lineage tool. For more information about configuring other lineage tools, see the IBM Cognos Business Intelligence Administration and Security Guide.

Tip: The Customer Returns and Satisfaction sample report “Customer Returns and Satisfaction” on page 515 in the GO Data Warehouse (analysis) package includes lineage information. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

You cannot use lineage information to troubleshoot queries. For example, lineage information will not explain why a data item is double counted. Also, you cannot view lineage information when running a report from a mobile device.
**Before you begin**

Before you can access lineage information for a report, your administrator must configure lineage in IBM Cognos Administration. Also, the administrator must enable the lineage capability and grant read permission for you on the report.

**Note:** The IBM Cognos BI lineage tool shows lineage on a report at its highest level. The lineage does not change after you drill down on a report. Because the selection context used to launch lineage can be affected by drill-down operations, we recommend that you always launch lineage at the highest report level before drilling down on the report. Otherwise, the lineage may not start properly.

**Procedure**

From the **Source** tab, right-click the data item and click **Lineage**.

**Tip:** You can view lineage information for multiple data items at the same time by first **Ctrl**+clicking the items. The IBM Metadata Workbench does not support viewing lineage for multiple data items at once.

**Results**

The lineage tool opens showing the lineage information of the selected data item.

**The IBM Cognos Business Intelligence Lineage Tool**

The IBM Cognos Business Intelligence lineage tool includes two views: the business view and the technical view.

The business view displays high-level textual information that describes the data item and the package from which it comes. This information is taken from IBM Cognos Connection and the IBM Cognos Framework Manager model.

The technical view is a graphical representation of the lineage of the selected data item. The lineage traces the data item from the package to the data sources used by the package.

When you click an item, its properties appear below it. If you click an item in the **Package** area, you see the model properties of the item. If you click an item in the **Data Sources** area, you see the data source properties of the item.
You can also view lineage information in IBM Cognos Viewer after you run a report. For example, you can click a cell in a crosstab to see how the cell value was calculated. To view lineage information in IBM Cognos Viewer, right-click an item in the report and then click **Lineage**. If you or an administrator runs a saved report with the IBM Cognos BI lineage tool, both the business view and the technical view are visible. Report consumers can see only the business view. In addition to the **Package** and **Data Sources** areas, a **Report** area exists when looking at the technical view.

**Access the IBM WebSphere Business Glossary**

Business glossaries help you manage and share an enterprise vocabulary and classification system.

If you use the IBM WebSphere® Business Glossary, you can access the glossary from any of the following data objects:

- Query subject
- Query item
- Measure
- Dimension
- Hierarchy
- Level
- Property or attribute
- Top node member
- Member
- Level item

**Procedure**

Right-click the data item and click **Glossary**.
The IBM WebSphere Business Glossary appears.
Support for bidirectional languages

You can author reports that support bidirectional languages. You can specify base text direction, digit shaping, and container direction.

Arabic, Hebrew, Urdu, and Farsi are languages written from right to left, using the Arabic or Hebrew scripts. However, numbers in those languages, as well as embedded segments of Latin, Cyrillic, or Greek text, are written from left to right. Using the bidirectional settings in IBM Cognos Report Studio, you can control the direction in this type of text in reports.

Report formats

Bidirectional languages are supported for reports produced in HTML, PDF, and Microsoft Excel. IBM Cognos Active Report also supports bidirectional languages.

Enabling bidirectional support

To enable support for bidirectional languages when a report is run, in IBM Cognos Connection, users must open their My Preferences page and select the Enable bidirectional support check box.

To enable support for bidirectional languages when a report is run from Report Studio, select the Enable bidirectional support check box in the Run Options window.

Tip: If you enable bidirectional support in Cognos Connection and you then start Report Studio, the bidirectional support run option in Report Studio is selected. If you disable bidirectional support in Cognos Connection and you then start Report Studio, the run option in Report Studio is cleared.

Base text direction

Base text direction sets the text direction as left-to-right or right-to-left. You can specify the base text direction for any text object in a report. You specify the base text direction for text by clicking the Direction & Justification property for the text or for the object that contains the text, such as a list column.

A contextual option also exists in the Direction & Justification property that sets the text direction based on the first letter in the text.

For compound objects that contain text, you specify the base text direction of the text contained in the object by clicking the Contained Text Direction property for the object. Examples of compound objects are charts, prompts, and active reports.

Digit shaping

Digit shaping allows users to consume reports with numbers that they can read after they select the content language in IBM Cognos Connection. You can specify digit shaping at the following levels:

- Report
- Container (except for charts and maps)
- Text
- Number
You specify digit shaping for an object by clicking the Data Format property for the object. To specify digit shaping at the report level, click the Default Data Formats option in the Data menu.

**Tip:** Digit shaping has no impact on reports produced in Excel format, since the shaping of digits in Excel depends on Windows regional settings.

**Container direction**

Container direction sets the direction of container objects in a report as left to right or right to left. You specify container direction for an object by clicking the Direction & Justification property for the object.

**Tip:** Container direction is not supported in reports produced in Excel format. Excel spreadsheets do not natively support mirroring at the container level.

**Related concepts:**

"Digit shaping in charts and maps" on page 388

When working with bidirectional content, you cannot specify digit shaping at the chart or map level. You can specify digit shaping for the objects in charts and maps.

**Related tasks:**

"Run a Report" on page 45

Run your report to see the data that is retrieved. Save time by validating it first to check for errors.

"Specify text and container direction" on page 346

You can specify text and container direction by choosing any of these options.

"Set the Default Data Formats" on page 387

Set the default data properties for each type of data, including text, number, currency, percent, date, time, date/time, and time interval.

"Specify the Data Format for an Object" on page 388

Specify the format for a particular object if you are not getting the results you want.

---

**Reports for workspaces in Cognos Workspace**

When building a workspace in IBM Cognos Workspace, business users can insert an entire IBM Cognos Report Studio report or insert only individual objects from within a report.

To create effective reports for use in workspaces in Cognos Workspace:

- Ensure that all the objects and pages within the report have meaningful names.

  For some objects, such as data containers, Report Studio gives the object in a report a default name such as List 1, List2, and so on. In Cognos Workspace, these names appear in the **Content** tree. To help business users recognize the report objects, rename them to something more meaningful (**Properties** pane, **Name** property).

  If you do not want an object to be added in a workspace, clear the **Name** property for that object. The object will not appear in the **Content** tree. However, you cannot clear the name of data containers. Report Studio requires that data containers have names.
Note: Cognos Workspace does not support tables and blocks. Unsupported objects do not appear in the Content tree in Cognos Workspace, even if you specify a name for the object.

- Create a container report that includes all the report objects that workspace users might want to add in a report. Such a report is not meant to be run and does not need to show report objects in a proper layout. It is only used as a container for the various report parts. Each page of a report appears as a folder within the Cognos Workspace Content pane.

  For example, on one page named Prompts you can insert all the possible prompts that users might want to add to a workspace, and give each prompt a useful business name, such as Value Prompt for Region. On a second page named Charts, you can insert a number of charts useful for workspaces.

- Ensure that the titles and labels used in your report are understandable by business users.

- Ensure that you use report objects that are well suited for dashboards. They convey the most information in as little space as possible. For example, if you include a chart, use bullet charts, microcharts, line charts, or column and bar charts. Use conditional blocks to show business indicators.

- If your report includes charts, customize the font sizes so that the charts appear correctly in a dashboard. You may need to reduce the font sizes.

- If your report includes embedded objects, such as a list with an embedded chart and crosstab, the embedded objects do not appear in the Cognos Workspace Content pane, unless the embedded objects are in a table object. When you insert an object that contains embedded objects, the embedded objects will be inserted in the workspace.

  When embedded objects are in a table object, the embedded objects appear in the Content pane. Table objects do not appear in the Content pane because Cognos Workspace does not support Report Studio table objects. Consequently, you can insert the embedded objects in a workspace, but not the table.

- If your report includes a map, customize the colors of the map in Report Studio so that they match the color palette of the dashboard. You cannot edit the color palette of a map from within a workspace in Cognos Workspace.

- If your report is using a dimensional data source and you enable drill-up and drill-down, in the Drill Behavior window, specify that the column title uses member caption values.

  Cognos Workspace does not support drill up and drill down with level label values as the column titles. Cognos Workspace always uses member captions.

Related tasks:

“Create a Drill-up and Drill-down Report” on page 308

You can link groups of data items from different queries so that when you drill up or drill down in one query, the data item also drills up or drills down in the linked queries.

Prompts in workspaces in Cognos Workspace

A prompt control, when assigned a Name property in IBM Cognos Report Studio, appears as an object that you can insert from the Content pane in a workspace in IBM Cognos Workspace.

If a workspace includes reports and report parts from the same model, prompts automatically filter the appropriate data item in all the widgets in the dashboard. For example, a prompt filters on Product Line. Any widget in the dashboard that includes the Product Line data item is filtered when you select a value from this
prompt control. If you do not want a prompt to control a widget in the dashboard, you can disable communication between the two widgets.

If a dashboard includes prompted reports or report parts from different models, ensure that the prompts use the same parameter names. For example, a year prompt should use the same parameter `p_Years` in reports from both models. Otherwise, the prompt control is unable to control reports from another package.

Enable filters in workspaces in Cognos Workspace

You can include a data item in the query of the report, but not show that data item in the actual report. This is useful for creating calculations and filters. If you want IBM Cognos Workspace users to filter content in a workspace based on a data item that is not visible in the report, you must include this data item in its own query and name it `_BusinessInsight_` within the IBM Cognos Report Studio report. Therefore, the data item must exist in both the query used for the data container (such as the list, crosstab, or chart) and in this separate query.

For example, a chart shows the revenue for the product lines for each region. You want to filter the chart to show only data for the year 2012. When you author the report, in addition to including Year in the Query1 used for the chart data container, you create a new query named `_BusinessInsight_` that includes Year.

**Procedure**

1. Pause your pointer over the Query Explorer and click Queries.
2. From the Toolbox tab, drag a Query object to the work area.
3. Click the Query object, and in the Properties pane, in the Name box, type `_BusinessInsight_`.
4. Double-click the Query object, and from the Source tab, add the data item to use as a filter.
   Ensure that the same data item also exists in the query used for the data container.

Creating Reports for IBM Cognos for Microsoft Office

IBM Cognos for Microsoft Office provides an integrated environment for IBM Cognos products and Microsoft Office. You can use IBM Cognos for Microsoft Office to select pieces of reports to embed in Microsoft Excel workbooks, Microsoft Word documents, or Microsoft PowerPoint presentations, including data, metadata, headers, footers, and charts. You can use predefined reports or you can create new content using IBM Cognos PowerPlay® Web, IBM Cognos Query Studio, IBM Cognos Workspace Advanced, or IBM Cognos Report Studio.

**Note:** Note that to access PowerPlay content, your administrator must configure PowerPlay to work with IBM Cognos Business Intelligence. PowerPlay content that is published only to Upfront is not available to IBM Cognos for Microsoft Office.

Because IBM Cognos for Microsoft Office cannot fully convert highly formatted reports into Excel or other Microsoft document output types, you may not get the results that you want.

To create effective reports for IBM Cognos for Microsoft Office, follow these recommendations:
• Create content to meet specific Microsoft Office integration needs.
  For example, in IBM Cognos BI, many options are available to format data. Use
  less formatting to make data more accessible to Office applications.

• Organize reports.
  You can publish workbooks to IBM Cognos Connection and organize them with
  your reports in Public Folders or My Folders. For more information, see the
  IBM Cognos Connection User Guide. By organizing your content, you can
  quickly retrieve the information that you want.

  **Tip:** Workbooks, documents, and presentations that are enabled for IBM Cognos
  for Microsoft Office are identified by their own unique icons, helping you to
  distinguish them from other types of files.

• Optimize report templates for Microsoft Office.
  If you rely on IT personnel or other report authors to create content, request
  report templates that are optimized for your Microsoft Office integration needs.
  You may want to request only the data elements or queries that you need and
  request minimal formatting so that you can more easily use Microsoft Office
  formatting capabilities with the IBM Cognos content. For example, reports
  authored in Report Studio can contain list objects embedded within list objects
  with specific formatting applied. When converted to the tabular representation
  available in Excel, these reports may not be rendered in the same way in which
  they appear in IBM Cognos BI.

• Format elements in the Office application.
  Instead of formatting objects in IBM Cognos BI, add the formatting in the Office
  application. By applying less formatting in IBM Cognos BI, you can import more
  data into the desired locations.

• Label report elements using descriptive names.
  This practice makes them more easier to find after you import them. Examples
  of report elements include lists, crosstabs, and charts.

• Do not nest report objects.
  If you nested report objects, some objects may not appear in the correct location,
  or they may not appear at all. In addition, nesting report objects may cause the
  following error message to appear:

  **RDS-ERR-1000 Report Data Service could not process from the content provider.**
  For example, this error occurs if a repeater or repeater table is inside a block or
  table. It also appears when layout objects, such as lists, crosstabs, and charts, are
  in a conditional block that is inside another block or table.

• Keep table sizes small.
  For example, because of the size of slides, the maximum number of rows and
  columns that you can have in Microsoft PowerPoint tables is 25. Although Word
  and Excel permit larger tables, it takes more time to download and render them.

• Use images with transparent backgrounds.
  The background will show through the image in the Office application, making
  the image look like part of the presentation. If you want, you can then supply
  your own background color.

• Specify the height and width of images in a list.
  This practice ensures that the image appears in the correct size in the Office
  application.

• Remember that graphs and charts are imported as images.
Images in IBM Cognos BI have image maps associated with them to enable tooltips and hotspots. IBM Cognos BI for Microsoft Office cannot import tooltips and hotspots into Office applications.

- Consider the additional limitations that exist when producing reports in Excel format.

---

### Creating Reports for Mobile Devices

You can send IBM Cognos Report Studio reports to mobile devices that have IBM Cognos Mobile installed, or that are using the IBM Cognos Mobile Web portal.

**Creating location-aware reports**

A location-aware report filters report data based on the user’s current location as determined by the GPS coordinates of the mobile device. For example, a Vice President of Sales travels frequently to visit his Regional Sales Managers. When he arrives at an airport, he wants to filter his sales report to show information about the region that he is visiting.

You can create location-aware reports in IBM Cognos Report Studio if your IBM Cognos Mobile administrator has set up location-aware parameters. For more information, check with your IBM Cognos Mobile administrator or see the IBM Cognos Mobile Installation and Administration Guide.

Location-aware reports must include at least one prompt that requests the location information. This location information, as defined by your IBM Cognos Mobile administrator, could use either coordinates or defined regions, such as states, provinces, cities, or voter regions. Your administrator also defines the parameter names for the prompts, which you must provide when adding the location prompt.
Chapter 4. Lists

Use lists to show detailed information from your database, such as product lists and customer lists.

A list shows data in rows and columns. Each column shows all the values for a data item in the database or a calculation based on data items in the database.

You can specify whether to automatically add an overall aggregate summary in the list footer and a summary for any groups in the list by selecting Automatic group and summary behavior for lists in the IBM Cognos Report Studio options. The default aggregation as specified in your enterprise data source is used.

Tip: The Order Invoices - Donald Chow, Sales Person sample report in the GO Sales (query) package includes a list. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Related concepts:
“Limitations When Using Set Expressions in List Reports” on page 532

In list reports, we recommend that you avoid using set expressions. When in a list, set expressions, such as TopCount, may produce fewer rows than in the corresponding crosstab.

Related tasks:
“Change a List into a Crosstab” on page 75
Change a list into a crosstab to view your data from a different perspective.

Set List Properties

Format lists to give them the appearance you want. You can specify formatting for individual column titles or bodies or for all columns in a list. When you specify formatting for all columns in a list, the formatting is automatically applied to new columns you subsequently add.

You can also quickly format lists by applying table styles.

Formatting for specific columns overrides formatting for entire columns. For example, you specify red as the background color for a specific column body and green as the background color for all columns. When you run the report, the specific column body is red and the remaining columns in the list are green. New columns added to the list will be green.
Procedure

1. To format an individual column, do the following:
   a. Click the column title or column body.
   b. To format the text in the column title or the data in the column body, click the unlock button in the toolbar and then click the title or body.
   c. To format the whole column, click the select ancestor button in the title bar of the Properties pane and click List Column.
   d. In the Properties pane, set the property value.
      For example, to specify a background color, click Background Color and choose the color.

2. To format all columns, do the following:
   a. Click a column in the list.
   b. To format list column bodies, click the select ancestor button in the title bar of the Properties pane and click List Columns Body Style.
   c. To format list column titles, click the select ancestor button in the title bar of the Properties pane and click List Columns Title Style.
   d. To format entire columns, click the select ancestor button in the title bar of the Properties pane and click List Columns.
   e. To format the entire list, click the select ancestor button in the title bar of the Properties pane and click List.

   Tip: You can also click the container selector (three orange dots) of the list to select it.
   f. In the Properties pane, set the property value.
      For example, to specify a background color, click Background Color and choose the color.

Related concepts:

Chapter 12, “Laying Out a Report,” on page 329

When creating a report, a good layout is essential to ensure that the information in the report is presented in a clear and effective manner.

Hide Columns in List Reports

You can hide columns from the output of list reports. If you select to hide a column, the query for that column is still executed.

Procedure

1. Select the column that you want to hide.
2. Click the select ancestor button in the title bar of the Properties pane and click List Column.
3. Set the Render property to No.

Results

When you run your report, the column is hidden in the report output.

Using Repeaters

Use repeaters to repeat items when you run the report. For example, you can use repeaters to create mailing labels, including customer names and addresses.
To build a repeater, drag the **Repeater** or **Repeater Table** object from the **Toolbox** tab to the work area. Use repeaters to repeat items across a single row without a particular structure. For example, you want to create a list that contains Year, and Product line. For each year, you want all product lines to appear in a single row. To do this, create a list with Year as a column and with a repeater as a second column. Then insert Product line into the repeater. Use repeater tables to repeat items in a table structure. Drop the items in the repeater, and modify the properties of the repeater to obtain the results you want. For example, you can specify how many frames appear per page in a repeater table by typing values in the **Across** and **Down** properties.

Horizontal pagination is not supported for data containers, such as lists or crosstabs, that are nested in repeater tables.

**Convert a List into a Repeater**

You can convert a list into a repeater table to take advantage of an existing list.

**Procedure**

1. Click any part of the list.
2. From the **Structure** menu, click **Convert List to Repeater**.

**Example - Create Mailing Labels**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create mailing labels for all of the company’s retailers.

**Procedure**

1. Open IBM Cognos Report Studio with the **GO Data Warehouse (query)** package.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Repeater Table** and click **OK**.
4. Click the repeater, click the select ancestor button in the **Properties** pane title bar, and click **Repeater Table**.
   
   **Tip:** You can also click the container selector (three orange dots) of the repeater table to select it.
5. In the **Properties** pane, set the following properties:
   - Set the **Across** property to 2.
   - Set the **Down** property to 5.
   - Double-click the **Table Properties** property, select the **Fixed size** check box, and click **OK**.
6. From the **Toolbox** tab, drag the **Table** object to the repeater.
   
   The **Insert Table** dialog box appears.
7. In the **Number of columns** box, type 1 and click **OK**.
8. Click the table, ensure that you see **Table Cell** in the **Properties** pane title bar, and modify the following properties:
   - Double-click **Background Image**, click **Specified** and click **Browse**.
   - Click **logo.jpg** and click **OK**.
   - In the **Background Image** dialog box, under **Position**, click the align top right button.
• Under Tiling, click Do not tile and click OK.
• Double-click Size & Overflow, and in the Height box, type 175, and click OK.

9. Click the table, click the select ancestor button in the Properties pane title bar, and click Table.

Tip: You can also click the container selector (three orange dots) of the table to select it.

10. In the Properties pane, specify properties for the table:
• Double-click Border.
• In the Style box, click Solid line.
• In the Width box, click 1 pt.
• In the Color box, click Black.

• Under Preview, click the apply all borders button and click OK.
• Double-click Font, and under the Size box, click 8pt, and click OK.

11. From the Toolbox tab, drag the Block object to the repeater 8 times to create 8 blocks.

12. Drag the Text Item object to the first block:
• In the Text dialog box, type To: and click OK.
• Select the text item.
• Double-click the Font property, set the weight to Bold, and click OK.

13. Click the first block, and, in the Properties pane, specify properties for the first block:
• Double-click the Padding property, type 35 in the box on the right, click mm as the unit, and click OK.
• Set the Horizontal Alignment property to Center.

14. From the Source tab, expand Sales and Marketing (query), Sales (query), and Retailer site and drag seven data items to the remaining seven blocks.

15. Ctrl+click the seven blocks to select them, and in the Properties pane, specify properties for the seven blocks:
• Double-click the Padding property, type 25 in the box on the left, click mm as the unit, and click OK.
• Set the Horizontal Alignment property to Left.

Results

When you run the report, each page contains 10 mailing labels in two columns.
Chapter 5. Crosstabs

Use crosstab reports, also known as matrix reports, to show the relationships between three or more query items. Crosstab reports show data in rows and columns with information summarized at the intersection points.

For example, the crosstab below shows the gross profit and revenue by product line for each year.

<table>
<thead>
<tr>
<th>Profitability by Product Line</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Tip: The Same Month Prior Year sample report in the Sales and Marketing (cube) package includes a crosstab. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Related tasks:
“Insert a Microchart into a Crosstab” on page 152
You can use microcharts to improve the visualization of data in crosstabs.

Crosstab Nodes and Crosstab Node Members

When you add data items to crosstabs, you create crosstab nodes and crosstab node members. These objects allow you to easily create crosstabs, by dragging data items to crosstabs.

Crosstabs are dimensional objects that have row edges and column edges. Each edge is composed of a set of crosstab nodes. Each crosstab node contains the following:

- One or more crosstab node members.
- Zero or one nested crosstab node, which contains one or more crosstab node members or nested crosstab nodes.

Each crosstab node member refers to a data item that contains an expression to define the members in the crosstab.

The following crosstab contains four crosstab nodes.
1. Crosstab node 1 contains a single node member for the total. This node refers to the data item Total(Product line).

2. Crosstab node 2 contains a crosstab node member that refers to the data item Product line. This member has a nested crosstab node containing a crosstab node member that refers to the data item Product type.

3. Crosstab node 3 contains a single node member for the average. This node refers to the data item Average(Product line).

4. Crosstab node 4 contains a crosstab node member that refers to the data item Order year. This member has two nested crosstab nodes. The first node contains a crosstab node member that refers to the data item Order month. The second node contains a crosstab node member for the total. This node refers to the data item Total(Order month).

Tip: Nodes are also created when you add data items to charts.

Moving Crosstab Nodes

Crosstab nodes can be placed anywhere in the crosstab. For example, in the previous diagram, you can drag Order month under Average to create a row edge.

To move nested items from one edge to another, ensure that you select the crosstab node and not the crosstab node member. For example, in the previous diagram, you want to move Product line and Product type to the column edge. To do this, click Product line and, in the Properties pane, click the select ancestor button and click Crosstab Node. Both Product line and Product type are selected.

Crosstab Node Creation Option

In the Structure menu, the Create Crosstab Nodes option affects the drag-and-drop behavior in crosstabs. When the option is turned on and you add a data item to a crosstab, the item is created as a crosstab node. If the option is turned off, the item is created as a crosstab node member to an existing crosstab node.

For example, in the previous diagram, if the option is turned on and you drag Country or Region from the Source tab to beneath Product line, Country or Region becomes a new peer node to Product line. If the option is turned off, Country or Region becomes a new peer node to Product line and has Product type as a nested crosstab node. This happens because instead of adding a new node, you are adding a new member to the existing node that already contains Product line.
Tip: To create discontinuous crosstabs, turn the Create Crosstab Nodes option on. If you want the items on the edges of crosstabs to be related (contain the same nested items), turn the Create Crosstab Nodes option off.

**Set Crosstab Properties**

Format crosstabs to give them the appearance you want. You can specify formatting for rows, columns, and fact cells in a crosstab or for the entire crosstab. When you specify formatting for all rows, columns, fact cells, or the crosstab, the formatting is automatically applied to any new items you add.

If you apply the same styles, such as font color, to crosstab rows, columns, and crosstab intersections, the styles are applied in the following order:

1. crosstab fact cells
2. fact cells in the outermost rows
3. fact cells in the innermost rows
4. fact cells in the outermost columns
5. fact cells in the innermost columns
6. crosstab intersections

The style applied to the last object in the list overrides styles applied to previous objects. In addition, class styles are applied before styles that you apply manually.

You can also quickly format crosstabs by applying table styles and add white space to a crosstab by inserting crosstab space objects.

Tip: In cases where fact cell formatting applied to rows conflicts with fact cell formatting applied to columns, you can set the Fact Cells Precedence property in the Properties pane to determine whether the row formatting or the column formatting has precedence. To set this property, click anywhere in the crosstab, click the select ancestor button in the title bar of the Properties pane, and click Crosstab. Or you can click the container selector (three orange dots) of the crosstab to select it.

**Procedure**

1. To format the entire crosstab, do the following:
   a. Click anywhere in the crosstab.
   b. Click the select ancestor button in the title bar of the Properties pane and click Crosstab.
   c. In the Properties pane, set the property value. For example, to specify a background color, click Background Color and choose a color.

2. To format all rows, columns, or fact cells, click a row, column, or fact cell in the crosstab and then do the following:
   a. If you clicked a row, click the select ancestor button in the title bar of the Properties pane and click Crosstab Rows.
   b. If you clicked a column, click the select ancestor button in the title bar of the Properties pane and click Crosstab Columns.
   c. If you clicked a fact cell, click the select ancestor button in the title bar of the Properties pane and click Crosstab Fact Cells.
   d. In the Properties pane, set the property value. For example, to specify a background color, click Background Color and choose the color.
Tip: You can also right-click the row or column and click **Select Fact Cells**.

3. To format all crosstab cells for a specific row or column, do the following:
   a. Click the row or column.
   b. Click the select ancestor button in the title bar of the Properties pane and click **Crosstab Member Fact Cells**.

   **Tip:** You can also right-click the row or column and click **Select Member Fact Cells**.
   c. In the Properties pane, set the property value. For example, to specify a background color, click **Background Color** and choose the color.

4. To format all row or column titles, do the following:
   a. Click a row or column title.
   b. Click the select ancestor button in the title bar of the Properties pane and click **Crosstab Rows** or **Crosstab Columns**.
   c. In the Properties pane, set the property value. For example, to specify a background color, click **Background Color** and choose the color.

5. To format an individual row, column, or intersection, do the following:
   a. Click the row, column, or intersection.

   **Tip:** To format the data in a row, column, or intersection, click the unlock button in the toolbar and then click the text item to format.
   b. In the Properties pane, set the property value. For example, to specify a background color, click **Background Color** and choose the color.

**Related concepts:**

[Chapter 12, “Laying Out a Report,” on page 329](#)

When creating a report, a good layout is essential to ensure that the information in the report is presented in a clear and effective manner.

---

**Create a Single-Edge Crosstab**

Create a single-edge crosstab report to show data in a list-like form. For example, to show the quantity of products sold for each year and for each order method, you could create a crosstab with **Order Year** and **Order Method** as rows and **Quantity** as the measure.

**Procedure**

1. From the Toolbox tab, insert a crosstab object to the work area.

2. From the Source tab, drag data items to the **Rows** or **Columns** drop zone. A black bar indicates where you can drop the data item.

3. Repeat step 2 to insert additional data items:
   - If you dragged the data item in step 2 to the **Rows** drop zone, drag the additional items above or below the first item.
   - If you dragged the data item in step 2 to the **Columns** drop zone, drag the additional items to the left or right of the first item.

4. To add measures to the crosstab, drag the measures to the **Measures** drop zone.
Results

When you run the report, a crosstab is produced that has only one edge.

Create a Nested Crosstab

Nest data in a crosstab report to compare information by using more than one data item in a column or row. For example, a report shows the number of sales by product line for the past fiscal year. You decide to add a data item to further break down the number of sales by quarter.

When nesting columns in a crosstab report, there are four distinct drop zones where you can insert a new data item. The drop zone you choose will define the relationship between the data item and the column.

The following relationships are created when you insert a data item as a row:

- Inserting a data item to the left or right of a column creates a parent-child relationship between them.
  
  When you insert a data item to the left of a column, the data item becomes a parent to the column. When you insert a data item to the right of a column, the data item becomes a child of the column.

- Inserting a data item above or below a column creates a union relationship between them.

The following relationships are created when you insert a data item as a column:

- Inserting a data item to the left or right of a column creates a union relationship between them.
  
- Inserting a data item above or below a column creates a parent-child relationship between them.

When you insert a data item above a column, the data item becomes a parent to the column. When you insert a data item below a column, the data item becomes a child of the column.

For example, you have a crosstab with Product line as rows and Quantity and Revenue as nested rows. For columns, you have Order method with Country or Region as a nested column. In this crosstab,

- Product line is a parent to Quantity and Revenue.
- Quantity and Revenue are peers.
- Order method is a parent to Country or Region.

Procedure

1. From the Source tab, click the data item to add.
2. Drag the data item to the report as a nested column or nested row.
   
   A black bar indicates where you can drop the data item.
3. Repeat steps 1 to 2 to add other nested columns or rows.

Tip: If you add more than one measure to a crosstab, all measures appear as columns. You cannot have one measure appear as a row and another as a column. To make all measures appear as rows, swap columns and rows.
Creating headers automatically for sets in crosstabs

You can automatically add header labels on columns and rows when working with sets in a crosstab that uses a dimensional data source. The headers help consumers of the report to understand where the data is in the hierarchy.

Header labels are added only when you add new sets to your crosstab. The labels are not added to existing sets. After you enable this option, it remains on until you disable it, and applies to any new crosstabs that you create.

Procedure
1. From the Tools menu, click Options, then click the Report tab.
2. Select Automatically create crosstab headers for sets.
3. Click OK.

Results
Header labels are created in the rows and columns in a crosstab.

Showing crosstab values as percentages

When you work with dimensional data sources, you can show crosstab values as a percentage of a summary instead of the actual values. When you show values as a percentage, you can compare the contribution of an item to the whole.

About this task
You can show crosstab values as a percentage of a summary on the rows, the columns, or the rows and columns. For example, a crosstab has Product line as rows, Year as columns, and Revenue as the measure. You can show the following crosstab values:
- The actual values of Revenue
- A percentage of a summary on Product line
- A percentage of a summary on Year
- A percentage of a summary on Product line, Year

By default, the summary type that is used to calculate the percentages is Automatic summary.

When you show values as a percentage, a percentage measure data item is created.

You can also create a custom percentage calculation. For example, you can show values as a percentage of an intersection (tuple). Or you can show values as a percentage of a different summary type, such as Maximum.

When you show values as a percentage, the fact cells are formatted with the percent data format. When you show the actual values, the fact cells are formatted with the number format. If you modify the data format of the actual values, the change is lost when you switch between showing the actual values and percentage values.

You can show values as a percentage only in crosstabs. If you insert a chart, you cannot fill the chart with data from a crosstab that shows values as a percentage. If
you create a chart from a crosstab that shows values as percentages, an empty chart is created in the report.

Procedure
1. Select the measure for which you want to show values as a percentage.
   If there is only one measure in the crosstab, click the crosstab corner.
2. From the Data menu, click Show Value As, and click the percentage values that you want to show.
3. If you click Custom, provide the information that is required to calculate the percentage values.
   a. To change the default name of the percentage data item that you are creating, type the new name in the Name field.
   b. In the Percentage based on box, choose to build the percentage data item based on a summary of sets in the crosstab or based on an intersection (tuple).
   c. If you choose to build the percentage data item based on a summary of sets, in the Choose sets box, click the sets that you want to use. Then, click the Summary type menu and select the summary that you want to use to calculate the percentage values.
      If there is more than one data item on the crosstab edge that you chose for the sets, select which data item that you want to use.
   d. If you choose to build the percentage data item based on an intersection (tuple), click the ellipsis and select the items that you want to use to create the intersection.
   e. Click OK.
4. To edit a measure percentage data item, follow these steps.
   a. Select the measure.
   b. From the Data menu, click Show Value As > Edit.
   c. Modify the information for the data item as described in step 3.
   d. If you replaced a set in the crosstab with a different set, in the Choose sets box, click Existing sets when you want to base the percentage values on the set that you replaced.

Related tasks:
“Create an Intersection (Tuple)” on page 306
When working with dimensional data, an intersection, also known as a tuple, is useful for obtaining a value from the combination of two or more members that you specify. Each member must be from a different hierarchy. The intersection can include only one measure.

Specify the Default Measure

You can specify the default measure for crosstabs and certain chart types. Specify the default measure to be used when the measures cannot be determined by what is on the edges. For example, you create a crosstab with Order method as rows and Product line as columns. You add Quantity and Revenue as nested rows, making Order method their parent. You then add Country or Region under Order method. Since there is no measure specified for Country or Region, you specify the default measure so that data is returned for each country or region.

Note: In crosstabs, IBM Cognos Report Studio automatically sets the default measure when you insert a measure into the crosstab cells.
**Procedure**

1. To specify the default measure for a crosstab, do the following:
   - Click any part of the crosstab, and then click the select ancestor button in the title bar of the Properties pane.
   - Click Crosstab.
     
     **Tip:** You can also click the container selector (three orange dots) of the crosstab to select it.
   - Set the Default Measure property to the default measure.

2. To specify the default measure for a chart, drag the measure to the Default measure box in the chart.

---

**Swap Columns and Rows**

Swap columns and rows to look at information from a different perspective. This may help you discover high and low points in the data that you hadn't previously noted.

You can only swap columns and rows in a crosstab or chart. In a chart, you swap the x- and y-axes.

**Procedure**

From the toolbar, click the swap rows and columns button.

**Results**

In the report, the rows become the columns and the columns become the rows.

---

**Indent Data**

You can indent crosstab node members to set them apart from surrounding data.

When you insert a hierarchy in rows, all members are automatically indented according to their level. By default, the first member in a set is not indented. If you insert a hierarchy in columns, the members are not automatically indented. You can change the indentation properties of members in the report.

Relative indentation means that the member will shift by one tab when the member's level increases relative to the previous member in the hierarchy.

Indenting based on the level in the hierarchy means that the member will shift the number of tabs equivalent to the level in the hierarchy.

You can also indent objects by applying padding.

**Procedure**

1. Click the crosstab node member to indent.
2. In the Properties pane, set the Level Indentation property to your indentation type, length, and direction.
Change a List into a Crosstab

Change a list into a crosstab to view your data from a different perspective.

Procedure
1. Click the columns to appear as columns or nested columns in the crosstab.
2. From the Structure menu, click Pivot List to Crosstab.

Results
The list becomes a crosstab with the columns you selected in step 2 appearing as columns and nested columns. The unselected columns, except for measures, appear as rows and nested rows. If you have one measure, it becomes the cells of the crosstab. If you have more than one measure, they appear as columns.

Tip: To make all measures appear as rows, swap columns and rows.

Example - Add Aggregate Data to a Crosstab

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a report showing sales by order method to determine which methods are generating the most revenue and the highest sales volume.

Procedure
1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click Crosstab and click OK.
4. From the Source tab:
   - Expand Sales and Marketing (query) and Sales (query).
   - Expand Product and double-click Product line to add it as rows.
   - Expand Order method and double-click Order method to add it as columns.
   - Click Product type and drag it just to the right of Product line.
     Product type is now nested in Product line.
   - Expand Sales fact and drag Quantity to the Measures drop zone to add it as a measure.
   - Drag Revenue to the right of Quantity to add it as a second measure.

   Tip: Drop Revenue into the crosstab when you see a vertical bar between the Product type column and the Order method column.
5. Click any part of the crosstab, and then click the select ancestor button in the title bar of the Properties pane.
6. Click Crosstab.

   Tip: You can also click the container selector (three orange dots) of the crosstab to select it.
7. In the Properties pane, double-click the Font property.
8. In the Size box, click 8pt and click OK.
9. Click one of the measures.

10. Click the summarize button and click **Maximum**.

**Results**

When you run the report, you can see that for camping equipment, **Cooking Gear** generated the highest sales volume for the **Special** order method and **Tents** generated the most revenue. Tents ordered by the **Web** order method generated the largest revenue.

---

**Example - Create a Discontinuous Crosstab**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a report showing sales for each product line by quarter and by order method. Since the report will have columns with data from different dimensions, you create a discontinuous crosstab report. Discontinuous crosstabs are also known as disconnected or disjoint crosstabs or crosstabs with unrelated columns.

**Procedure**

1. Open IBM Cognos Report Studio with the **GO Data Warehouse (query)** package.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Crosstab** and click **OK**.
4. From the **Source** tab:
   - Expand **Sales and Marketing (query)** and **Sales (query)**.
   - Expand **Product** and drag **Product line** to the **Rows** area.
   - Expand **Order method** and drag **Order method** to the **Columns** area.
   - Expand **Sales fact** and drag **Revenue** to the **Measures** area.
5. On the **Source** tab, expand **Retailer site** and drag **Region** to the left of **Order method**.

   **Tip:** Drop **Region** into the crosstab when a flashing vertical bar appears between the **Product line** column and the **Order method** column. Otherwise, **Region** may appear as a nested row instead of a column.

6. Click **Region**.

7. In the **Properties** pane, double-click the **Sorting** property.

8. From the **Data Items** pane, drag **Region** to the **Sort List** pane.

9. Click the sort order button to sort in ascending order and click **OK**.

10. Run the report.

**Results**

<table>
<thead>
<tr>
<th>Revenue</th>
<th>Americas</th>
<th>Asia Pacific</th>
<th>Central Europe</th>
<th>Northern Europe</th>
<th>Southern Europe</th>
<th>E-mail</th>
<th>Fax</th>
<th>Mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Equipment</td>
<td>481,415,781.04</td>
<td>421,639,391.32</td>
<td>343,845,848.36</td>
<td>108,851,596.85</td>
<td>161,454,248.13</td>
<td>75,896,985.13</td>
<td>21,056,398.48</td>
<td>21,545,844.06</td>
</tr>
<tr>
<td>Golf Equipment</td>
<td>217,262,995.22</td>
<td>193,677,873.68</td>
<td>153,632,933.99</td>
<td>41,424,360.9</td>
<td>77,413,366.7</td>
<td>47,933,932.16</td>
<td>15,241,303.27</td>
<td>13,602,797.41</td>
</tr>
<tr>
<td>Mountaineering Equipment</td>
<td>123,127,397.88</td>
<td>107,615,775.01</td>
<td>88,051,152.89</td>
<td>46,091,108.04</td>
<td>44,684,319.05</td>
<td>7,476,491.96</td>
<td>11,848,370.08</td>
<td>3,531,858.68</td>
</tr>
<tr>
<td>Personal Accessories</td>
<td>593,596,705.30</td>
<td>439,000,120.35</td>
<td>431,030,405.23</td>
<td>210,980,208.02</td>
<td>204,251,710.0</td>
<td>42,951,000.54</td>
<td>17,902,905.30</td>
<td>8,419,357.07</td>
</tr>
<tr>
<td>Outdoor Protection</td>
<td>23,002,647.68</td>
<td>19,716,018.32</td>
<td>17,480,970.77</td>
<td>8,406,431.7</td>
<td>7,440,928.31</td>
<td>5,802,477.67</td>
<td>1,966,404.72</td>
<td>2,068,391.7</td>
</tr>
</tbody>
</table>

*Figure 4. Crosstab showing revenue by product line by region*
Chapter 6. Charts

You can use IBM Cognos Report Studio to create many chart types, such as column, bar, area, and line charts.

Current Default Charts and Legacy Charts

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

You will find procedures in this chapter for using both the legacy charts and the current default charts. To use the legacy chart type, set the Use legacy chart authoring option (Tools > Options > Advanced tab).

Related concepts:
“Limitations When Converting Legacy Charts” on page 103

When you convert a legacy chart to the current default chart, some chart types or chart properties might not be migrated properly. For example, clustered area charts and 100% stacked line charts are not available in the current default charts. If your exact chart configuration is not available, select the closest matching template.

Related tasks:
“Convert Legacy Charts to Current Default Charts” on page 103

You can convert charts from the legacy charts to the current default charts. When you convert a legacy chart, select the template that most closely matches your legacy chart in order to preserve as many settings as possible.

Creating Charts

Before creating charts, review the available chart types to select the best chart for your needs. Also review the chart objects that make up charts.

Tip: To view the available chart types, from the File menu, click New, and then double-click the chart icon. You can also view the available chart types by adding a Chart object from the Toolbox tab in the Insertable Objects pane to an existing report.

To create charts, drag data items from the Source tab to the measures, data series and categories drop zones.

To help you when creating charts, you can do the following:

- Preview your chart style changes.
  As you author your chart and edit the chart properties, IBM Cognos Report Studio provides you with a preview of your chart, using simulated data. This allows you to view your style changes without running your report. You need to run your report to see the chart with your actual data. You can disable the chart preview in the options.
- Resize your charts.
- Move the chart drop zones into flyouts so that they appear only when you pause your pointer over the chart.
After you create a basic chart, modify the various chart objects to customize your chart.

Chart Objects
The following shows the most common chart objects as they appear in your report output in IBM Cognos Viewer.

The following shows the same chart as it appears in the IBM Cognos Report Studio user interface. The Y-axis is selected, and the axis titles are displayed.
Click chart objects to specify properties

The default charts have more properties that allow you to customize more aspects of the chart than in earlier versions. You can click in different areas of the chart to get access to different properties.

For example, by clicking the axis, you can customize the color, style, weight, and transparency of chart axis lines and specify where to display major and minor gridlines on the chart.
**Data Series**

A data series is a group of related data points that are plotted in a chart. Each series has a unique color or pattern and is described in the legend. You can plot one or more data series in a chart; pie charts have only one data series.

In the example chart, the data series are order years 2010, 2011, 2012, and 2013.

**Categories**

Categories are groups of related data from the data series that are plotted on the X-axis. Categories of multiple data series are shown together using clustered and stacked data markers.

In the example chart, the categories are the product lines of The Sample Outdoors Company in clustered columns.

**Axes**

Axes are lines that provide references for measurement or comparison.

The primary axis (or Y-axis) refers to measures of quantitative data, such as sales figures or quantities. Charts can have more than one primary axis.
The category axis (X-axis or ordinal axis) plots qualitative data, such as products or regions. It runs horizontally, except in bar charts.

The z-axis is the vertical axis in a 3-D chart.

Major gridlines extend from the tick marks on an axis and run behind the data markers.

Related concepts:

"Customizing the Axes of a Chart" on page 119

Chart axes are lines that border the chart area and provide a reference for measurements. They act as a vehicle for tick marks and scales and form a frame around the chart. The Y-axis is usually the vertical axis and contains data. The X-axis is usually the horizontal axis and contains categories.

Legend

A legend is a key to the patterns or colors assigned to the data series or categories in a chart.

Columns, Lines, and Areas

Charts use graphical elements such as columns, horizontal bars, points, bubbles, lines, and areas as visual representations of data points.

Chart Types

IBM Cognos Report Studio provides many types of charts for presenting your data in a way that is meaningful for your users.

You can select from a variety of chart types (such as pie, bar, line, gauge, scatter, and so on) and you can select from a variety of chart configurations (such as stacked columns, 3-D pies). Combination charts allow you to use more than one chart type within your chart.

Some chart types are not supported for Microsoft Excel output or appear differently in Excel. For more information, see Chapter E, “Limitations When Producing Reports in Microsoft Excel Format,” on page 549.

Choosing a Chart Type and Configuration

To choose a chart type, consider what you want the chart to illustrate. Different chart types and configurations emphasize different things.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Chart type or configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show contributions of parts to a whole</td>
<td>Bar Charts</td>
</tr>
<tr>
<td></td>
<td>Pie Charts</td>
</tr>
<tr>
<td></td>
<td>Stacked Charts</td>
</tr>
<tr>
<td></td>
<td>when you want to display measures of the whole, as well as the parts</td>
</tr>
<tr>
<td></td>
<td>100 Percent Stacked Charts</td>
</tr>
<tr>
<td>Purpose</td>
<td>Chart type or configuration</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Show trends in time or contrast values across different categories</td>
<td>Line Charts</td>
</tr>
<tr>
<td></td>
<td>Area Charts</td>
</tr>
<tr>
<td></td>
<td>Bar Charts</td>
</tr>
<tr>
<td></td>
<td>Column Charts</td>
</tr>
<tr>
<td>Always place time in the horizontal axis.</td>
<td></td>
</tr>
<tr>
<td>Compare groups of related information against actual values</td>
<td>Bar Charts</td>
</tr>
<tr>
<td></td>
<td>Radar Charts</td>
</tr>
<tr>
<td>Compare different kinds of quantitative information</td>
<td>Combination Charts</td>
</tr>
<tr>
<td>Rank values in descending or ascending order</td>
<td>Bar Charts</td>
</tr>
<tr>
<td></td>
<td>Column Charts</td>
</tr>
<tr>
<td>Show correlation between two sets of measures</td>
<td>Point Charts</td>
</tr>
<tr>
<td>Show key performance indicators in an executive dashboard report</td>
<td>Gauge Charts</td>
</tr>
<tr>
<td></td>
<td>Bullet Charts</td>
</tr>
</tbody>
</table>

**Column Charts**

Column charts are useful for comparing discrete data or showing trends over time. Column charts use vertical data markers to compare individual values.

**Note:** The Sales Growth Year Over Year sample report "Sales Growth Year Over Year” on page 521 in the GO Data Warehouse (analysis) package includes a column chart. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

The following example shows the revenue for each product line.
Column charts can plot data using standard, stacked, 100 percent stacked, and three-dimensional configurations.

**Line Charts**

Line charts are useful for showing trends over time and comparing many data series.

Line charts plot data at regular points connected by lines.

Line charts can plot data using standard, stacked, 100 percent stacked, and three-dimensional configurations. It is best not to use stacked line charts because they are difficult to distinguish from unstacked line charts with multiple data series.

The following example shows a revenue trend that peaked in 2012 in every territory.
Note: The Top Retailers by Country or Region sample report "Top Retailers by Country or Region" on page 514 in the Sales and Marketing (cube) package includes a line chart. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Pie Charts**

Pie charts are useful for highlighting proportions.

They use segments of a circle to show the relationship of parts to the whole. To highlight actual values, use another chart type, such as a stacked chart.

Pie charts plot a single data series. If you need to plot multiple data series, use a 100 percent stacked chart.

Reports in PDF or HTML format show a maximum of 16 pies or gauges per chart.

Note: The Returns by Failed Orders in 2012 sample report "Returns by Failed Orders in 2012" on page 519 in the GO Data Warehouse (analysis) package includes a pie chart. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

The following example shows that the largest proportion of revenue comes from the Americas, followed by the Asia Pacific region.
Pie charts can plot data using standard, 100 percent, and three-dimensional configurations.

**Related tasks:**

- **“Set the Position of the First Slice in a Current Default Pie Chart” on page 148**
  You can specify the angle in a pie chart where the first pie slice starts. You can also change the direction of the slices so they appear clockwise or counter-clockwise.

- **“Create a Donut Chart from a Current Default Pie Chart” on page 149**
  You can add a hole to the middle of your pie chart to create a donut chart. You can then display something in the hole, such as a company logo, a calculation, or the legend.

- **“Pull Out Pie Slices in a Current Default Pie Chart” on page 149**
  You can highlight pie slices by pulling them out from the rest of the pie. For example, the following chart shows revenue by product line with slices of less than 1,000,000,000 in revenue pulled out by 25%.

**Bar Charts**

Bar charts are useful for showing trends over time and plotting many data series.

Bar charts use horizontal data markers to compare individual values.

**Note:** The Manager Profile sample report “Manager Profile” on page 517 in the GO Data Warehouse (analysis) package includes a bar chart. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

The following example shows revenue for every country or region.
Bar charts can plot data using standard, stacked, and 100 percent stacked configurations.

**Area Charts**

Area charts are useful for emphasizing the magnitude of change over time. Stacked area charts are also used to show the relationship of parts to the whole.

Area charts are like line charts, but the areas below the lines are filled with colors or patterns.

Do not use standard area charts to show multiple data series because it is possible for areas with lower values to be covered by others. For multiple data series, use a stacked area chart.

The following example is a stacked area chart showing the quantity of products sold over a four-year period in multiple territories.
Area charts can plot data using standard, stacked, 100 percent stacked, and three-dimensional configurations.

**Point Charts**

Point charts are useful for showing quantitative data in an uncluttered fashion.

Point charts use multiple points to plot data along an ordinal axis. A point chart the same as a line chart without the lines. Only the data points are shown.

The following example shows the revenue for each product line.

*Figure 6. Point chart*
Combination Charts

Combination charts plot multiple data series by using combinations of columns, areas, and lines within one chart. They are useful for highlighting relationships between the various data series.

Note: The Quantity Sold vs. Shipped and Inventory sample report in the GO Data Warehouse (analysis) package includes a combination chart. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

The following example shows a combination chart that includes planned revenue as a line chart and actual revenue as a column chart.

Combination charts can plot data using standard, stacked, 100 percent stacked, and three-dimensional configurations.

Scatter Charts

Scatter charts use data points to plot two measures anywhere along a scale, not only at regular tick marks.

You can also specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see “Specify the Default Measure” on page 73.

Scatter charts are useful for exploring correlations between different sets of data.

Note: The Employee Satisfaction 2012 sample report in the GO Data Warehouse (analysis) package includes a scatter chart. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.
The following example shows the correlation between production cost and gross profit for each product line.

![Chart showing correlation between production cost and gross profit](chart.png)

**Related tasks:**
- "Define Color by Value in Current Default Scatter and Bubble Charts" on page 146

In a scatter or bubble chart you can specify that data points or bubbles appear in different colors based on an additional measure or fact. The color of the points or bubbles adds an additional aspect to the chart to assist you in finding relationships in large amounts of data.

**Bubble Charts**

Bubble charts, like scatter charts, use data points and bubbles to plot measures anywhere along a scale. The size of the bubble represents a third measure.

You can also specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see "Specify the Default Measure" on page 72.

Bubble charts are useful for representing financial data. These charts are not supported for Microsoft Excel output.

The following example plots quantity and revenue by product line. The size of the bubble represents the amount of gross profit.
In a scatter or bubble chart, you can specify that data points or bubbles appear in different colors based on an additional measure or fact. The color of the points or bubbles adds an additional aspect to the chart to assist you in finding relationships in large amounts of data.

You can also specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see “Specify the Default Measure” on page 73.

Legacy quadrant charts use baselines to create the quadrants. Current default charts use colored regions.

Use a quadrant chart to present data that can be categorized into quadrants, such as a SWOT (strengths, weaknesses, opportunities, and threats) analysis.

The following example shows the relationship between production cost and gross profit. The size of the bubble represents the quantity of units sold.
Bullet Charts

Bullet charts are a variation of bar charts. They compare a featured measure (the bullet) to a targeted measure (the target). They also relate the compared measures against colored regions in the background that provide additional qualitative measurements, such as good, satisfactory, and poor.

Bullet charts are often used instead of gauge charts in executive dashboards. Bullet charts can be horizontal or vertical.

Note: This chart type applies only to the current default charts, and does not apply to the legacy charts.

A bullet chart contains the following components:

- A bullet measure.
  The bullet measure, Revenue, appears as the blue bar in the chart below.

- A target measure.
  The target measure, Planned revenue, appears as the black indicator in the chart below.

- From zero to five colored regions along the numeric scale to provide information about the featured measures’ qualitative state.
  The chart below includes three colored regions: 0-50%, 50-75%, and 75-100%.

- A label that identifies the measures.

- A numeric scale.
Related tasks:
“Customize a Current Default Bullet Chart” on page 152

After you create a bullet chart, you can customize the shape, color, and size of the bullet and target indicators.

Gauge Charts

Gauge charts, also known as dial charts or speedometer charts, use needles to show information as a reading on a dial.

On a gauge chart, the value for each needle is read against the colored data range or chart axis. This chart type is often used in executive dashboard reports to show key business indicators.

Gauge charts are useful for comparing values between a small number of variables either by using multiple needles on the same gauge or by using multiple gauges.

Reports in PDF or HTML format are limited to show a maximum of 16 pies or gauges per chart. These charts are not supported for Microsoft Excel output.

A gauge chart consists of a gauge axis (which contains the data range, color ranges, and intervals markers), needles, and a center pivot point. The following example shows a basic gauge chart with default attributes. It is a degree dial chart with two axes.
Related tasks:
"Customize a Current Default Gauge Chart" on page 144
When you create a gauge chart, you can choose from a variety of chart templates, which offer different shape, axis, and border options.

"Customize a Legacy Gauge Chart" on page 145
You can customize the colors and threshold boundaries of the gauge axis areas and whether they use numbers instead of percentages. You can also change the colors of the gauge face and outline and hide gauge labels. By default, the legacy gauge chart uses a band divided into thirds that runs from green to red.

Pareto Charts

Pareto charts help you to improve processes by identifying the primary causes of an event. They rank categories from the most frequent to the least frequent. These charts are frequently used for quality control data, so that you can identify and reduce the primary cause of problems.

Pareto charts include a cumulation line, which shows the percentage of the accumulated total of all the columns or bars.

You can create before and after comparisons of Pareto charts to show the impact of corrective actions. These charts are not supported for Microsoft Excel output.

The following example shows that the most frequent reason for product returns is unsatisfactory product.

You can also create Pareto charts using horizontal bars.
Related tasks:

“Define the Cumulation Line in a Current Default Pareto Chart” on page 151
The cumulation line on a Pareto chart displays the percentage of the accumulated total of all the columns or bars. For example, if your chart displays revenue by product line by year, the cumulation line at the second year’s column would be the total revenue of the first and second years.

“Define the Cumulation Line in a Legacy Pareto Chart” on page 151
The cumulation line on a Pareto chart displays the percentage of the accumulated total of all the columns or bars. For example, if your chart displays revenue by product line by year, the cumulation line at the second year’s column would be the total revenue of the first and second years.

Progressive Column Charts
Progressive column charts, also known as waterfall charts, are like stacked charts with each segment of a single stack displaced vertically from the next segment.

Progressive column charts are useful for emphasizing the contribution of the individual segments to the whole.

These charts are not supported for Microsoft Excel output.

The following example analyzes the contribution of each product line to revenue.

Progressive column charts can plot data using [standard](#) and [three-dimensional](#) configurations. You can also create progressive charts using horizontal bars.

Microcharts
Microcharts are smaller versions of column charts, bar charts, and line charts that you can use in crosstabs and dashboards. Microcharts include column microcharts and bar microcharts, which are available in stacked and 100 percent stacked configurations, and win-loss charts and modified line charts that contain the following:

- A closing value marker.
- Opening, closing, high, and low value markers.
A reference line.

Note: If you specify custom labels for a microchart, the custom labels appear in the tooltips, not on the chart itself. For more information about custom labels, see “Specify Custom Label Text” on page 119.

Win-loss charts are microcharts in which the value of each column is either 1 or -1, often denoting a win or loss.

Win-loss charts use two measures (the default and the win-loss measure) and no series. The win-loss measure is the measure or calculation that you define. You can also specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see “Specify the Default Measure” on page 73.

The following example shows the quarters that have a margin of less than 10,000 in red.

Marimekko Charts

Marimekko charts are 100 percent stacked charts in which the width of a column is proportional to the total of the column's values. Individual segment height is a percentage of the respective column total value.

The following example shows the contribution of revenues for product lines in different regions.
Radar Charts

Radar charts integrate multiple axes into a single radial figure. For each figure, data is plotted along a separate axis that starts at the center of the chart.

The following example shows the revenue from multiple retailer types in multiple territories.

![Radar Chart Example](image)

Radar charts can plot data using standard and stacked configurations.

Polar Charts

Polar charts are useful for showing scientific data.

They are circular charts that use values and angles to show information as polar coordinates.

You can specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see “Specify the Default Measure” on page 73.
The following example shows the revenue and quantity for each product line. The distance along the radial axis represents quantity, and the angle around the polar axis represents revenue.

![Radial Chart Example]

**Range Indicator Charts**

Range indicator, or metrics range, charts are useful for showing a target range and a tolerance range.

A range indicator chart adds a target and range marker to a column, line, or area chart.

The following example shows actual revenue versus planned revenue. The markers indicate the planned revenue, and the range and tolerance lines are defined by the chart author.

![Range Indicator Chart Example]

You can also insert IBM Cognos Metric Studio diagrams in your report to show the relationship between metrics. For more information, see "Add a Static Metric Studio Diagram to a Report" on page 159 and "Add a Data-driven Metric Studio Diagram to a Report" on page 159.
Chart Configurations

Chart configurations specify the grouping type of the columns, bars, lines, and areas in a chart. Some examples are standard, stacked, and 100 percent stacked charts.

Standard Charts

Standard or absolute charts are useful for comparing specific values and for representing discrete data, such as data for different regions or individual employees. For example, a standard column chart that plots regional sales emphasizes the actual value that each region achieves in sales.

Standard charts plot the actual value of each data series from a common axis.

When you create charts using multiple data series, you can distinguish each series by the color or pattern of its data marker. Related data series are shown together in clusters for easy comparison.

In standard area and radar charts that have multiple data series, the colored areas that represent lower values might be covered by the larger colored areas that represent higher values. Use the stacked configuration for area and radar charts with multiple data series.

The following example shows the revenue values for each product line within each territory.

Stacked Charts

Stacked charts are useful for comparing proportional contributions within a category. They plot the relative value that each data series contributes to the total. For example, a stacked column chart that plots product line sales will emphasize the proportion that each product line contributes to the total in each territory.

You can distinguish each data series by the color or pattern of its section in the stack. The top of each stack represents the accumulated totals for each category.
Do not use the stacked configuration in line charts that have multiple data series because it is difficult to distinguish between unstacked and stacked configurations, and your chart consumers might misunderstand your data.

The following example shows that camping equipment contributed a large proportion of the actual revenue in most sales territories.

100 Percent Stacked Charts
100 percent stacked charts are useful for comparing proportional contributions across all categories. They plot the relative contribution of each data series to the total as a percentage. For example, a 100 percent stacked column chart that plots product line sales emphasizes the percentage within each region without referring to actual values.

You can distinguish each data series by the color or pattern of its section in the stack. Each stack represents 100 percent.

100 percent stacked charts highlight proportions. When actual values are important, use another chart configuration.

The following example shows the percentage of sales for each product line in each region.
Three-dimensional Charts

Three-dimensional charts provide a visually effective display that is suitable for presentations.

Three-dimensional column, bar, line, and area charts plot data by using three axes.

Three-dimensional pie charts have a three-dimensional visual effect.

Do not use three-dimensional charts when you need to show exact values, such as for control or monitoring purposes. The distortion in three-dimensional charts can make them difficult to read accurately. For example, the following chart shows actual revenue for each product line in each territory, but some data labels are omitted because there is not enough room to display them all.

---

Convert Charts From One Type to Another

You can convert a chart from one type (for example, a bar chart) to another type (for example, a line chart).
When you convert an existing chart to a new chart type, IBM Cognos Report Studio keeps the properties from the existing chart if those properties exist in the new chart type. For example, if you convert a pie chart to a bar chart, Report Studio maps your chart palette to the new chart, but does not map the exploding slices, because the exploding slices property does not exist in a bar chart.

**Note:** Report Studio converts legacy charts to the current default chart types automatically if you change the chart type unless you select the *Use legacy chart authoring* option. You can convert legacy charts to current default charts, but you cannot convert current default charts back to legacy charts.

**Procedure**
1. Right-click a chart and click **Convert Chart**.
2. Select a new chart type, and click **OK**.
   
   If some chart properties cannot be converted, the **Chart Property Changes** dialog box appears notifying you of the properties that cannot be converted.

---

**Convert Legacy Charts to Current Default Charts**

You can convert charts from the legacy charts to the current default charts. When you convert a legacy chart, select the template that most closely matches your legacy chart in order to preserve as many settings as possible.

**Note:** You can convert legacy charts to current default charts, but you cannot convert current default charts back to legacy charts.

**Procedure**
1. From the **Tools** menu, click **Options**.
2. From the **Advanced** tab, clear the *Use legacy chart authoring* check box.
3. Right-click the legacy chart and click **Convert Chart**.
4. Select a new chart type, and click **OK**.
   
   If some chart properties cannot be converted, the **Chart Property Changes** dialog box appears notifying you of the properties that cannot be converted.

**Related concepts:**

"Current Default Charts and Legacy Charts" on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

---

**Limitations When Converting Legacy Charts**

When you convert a legacy chart to the current default chart, some chart types or chart properties might not be migrated properly. For example, clustered area charts and 100% stacked line charts are not available in the current default charts. If your exact chart configuration is not available, select the closest matching template.

The following legacy chart properties are not migrated to the current default charts:

- **Conditional Palette**
- **Style Variable**
- **Display Frequency** and **First Label Index** on the category axis
- **Visual Angle**
- **Rotate Values**
• Conditional Style, Style Variable, and Drill-Through Definitions on pie labels and gauge labels
• Connecting Lines on combination charts
• line styles; only four line styles are supported in the current default charts

The following table outlines some behavior differences between legacy charts and the current default charts.

<table>
<thead>
<tr>
<th>Legacy Charts</th>
<th>Current Default Charts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combination charts are always ordered from back to front - area, bar, and line.</td>
<td>Combination charts support any order.</td>
</tr>
<tr>
<td>In bar charts, multiple bar definitions for the Y1-axis are placed side-by-side.</td>
<td>In bar charts, multiple bars are placed on top of one another and you can control the width of the bars.</td>
</tr>
<tr>
<td>In combination charts, the color palette is continued over multiple bars, lines, and areas.</td>
<td>In combination charts, color palettes are restarted for each bar, line, and area.</td>
</tr>
<tr>
<td>The legend order respects the order in which bars, lines, and areas are specified. For example, bars can appear before areas in the legend.</td>
<td>The legend order is the same as the order specified in the chart.</td>
</tr>
</tbody>
</table>

The following chart types continue to use the legacy chart technology and therefore cannot be upgraded to the current default chart technology. If you want to convert these chart types to the current default charts, upgrade them to a different chart type, such as a column or line chart.
• Win loss
• Radar
• Polar
• 3-D Bar, 3-D Line, 3-D Area, and 3-D Combination
• 3-D Scatter
• Metrics range
• Marimekko

Related concepts:
“Current Default Charts and Legacy Charts” on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Customizing Chart Properties**

After you create a chart, you can customize it by changing its properties.

You select the chart object in IBM Cognos Report Studio to view its properties. Some properties are dependent on the existence of other properties.

If you are familiar with using conditional variables, you can customize the chart to change its appearance or provide information in response to expressions or conditions.

As you edit the chart properties, Report Studio provides you with a preview of your chart, using simulated data. This allows you to view your style changes without running the report. You can disable the chart preview in the options.
The following table shows some of the properties you can change in charts. These properties are available when you select the chart object unless specified otherwise in the **Action to perform in the Properties pane** column.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Action to perform in the Properties pane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide or show the title, subtitle, or footer</td>
<td>Select the chart. Under <strong>Chart Titles</strong>, set the <strong>Title</strong>, <strong>Subtitle</strong>, or <strong>Footer</strong> property.</td>
</tr>
<tr>
<td>Hide or show the legend</td>
<td>Select the chart. Under <strong>Chart Annotations</strong>, set the <strong>Legend</strong> property.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Customize the Legend in a Current Default Chart” on page 124.</td>
</tr>
<tr>
<td>Hide or show baselines</td>
<td>Select the chart. Under <strong>Chart Annotations</strong>, set the <strong>Baselines</strong> property.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Add a Baseline to a Current Default Chart” on page 130.</td>
</tr>
<tr>
<td>Hide or show trendlines or regression lines</td>
<td>Select the chart. Under <strong>Chart Annotations</strong>, set the <strong>Trendlines</strong> or <strong>Regression Line</strong> property.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Display Trendlines in Current Default Charts” on page 134 or “Display Regression Lines in Legacy Scatter or Bubble Charts” on page 135.</td>
</tr>
<tr>
<td>Hide or show markers</td>
<td>Select the chart. Under <strong>Chart Annotations</strong>, set the <strong>Markers</strong> property.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Add a Marker to a Current Default Chart” on page 132.</td>
</tr>
<tr>
<td>Hide or show notes</td>
<td>Select the chart. Under <strong>Chart Annotations</strong>, set the <strong>Notes</strong> property.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Add a Note to a Current Default Chart” on page 127.</td>
</tr>
<tr>
<td>Hide or show the axes</td>
<td>Select the chart. Under <strong>Axes</strong>, set the <strong>Y1 Axis</strong>, <strong>Y2 Axis</strong>, or <strong>Category Axis</strong> property.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Customize the Axes of a Current Default Chart” on page 120.</td>
</tr>
<tr>
<td>Hide or show the axis title or axis line</td>
<td>Select the axis. Under <strong>General</strong>, set the <strong>Axis Line</strong> or <strong>Axis Title</strong> property.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Customize the Axes of a Current Default Chart” on page 120.</td>
</tr>
<tr>
<td>Goal</td>
<td>Action to perform in the Properties pane</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Change an axis title</td>
<td>Select the axis title. Under General, set the Default Title property to No, and then double-click the axis title and type a new title. <strong>Tip:</strong> You can use the Properties pane to change the axis title properties, such as font, positioning, color, and so on. For more information, see “Customize the Axes of a Current Default Chart” on page 120.</td>
</tr>
<tr>
<td>Change axis properties, such as range, scale interval, and so on</td>
<td>Select the numeric axis. Under General, set the Minimum Value, Maximum Value, Scale Interval, or Scale property. For more information, see “Change the Axis Scale of a Current Default Chart” on page 120.</td>
</tr>
<tr>
<td>Hide or show the tooltips</td>
<td>Select the chart. Under Chart Labels, set the Tooltips property. When you pause your pointer over a data marker in the report output, the corresponding absolute or cumulative value appears in a tooltip. Tooltips are not supported in PDF output.</td>
</tr>
<tr>
<td>Change the data format</td>
<td>Select the numeric axis. Under Data, set the Data Format property.</td>
</tr>
<tr>
<td>Change the white space around the chart</td>
<td>Select the chart. Under Box, set the Padding or Margin property. For more information, see “Apply Padding to an Object” on page 341 and “Set Margins for an Object” on page 342.</td>
</tr>
<tr>
<td>Change the color or pattern in the palette for columns, lines, and areas</td>
<td>Select the chart object. Under Color &amp; Background, set the Palette or Conditional Palette property. For more information, see “Customizing the Color Palette of a Chart” on page 108 and “Create a Simple Conditional Palette for a Current Default Chart” on page 111.</td>
</tr>
<tr>
<td>Change the chart orientation</td>
<td>Select the chart. In the Properties pane, under Chart Orientation, select Vertical or Horizontal.</td>
</tr>
<tr>
<td>Goal</td>
<td>Action to perform in the Properties pane</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------</td>
</tr>
</tbody>
</table>
| Change the default color or font for all chart objects | Select the chart. Under **Color & Background**, set the **Background Color**, **Foreground Color**, or **Fill Effects** property.  
For more information, see [“Change a Chart Background in a Legacy Chart” on page 115](#).  
Under **Font & Text**, set the **Font** or **Relative Alignment** property.  
Tip: The 2011 Quarterly Sales Forecast sample report in the GO Sales (analysis) package includes text alignment. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507. |
| Override the default font or color for axes and chart values | Click the unlock button in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and set the **Font**, **Background Color**, **Foreground Color**, or **Fill Effects** property. |
| Change the three-dimensional appearance of a chart | Select the chart. Under **General**, set the **Depth** or **Visual Angle** property.  
For example, for a pie chart, a **Depth** value of 0 (zero) leaves the pie flat, or two-dimensional. A value of 100 raises the pie almost to a round ball shape.  
**Tip:** You can select a value from the **Depth** list or type a value that is not listed. |
| Insert a background image or watermark in the chart body. | Click the lock button in the toolbar to unlock the chart object, select the chart body by clicking between the axes.  
Under **Color & Background**, set the **Background Image** property.  
For more information, see [“Change a Chart Background in a Legacy Chart” on page 115](#). |
| Go to another report | Select the chart. Under **Data**, set the **Drill-Through Definitions** property.  
For more information, see Chapter 18, “Using Drill-through Access,” on page 415. |
| Specify which labels and values to use when generating the text shown on the chart | Select the chart. Under **Chart Labels**, set the **Values** property.  
**Note:** When you show all the labels and values on some chart types, such as scatter charts, bubble charts, and polar charts, the text shown might be too long. |

## Resize a Current Default Chart

You can change the size of a current default chart by dragging the resize handle in the lower right corner of the chart.
Procedure
1. Select the chart object.
2. Click the resize handle in the lower right corner of the chart. Use Shift+drag to maintain the aspect ratio of the chart.
3. To see the actual size of the chart as it will appear in the report output, collapse the chart objects with the minus (-) button in the upper left corner of the chart.

Results
The new chart size appears in the Size & Overflow property. You can also edit the chart size with this property.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Resize a Legacy Chart
You can resize a legacy chart by defining the height and width in the Size & Overflow property.

The options available with this property are detailed in “Specify the Height and Width of an Object” on page 348.

Procedure
1. Select the chart object.
2. In the Properties pane, double-click the Size & Overflow property and specify the height and width.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Customizing the Color Palette of a Chart
You can use the chart palette to control the colors or patterns used in the columns, lines, data markers or areas in a chart. For example, if “Telephone” is the first in a data series of order methods, and you want it to appear in blue, use the palette to make the first item in the series blue.

You can customize the color palette for area, bar, bubble, gauge, Pareto, pie, and scatter charts.

You can also apply background effects to chart objects or change the colors of specific chart elements.

Related tasks:
“Add Background Effects to a Chart Object in a Legacy Chart” on page 116
You can change the look of certain charts and chart objects by applying visual effects such as drop shadows, borders, fills, texture effects, and bevel effects.

Customize the Color Palette of a Current Default Chart
You can use the chart palette to control the colors or patterns used in the columns, lines, data markers or areas in a chart.

Progressive charts and legacy charts have different steps.
If specified, the default color is used for outputs that do not support the palette definition. For example, if the palette is defined to use a radial rectangle gradient and you run the report as PDF output, the default color is used because radial rectangle gradients are not supported in PDF output.

**Procedure**

1. Select the chart object. For combination charts, select the bar, line, or area objects.

2. To choose a preset color palette, click the chart palette presets button on the style toolbar, and then click a palette.

3. To change the existing color palette:
   - In the Properties pane, under Color & Background, double-click the Palette property.
   - On the Entries tab, click the palette entry that you want to change.
   - Under Fill, select fill type, color, transparency, and other fill settings. The options in the Fill type list depend on the chart type you are using.
     - **Linear Gradient** displays a gradient that changes color horizontally. You can specify the position of each color in the gradient and add or remove colors from the gradient.
     - **Radial Line Gradient** displays a gradient that changes color outwards from a central point, or focus. You can specify the position of each color in the gradient and the size and position of the focus. You can also add or remove colors from the gradient.
     - **Radial Rectangle Gradient** displays a gradient that changes color outwards from a rectangle. You can specify the position of each color in the gradient and the size of the rectangle. You can also add or remove colors from the gradient.
     - **Pattern** displays a colored pattern that you choose from a preset list. You can specify the foreground and background colors.
     - **Color** displays a solid color. You can specify the transparency.
     - **Image** displays an image. You can specify the image to use.
   - To change the color of this palette definition, click the color in the Colors list, click Color, specify the color properties, and click OK.

4. Under Style, depending on the chart type you are customizing and the fill type you chose in step 2, you can change the marker shape, line style, default color, and line weight.

5. To add a new palette entry, click the new button and specify the new palette settings.

6. To change the order in which the colors, gradients, or patterns appear in the chart, use the arrow buttons under the Palette box to change their position.

7. To copy and paste a palette, open the palette dialog box and press Ctrl+C to copy the palette to the clipboard. Close the palette dialog box. Select another chart, open the palette dialog box, and press Ctrl+V to paste the palette from the clipboard.

8. To customize the data markers, on the Properties tab, depending on the chart type you are customizing and the fill type you chose in step 2, you can specify whether data markers are shown and change their color and size.
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Customize the Color Palette of a Current Default Progressive Chart

You can use the chart palette to control the colors or patterns used in the columns, lines, data markers or areas in a chart.

Legacy charts and other types of current default charts have different steps.

If specified, the default color is used for outputs that do not support the palette definition. For example, if the palette is defined to use a radial rectangle gradient and you run the report as PDF output, the default color is used because radial rectangle gradients are not supported in PDF output.

Procedure

1. Select the progressive chart object.
2. In the Properties pane, double-click the Progressive Palette property.
   - You can specify the following:
     - **Positive value fill** defines the appearance of the positive bars or columns on the chart.
     - **Negative value fill** defines the appearance of the negative bars or columns on the chart.
     - **First value fill** defines the appearance of the first bar or column on the chart.
     - **Total value fill** defines the appearance of the total bar or column on the chart, if a total bar or column is present.
3. To change the appearance of a fill:
   - Click the colored box beside the fill name.
   - Click a fill type from the Fill type list.
     - **Linear Gradient** displays a gradient that changes color horizontally. You can specify the position of each color in the gradient and add or remove colors from the gradient.
     - **Radial Line Gradient** displays a gradient that changes color outwards from a central point, or focus. You can specify the position of each color in the gradient and the size and position of the focus. You can also add or remove colors from the gradient.
     - **Radial Rectangle Gradient** displays a gradient that changes color outwards from a rectangle. You can specify the position of each color in the gradient and the size of the rectangle. You can also add or remove colors from the gradient.
     - **Pattern** displays a colored pattern that you choose from a preset list. You can specify the foreground and background colors.
     - **Color** displays a solid color. You can specify the transparency.
     - **Image** displays an image. You can specify the image to use.
   - To change the color of this fill, click the color in the Colors list, click **Color**, specify the color properties, and click **OK**.
4. To change the text of the label that appears next to a bar, double-click the text box beside the fill name and specify the text.
Related concepts:
“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Customize the Color Palette of a Legacy Chart
You can use the chart palette to control the colors or patterns used in the columns, lines, data markers or areas in a chart.

Current default charts have different steps.

Procedure
1. Select the chart object.
2. In the Color & Background section of the Properties pane, double-click the Palette property.
3. In the Palette dialog box, click the new palette entry button to define a new palette entry:
   a. To apply color, click Color, specify the color properties, and click OK.
   b. To apply a gradient, click Gradient, specify the Direction, From color, and To color properties, and click OK.
   c. To apply a pattern, click Pattern, specify the Pattern, Foreground color, and Background color properties, and click OK.
   Tip: You can also select a predefined palette from the Palettes box.
4. To change the order in which the colors, gradients, or patterns appear in the chart, use the arrow buttons under the Palette box to change their position.
5. To copy and paste a palette, open the palette dialog box and press Ctrl+C to copy the palette to the clipboard. Close the palette dialog box. Select another chart, open the palette dialog box, and press Ctrl+V to paste the palette from the clipboard.
6. Repeat step 3 for each chart series.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Creating a Conditional Color Palette in a Chart
You can create a conditional palette to color data items in your chart in different ways depending on a condition. For example, in a column chart that shows revenue per month, you want to make the columns for the months that have a revenue greater than $1000000 green.

You can also use conditional styles to highlight exceptional data and use conditions to control the layout of your report.

Related concepts:
Chapter 14, “Using Conditions,” on page 365
You can define conditions to control what users see when they run a report. Conditions can apply to specific items in a report. For example, you can define a conditional style to highlight exceptional data, such as product revenue that exceeds your target.

Create a Simple Conditional Palette for a Current Default Chart
You can create a simple conditional palette based on a string data item or a numeric data item. For example, if you want to create a condition to display only
Revenues over one million dollars, you can create a numeric condition; whereas, if you want to create a condition to display only product codes with the letter A in them, you can create a string condition.

**Procedure**

1. Select the chart series to which you want to apply the conditional palette.
2. In the **Properties** pane, under **Color & Background**, double-click the **Conditional Palette** property.
3. Click the new button and click **New Condition**.
4. If you want to create a conditional palette based on a string, do the following:
   a. Select the string data item to determine the condition.
   b. In the **Type of conditional style** list, click **String**, and click **OK**.
   c. Click the new button and select how to define the condition:
      - To select more than one individual value, click **Select Multiple Values** and click the values.
      - To type specific values, click **Enter Values** and type the values.
      - To specify your own criteria, such as values that begin with the letter A, click **Enter String Criteria** and specify the condition.
   d. Under **Palette Entry**, beside the new condition, click the edit button.
   e. Define the palette entry fill type, colors, and angle, and click **OK**.
5. If you want to create a conditional palette based on a numeric range, date/time range, interval range, date range, or time range, do the following:
   a. Select a numeric data item to determine the condition.
   b. In the **Type of conditional style** list, click the type of condition you want to create, and click **OK**.
   c. Click the new button and type a value to define a threshold.
      The value appears in the **Range** column, and two ranges are created.
   d. For each range, under **Style**, click the edit button to define the palette entry fill type, colors, and angle, and then click **OK**.
   e. Repeat the steps above to add other conditions.

**Tip:** Under **Style**, pause the pointer over each range to see the condition produced for that range.

f. If you want to use the default palette for the lowest range instead of specifying a conditional palette select the **Use default palette for lowest range** check box.

g. To move a value above or below a threshold, click the arrow button next to the value.
   For example, you insert a threshold value of five million. By default, the ranges are less than or equal to five million and greater than five million. Moving the five million value above the threshold changes the ranges to less than five million and greater than or equal to five million.
6. To customize the label for a conditional palette, do the following:
   a. Click **Label** beside the palette entry.
b. In the Conditional Palette Entry Label dialog box, click Specified text and click the ellipsis (...) button.

c. Type the default label text in the Label text box.

d. To add customized labels for other languages, click the add button, select the other languages, and click the language to define the customized label for that language.

7. Specify the order in which to evaluate the conditions by clicking a condition and then clicking the move up or move down arrow.
   Conditions are evaluated from top to bottom, and the first condition that is met is applied.

**Related concepts:**

“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Create an Advanced Conditional Palette for Current Default Charts**

Advanced conditions allow you to use a calculations or expression to evaluate the condition. If multiple advanced conditions are met, only the first conditional style is applied.

**Procedure**

1. Select the chart series to which you want to apply the conditional palette.
2. In the Properties pane, under Color & Background, double-click the Conditional Palette property.
3. Click the add button and click Advanced Condition.
4. Click the new advanced condition button and specify the expression that defines the condition.
   The new advanced condition appears in the Conditional Palette - Advanced dialog box.
5. Under Palette Entry, beside the new advanced condition, click the edit button.
6. Define the palette entry fill type, colors, and angle, and click OK.
7. Specify the order in which to evaluate the conditions by clicking a condition and then clicking the move up or move down arrow.
   Conditions are evaluated from top to bottom, and the first condition that is met is applied.

**Related concepts:**

“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Create a Conditional Color Palette for a Legacy Chart**

You can create a color palette that appears differently depending on a yes or no answer, an expression or calculation, or the report language.

**Procedure**

1. Click the chart object.
2. In the Properties pane, under Color & Background, double-click the Conditional Palette property.
3. If you want to create a variable that has only two possible values, Yes and No, do the following:
   a. Under Variable, click New boolean variable.
   b. In the New Variable dialog box, type a name for the variable.
   c. In the Expression Definition box, define the condition and click OK.
      For example, the following expression returns the value Yes if revenue is less than one million and the value No if revenue is greater than or equal to one million:
      \[\text{Revenue} < 1000000\]
      For information about creating expressions, see “Using Relational Calculations” on page 248 or “Using Dimensional Calculations” on page 302 and Appendix F, “Using the expression editor,” on page 555.
   d. In the Value box, select whether the expression evaluates to Yes or No.
   e. Under Effects, specify the color palette to use for the condition.

4. If you want to create a variable whose values are string-based, do the following:
   a. Under Variable, select New string variable.
   b. In the Expression Definition box, define the condition and click OK.
      For example, the following expression returns the value high if revenue is greater than one million and the value low if revenue is less than or equal to one million:
      \[\text{if} \left(\text{Revenue} > 1000000\right) \text{then} \text{('high')} \text{else} \text{('low')}\]
      For information about creating expressions, see “Using Relational Calculations” on page 248 or “Using Dimensional Calculations” on page 302 and Appendix F, “Using the expression editor,” on page 555.
   c. Click the add button in the Values pane.
   d. For each value that the variable can assume, type the name of the value that corresponds with the possible outcomes defined in the expression.
      For example, in the previous expression, you must create two values for the variable, high and low.
      Tip: You can create a group by clicking two or more values and then clicking the group values button. For example, you can create a group that includes the available French languages.

5. If you want to create a variable whose values are different languages, do the following:
   b. In the Languages dialog box, select the languages to support.
   c. Click OK to close the expression editor and leave the Conditional Palette dialog box open.
   d. Under Effect, click Pattern.
   e. Under Pattern, select a pattern and click OK.
   f. In the Box section of the Properties pane, set the Borders property to Show.
Change a Chart Background in a Current Default Chart

You can use a gradient, pattern, color, or image to customize the chart background.

Procedure

1. Click the chart object.
2. Click the unlock button, select the chart body by clicking between the axes, and, in the Properties pane, under Color & Background, double-click the Plot Area Fill property.
3. Click a fill type from the Fill type list.
   - **Linear Gradient** displays a gradient that changes color horizontally. You can specify the position of each color in the gradient and add or remove colors from the gradient.
   - **Radial Line Gradient** displays a gradient that changes color outwards from a central point, or focus. You can specify the position of each color in the gradient and the size and position of the focus. You can also add or remove colors from the gradient.
   - **Radial Rectangle Gradient** displays a gradient that changes color outwards from a rectangle. You can specify the position of each color in the gradient and the size of the rectangle. You can also add or remove colors from the gradient.
   - **Pattern** displays a colored pattern that you choose from a preset list. You can specify the foreground and background colors.
   - **Color** displays a solid color. You can specify the transparency.
   - **Image** displays an image. You can specify the image to use.

Change a Chart Background in a Legacy Chart

You can use a solid color, a pattern, or a gradient fill effect to customize the chart background.

You can also use an image as a background for a chart.

Procedure

1. Click the chart object.
2. Click the unlock button, select the chart body by clicking between the axes, and, in the Properties pane, under Color & Background, double-click the Fill Effects property.
3. In the Effect box, choose a color, a gradient fill effect, or a pattern:
   - To apply a color, click Color and specify the color properties.
   - To apply a gradient fill effect, click Gradient and then specify the Direction, From color, and To color properties.
   - To apply a pattern effect, click Pattern and specify the Pattern, Foreground Color, and Background Color properties.
The foreground color is the color of the selected pattern. The background color is the color of the area behind the pattern.

4. To remove a background fill effect, click **None**.

**Related concepts:**

Current Default Charts and Legacy Charts” on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Related tasks:**

“Insert a Background Image in an Object” on page 334

You can insert a background image for objects in a report. For example, use a background image to add a watermark to a page.

### Add Background Effects to a Chart Object in a Legacy Chart

You can change the look of certain charts and chart objects by applying visual effects such as drop shadows, borders, fills, texture effects, and bevel effects.

**Procedure**

1. Select the **chart object**.
2. To apply a preset background fill effect, from the toolbar, click the background effects presets button.
3. To add a custom fill effect and border do the following:
   a. In the **Properties** pane, under **Color & Background**, double-click the **Background Effects** property.
   b. Click **Border** and specify settings for border style, width, color, corner radius for rounded rectangles, and transparency.
      
      If the object also includes a fill with a transparency setting, select the **Allow transparent bleed** check box to apply the same transparency to the border.
   c. Click **Fill** and specify the settings.
      
      The fill effect can either be a solid color, a gradient, or a pattern.
4. To add a texture effect, in the **Properties** pane, under **Color & Background**, select an option under **Material Effects**.
5. To add a drop shadow, do the following:
   a. In the **Properties** pane, under **Color & Background**, double-click the **Drop Shadow** property.
   b. Select the **Drop shadow** check box.
   c. From the **Shadow color** list, click a color.
   d. Under **Blur**, click an intensity to set the drop shadow’s blur.
   e. Under **Offset**, enter a value and a unit to define the width and height of the drop shadow.
   f. Under **Transparency**, type a percentage to determine the transparency of the shadow.
6. To add a bevel, in the **Properties** pane, under **Color & Background**, double-click the **Bevel** property and click a bevel effect.
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

You can use the chart palette to control the colors or patterns used in the columns, lines, data markers or areas in a chart. For example, if “Telephone” is the first in a data series of order methods, and you want it to appear in blue, use the palette to make the first item in the series blue.

You can add background and foreground color to objects in the report. The foreground color applies to the text within objects.

Add Colored Regions in a Current Default Chart

You can define colored regions in the body of a chart. For example, you can divide the background of a scatter chart into quadrants and color each quadrant.

You can add colored regions to bubble, bullet, combination, Pareto, progressive column, and scatter charts.

In bullet charts, colored regions are automatically added.

You can use the following criteria to position the colored regions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent on Axis (%)</td>
<td>Uses a percentage of the full range of the axis.</td>
</tr>
<tr>
<td></td>
<td>For example, if the axis range is -500 to 1100, a Percent on Axis value of 25% puts the baseline at -100 (25% of the range, 1600).</td>
</tr>
<tr>
<td>Numeric Value</td>
<td>Uses a static numeric value.</td>
</tr>
<tr>
<td>Mean</td>
<td>Uses the statistical mean plus or minus a number of standard deviations based on all charted data values on the specified axis.</td>
</tr>
<tr>
<td>Percentile (%)</td>
<td>Uses a specified percentile.</td>
</tr>
<tr>
<td>Statistical Maximum</td>
<td>Uses the following expression:</td>
</tr>
<tr>
<td></td>
<td>25th percentile value - 1.5 * (75th percentile value - 25th percentile value)</td>
</tr>
<tr>
<td></td>
<td>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical minimum is -5 [2.5 -1.5(5) = -5].</td>
</tr>
<tr>
<td>Statistical Minimum</td>
<td>Uses the following expression:</td>
</tr>
<tr>
<td></td>
<td>75th percentile value + 1.5 * (75th percentile value - 25th percentile value)</td>
</tr>
<tr>
<td></td>
<td>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical maximum is 15 [7.5 +1.5(5) = 15].</td>
</tr>
</tbody>
</table>
### Option Position

<table>
<thead>
<tr>
<th>Option</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Calculation</td>
<td>Uses a query calculation from the same query or from a different query. For more information, see “Using Relational Calculations” on page 248.</td>
</tr>
<tr>
<td>Layout Calculation</td>
<td>Uses a layout calculation. For more information, see “Using Relational Calculations” on page 248.</td>
</tr>
<tr>
<td>Category Index</td>
<td>Specifies a position based on the index value of the data item in the categories axis. The value must be between 0 and 100. For example, a Category index value of 1 indicates that the position is located at the first data item. This is the default.</td>
</tr>
</tbody>
</table>

You can also combine colored regions with gridlines.

### Procedure

1. Click the chart object.
2. In the Properties pane, under Color & Background, double-click the Colored Regions property.
3. Click the new button .
4. Under Region label, type a name for the region.
5. Under Label location, specify whether the label should appear in the legend or in the chart.
6. If your chart includes multiple charts, such as in a combination chart, under Based on, select the chart to which the colored regions apply.
7. Under Region fill, click the color box to specify the color and fill effects.
8. Specify the start and end of the category and numeric positions.
9. If you want to add more colored regions, repeat steps 3 to 5.
10. If you add more than one colored region, specify their order using the up and down arrows.
    The regions are drawn in the same order that they appear in this list. The first region is drawn first, on the bottom, and the last region is drawn on top of the other regions.
    **Tip:** To delete a colored region, click the colored region icon and text and click the delete button .

### Results

The new region appears in the Colored regions box.
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Specify Custom Label Text**

By default, chart labels use the name of the underlying data item label. You can change the label text to make it more meaningful. For example, you can rename a chart item labeled ISO_3_Letter_Code to Country or Region.

If you specify custom labels for a microchart or win-loss charts, the custom labels appear in the tooltips, not on the chart itself.

**Procedure**

1. Click the chart node member or measure.
2. In the Properties pane, under the Chart Labels, set the Custom Label property to Show.
   
   A new chart text item appears above the default measure or below the chart node member.
3. Do one of the following:
   
   - To change the text of the label, double-click the chart text item and type the text.
   - To change the source of the label text, select the new chart text item. In the Text Source section of the Properties pane, set the Source Type property.

   If nothing is specified in the new chart text item, the label appears blank.

   **Tip:** To remove a custom label, set the Custom Label property to Hide.

**Customizing the Axes of a Chart**

Chart axes are lines that border the chart area and provide a reference for measurements. They act as a vehicle for tick marks and scales and form a frame around the chart. The Y-axis is usually the vertical axis and contains data. The X-axis is usually the horizontal axis and contains categories.

To make your chart data easier to understand, you can do the following to customize each axis:

- show or hide the axis labels (data values)
- show or hide the axis line
- change the color, style, and weight of the axis line
- show or hide major and minor tick marks and specify where to display them

In HTML/PDF, if you have a report with a line chart, you can use the Include Zero For Auto Scale property to adjust the scale of the Y-axis of the chart. If the difference between your measure’s maximum/minimum values is much less than the difference between your measure’s minimum value and zero, using this property will make the chart Y-axis begin at something closer to your measure’s minimum value, making the line utilize all of the chart’s vertical space. In some cases this is significant in improving the readability of the chart.
Customize the Axes of a Current Default Chart
You can customize the axis labels, axis lines, and minor and major tick marks.

You can change the scale for the major and minor tick mark using the Axis Range property.

Procedure
1. Select the Y-axis or the X-axis of the chart.
2. To show or hide the axis labels, in the Properties pane, set the Axis Label property.
3. If your chart includes nested categories, to change how the category axis labels are displayed, set the Nested Label Display property.
4. To show or hide axis lines, in the Properties pane, double-click the Axis Line property, and select or clear the Axis line check box.
5. To change the color, style, and weight of the axis line, in the Properties pane, set the Axis Line property.
6. To show or hide minor and major tick marks and specify their location, in the Properties pane, set the Axis Line property.

Related concepts:
“Axes” on page 82
Axes are lines that provide references for measurement or comparison.

Customize the Axes of a Legacy Chart
You can customize the axis labels and axis line.

Procedure
1. Select the Y-axis or the X-axis of the chart.
2. To show or hide the axis labels, in the Properties pane, set the Axis Label property.
3. Select the Axis line check box.
4. Click Line color and choose a color for this axis line.
5. To show or hide axis lines, in the Properties pane, double-click the Axis Line property, and select or clear the Axis line check box.
6. To change the color, style, and weight of the axis line, in the Properties pane, double-click the Axis Line property.

Change the Axis Scale of a Current Default Chart
By default, IBM Cognos Report Studio automatically determines the minimum and maximum scale values for the axes in a chart. For example, a Y-axis showing revenue values might have an axis range of zero dollars to one million dollars. You can customize the axis scale, or range, to make your chart easier to understand.
You can specify the following for each axis:

- the maximum and minimum values of the range
- whether to include zero in an automatic axis range
- whether to use a logarithmic scale, if the values in the chart cover a very large range
- how often major and minor gridlines appear

Logarithmic scales can be useful when some of the data you are displaying is much less or much more than the rest of the data or when the percentage or ratio differences between values are important.

A logarithmic scale in Report Studio shows the base value of 10 raised to the power of a value. For example, 10 has a logarithm of 1 because 10 raised to the power of 1 is 10, 100 has a logarithm of 2 because 10 raised to the power of 2 is 100, and so on.

For example, the chart below is using a normal scale:

![Chart with a normal axis range](image)

Figure 10. Chart with a normal axis range

In this chart, the Y-axis shows values up to 2,000,000,000 with evenly spaced intervals of 400,000,000. However, the Outdoor Protection column is so much less than the other columns that it is difficult to compare them.

The following is the same chart using a logarithmic scale:
In this chart, the Y-axis still shows the same revenue values, but the intervals reflect a logarithmic scale, which increases exponentially. You can now compare all the product lines because none of the product lines have much higher or much lower values than the others.

Logarithmic scales can also be useful when the percentage or ratio differences between values are important. For example, if the logarithmic scale chart above represents data from 2011, and you add data from 2012 as a second set of colored bars, you could see the differences between the 2011 revenue and 2012 revenue. In a logarithmic scale, differences between values on the Y-axis represent the same percentage for each bar. So if the 2011 and 2012 data differs by the same distance for each product line, you could deduce that your revenue went up by the same percentage for each product line. This would not be clear on a normal scale.

Procedure
1. Select the Y-axis or the X-axis of the chart.
2. In the Properties pane, under General, double-click the Axis Range property.
3. To set a maximum value for this axis, under Maximum, click Manual, and then type a maximum value in the Manual box.

   Note: You can use either a positive or negative value as the maximum value.
4. To set a minimum value for this axis, under Minimum, click Manual, and then type a minimum value in the Manual box.

   Note: You can use either a positive or negative value as the minimum value.
5. To include zero on this axis, select the Include zero check box.
6. To use a logarithmic scale for this axis, select the Logarithmic scale check box.

   Note: If you choose to use a logarithmic scale, ensure that your chart consumers know that the scale is logarithmic by including that information in the axis title or in a note on the chart.
7. To set the position of major gridlines and tick marks, under Major interval, click Manual, and then type the distance between major gridlines and tick marks in the Manual box.
The distance between major gridlines and tick marks is measured in the units of that axis. For example, if the axis is revenue in dollars, type the dollar value in the Manual box.

8. To add minor gridlines, type the number of minor gridlines that you want to see between each major gridline in the Number of minor intervals box.

Related concepts:
- "Current Default Charts and Legacy Charts" on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Related tasks:
- "Customize the Axes of a Current Default Chart" on page 120
You can customize the axis labels, axis lines, and minor and major tick marks.
- "Show Gridlines in a Current Default Chart"
You can also show alternating bands of color in the chart background that correspond to your axis gridlines.

Showing Gridlines in a Chart
To make the data in a chart that includes axes easier to read, you can show horizontal and vertical gridlines. You can show gridlines for the major or minor intervals on the axes.

You cannot show gridlines for chart types that do not display axes, such as pie and donut charts.

Show Gridlines in a Current Default Chart
You can also show alternating bands of color in the chart background that correspond to your axis gridlines.

You can change the scale for the major and minor gridlines using the Axis Range property.

Procedure
1. Select the Y-axis or the X-axis of the chart.
2. To show alternating bands of color as your chart background, do the following:
   a. In the Properties pane, double-click the Gridlines property.
   b. Select the Show alternating color bands check box.
   c. Set the color and transparency of the first and second colors.
3. To show major gridlines as your chart background, do the following:
   a. In the Properties pane, double-click the Gridlines property.
   b. Select the Show major gridlines check box.
   c. Set the color, style, and weight of the major gridlines.
   d. If you are working with a gauge chart, specify the length of the gridline.
4. To show minor gridlines as your chart background, do the following:
   a. In the Properties pane, double-click the Minor gridlines property.
   b. Select the Show minor gridlines check box.
   c. Set the color, style, and weight of the minor gridlines.
   d. If you are working with a gauge chart, specify the length of the gridline.
   e. To specify by how many minor intervals to divide the major intervals, in the Properties pane, type a number in the Number of Minor Intervals property.
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Related tasks:
“Change the Axis Scale of a Current Default Chart” on page 120

By default, IBM Cognos Report Studio automatically determines the minimum and maximum scale values for the axes in a chart. For example, a Y-axis showing revenue values might have an axis range of zero dollars to one million dollars. You can customize the axis scale, or range, to make your chart easier to understand.

Show Gridlines in a Legacy Chart
Gridlines are the lines on a chart that show the intervals along the axes.

Procedure
1. Select the Y-axis or the X-axis of the chart.
2. In the Properties pane, under General, double-click the Gridlines property.
3. To specify the color, style, and weight of major gridlines do the following:
   a. Select the Show major gridlines check box.
   b. Click Line color and choose a color for the major gridlines.
   c. To specify the thickness of the gridlines, click Point size and select a line weight from the list.
   d. Click a line style in the Line style list.
4. To specify the color, style, and weight of minor gridlines:
   a. In the Properties pane, under General, double-click the Minor Gridlines property.
   b. Select the Show minor gridlines check box.
   c. Click Line color and choose a color for the minor gridlines.
   d. To specify the thickness of the gridlines, click Point size and select a line weight from the list.
   e. Click a line style in the Line style list.
   f. To change the transparency of the gridlines, type a percentage in the Transparency box.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Customize the Legend in a Current Default Chart

You can hide or show the legend and change its position relative to the chart area, chart body, or a data item. For example, in a bar chart showing revenue for each product line by country or region, you could use the expression [Country or Region] = ‘Canada’ to position the legend relative to the Canada bar.

Note: When you choose a preset legend position, that position appears inside any padding that you have added to the chart object. Customized legend positions do not include any chart padding.

Procedure
1. Select the chart object
2. In the Properties pane, under Chart Annotations, double-click the Legend property.
3. To show the legend on the chart, select the **Show Legend** check box.
4. To choose a preset legend position, click **Preset** and click a position from the diagram.
5. To choose a customized legend position, do the following:
   a. Click **Advanced** and click the ellipsis (...) button.
   b. To display the legend at a set distance from the sides of the chart area, from the **Anchor** list, click **Relative to Chart**.
   c. To display the legend at a set distance from the sides of the chart body, from the **Anchor** list, click **Relative to Chart Body**.
   d. To display the legend at a set distance from a data item, such as a specific pie slice in a pie chart, from the **Anchor** list, click **Report Expression**, click the ellipsis (...) button beside **Expression**, and enter an expression in the **Report Expression** dialog box.
   e. Set the horizontal and vertical distances from the anchor.

**Related concepts:**
"Current Default Charts and Legacy Charts” on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Customize the Items in the Legend of a Current Default Chart**

If your legend includes items that are too long, you can truncate long legend items at a specific number of characters.

For example, if you want an ellipsis (...) to appear at the end of each truncated legend item, type ... in the **Truncation text** box.

You can also show the values of the data items in your legend.

**Procedure**

1. Select the legend.
2. To truncate the legend
   a. In the **Properties** pane, under **General**, double-click the **Text Truncation** property.
   b. To specify the number of characters at which the legend items are truncated, click **Manual** and type the number of characters in the **Maximum characters** box.
   c. To shrink the font of the legend item text until all the text fits in the legend, select the **Shrink font as needed** check box.
   d. To specify some text to appear at the end of truncated legend items, in the **Truncation text** box, type the text that you want to appear at the end of truncated items.
3. To show the values of legend items within the legend
   a. In the **Properties** pane, under **General**, set the **Show Values** property. **First Value** and **Last Value** refer to the first and last item in the child set under the legend data item. **Minimum Value** and **Maximum Value** refer to the lowest and highest value in the child set under the legend data item.
   b. If you want to change the separator between the legend item and value, type a new separator in the **Legend Separator** property.
4. To customize the title of the legend
   a. Click the default legend title area in the work area.
   b. In the **Properties** pane, under **General**, set the **Default Title** property to No.
c. Double-click the default legend title area in the work area and type the new title.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Customize the Items in the Legend of a Legacy Chart**

If your legend includes items that are too long, you can truncate long legend items at a specific number of characters. For example, if you want an ellipsis (...) to appear at the end of each truncated legend item, type ... in the **Truncation text** box.

You can also show the values of the data items in your legend and customize the legend title.

**Procedure**

1. Select the legend.
2. To truncate the legend text, do the following:
   a. In the **Properties** pane, under **General**, set the **Auto Truncation** property to **Yes**.
   b. In the **Properties** pane, under **General**, set the **Auto Truncation** property to **Yes**.
   c. In the **Maximum Characters** property, type the number of characters to appear before the text is truncated.
   d. In the **Truncation Text** property, type the characters to append when the text is truncated.
   e. To shrink the font of the legend item text until all the text fits in the legend, select the **Shrink font as needed** check box.
   f. To specify text to appear at the end of truncated legend items, type it in the **Truncation text** box.
3. To show the values of legend items within the legend, do the following:
   a. In the **Properties** pane, under **General**, set the **Show Legend Values** property.
      First Value and Last Value refer to the first and last item in the child set under the legend data item. Minimum Value and Maximum Value refer to the lowest and highest value in the child set under the legend data item.
   b. If you show multiple values and want to change the separator between the values, type a new separator in the **Separator** property.
4. To customize the title of the legend, do the following:
   a. Click the default legend title area in the work area.
   b. In the **Properties** pane, under **General**, set the **Default Title** property to **No**.
   c. Double-click the default legend title area in the work area and type the new title.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Adding a Note to a Chart**

Add a note to a chart to provide additional detail. Notes appear as text in a chart.
By default, notes are aligned with the upper left corner of the chart object. When you add a note to a chart, you can set the note’s position relative to the sides of the chart area or chart body. You can also use a report expression to position the note next to a data item within the chart. For example, in a bar chart showing revenue for each product line by country or region, you could type [Country or Region]='Canada' to position the note relative to the Canada bar.

Notes overwrite whatever is under them so you must position them properly.

If you apply more than one note, ensure that each note has a different position in the report so that they do not overwrite each other. You can also specify the order that they should be drawn in when the report runs. If you have two notes with the same coordinates, the first one in the list is drawn first and the next one is drawn on top of the first.

**Tip:** The Top 10 Retailers for 2011 sample report in the GO Data Warehouse (analysis) package includes a note.

**Related concepts:**
- Appendix C, “Sample Reports and Packages,” on page 507

Sample reports are included with IBM Cognos Business Intelligence. When installed, you can find them in the **Public Folders** tab in IBM Cognos Connection.

### Add a Note to a Current Default Chart
You can add and position multiple notes on your chart.

#### Procedure

1. Select the chart object.
2. In the **Properties** pane, under **Chart Annotations**, double-click the **Notes** property.
3. Click the new button.
4. Type the note text in the **Text** box and click **OK**.

   **Tips:** To delete a note, click the note and click the delete button. To move a note up or down in the list, click the up or down arrows.

5. In the **Notes** dialog box, click **OK**.
6. To position the note, do the following:
   a. Click the note icon in the list of notes.
   b. In the **Properties** pane, under **Positioning**, double-click the **Position** property.
   c. To align the note horizontally, click the left, center, or right alignment button.
   d. To align the note vertically, click the top, middle, or bottom alignment button.
   e. To set the margins around the chart body, type margin values and choose margin units.
   f. To display the note at a set distance from the sides of the chart area, from the **Anchor** list, click **Relative to Chart**.
   g. To display the note at a set distance from the sides of the chart body, from the **Anchor** list, click **Relative to Chart Body**.
To display the note at a set distance from a data item, from the Anchor list, click Report Expression, click the ellipsis (...) button beside Expression, and enter an expression in the Report Expression dialog box.

To edit the text, double-click the text next to the note object in the chart.

Related concepts:
"Current Default Charts and Legacy Charts" on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Add a Note to a Legacy Chart
You can add and position multiple notes on your chart.

Procedure
1. Click the chart object.
2. In the Properties pane, under Chart Annotations, double-click the Notes property.
3. Click the new button.
4. Type the note text in the Text box and click OK.

Tip: To delete a note, click the note and click the delete button. To move a note up or down in the list, click the up or down arrows.

5. In the Notes dialog box, click OK.

6. Click the note icon and set the note position, size, and border in the Properties pane.
   You set the location of the note by defining number of pixels from the bottom left corner of the chart area.

Adding a Baseline to a Chart
Baselines are horizontal or vertical lines that cut through the chart to indicate major divisions in the data. For example, you can add a baseline to show a sales quota or break-even point.

Each baseline represents a value on an axis.

Depending on the type of chart, you can use the following options to position the baseline.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric Value</td>
<td>Uses a static numeric value.</td>
</tr>
<tr>
<td>Query Calculation</td>
<td>Uses a query calculation from the chart query or from a different query. For more information, see &quot;Using Relational Calculations&quot; on page 248.</td>
</tr>
<tr>
<td>Layout Calculation</td>
<td>Uses a layout calculation. For more information, see &quot;Using Relational Calculations&quot; on page 248.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Category Index</td>
<td>Uses the index value of the data item in the categories axis. For example, a Category index value of 1 indicates that the baseline is located at the first data item. This is the default. <strong>Note:</strong> This option applies only to the current default charts, and does not apply to the legacy charts.</td>
</tr>
<tr>
<td>Member Value</td>
<td>When working with dimensional data sources, uses a position relative to a member. For example, a member calculation that uses an expression similar to [Query1].[Current year]=2011 places the baseline in the middle of the chart object that represents 2011. To place the baseline between 2011 and 2012 on the chart, a half member width to the right, set the Member Offset property to 50%. To place the baseline between 2010 and 2011, set the Member Offset property to -50%. You can also type in 100, -200, and so on to place the baseline on the chart.</td>
</tr>
</tbody>
</table>
| Statistical Limit   | **Statistical maximum** uses the following expression:  
  \[25\text{th percentile value} - 1.5 \times (75\text{th percentile value} - 25\text{th percentile value})\]  
  For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical minimum is -5 \[2.5 - 1.5(5) = -5\].  
  **Statistical minimum** uses the following expression:  
  \[75\text{th percentile value} + 1.5 \times (75\text{th percentile value} - 25\text{th percentile value})\]  
  For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical maximum is 15 \[7.5 + 1.5(5) = 15\].  
  **Statistical minimum** and **Statistical maximum** use percentiles to determine values, so the baseline might not appear on the chart if its value is off the axis. |
| Mean                | Uses the statistical mean plus or minus a number of standard deviations based on all charted data values on the specified axis. |
| Percentile (%)      | Uses a specified percentile. |
| Percent on Axis (%) | Uses a percentage of the full range of the axis. For example, if the axis range is -500 to 1100, a Percent on Axis value of 25% puts the baseline at -100 (25% of the range, 1600). |
If you apply more than one baseline, you can specify the order in which they should be drawn when the report runs. If you have two baselines with the same coordinates, the first one in the list is drawn first and the next one is drawn on top of the first one.

By default, the baseline and its label appear in the legend.

Baselines do not support color transparency.

Tip: The Positions to Fill sample report in the GO Data Warehouse (analysis) package includes a baseline.

Related concepts:

Appendix C, “Sample Reports and Packages,” on page 507
Sample reports are included with IBM Cognos Business Intelligence. When installed, you can find them in the Public Folders tab in IBM Cognos Connection.

Add a Baseline to a Current Default Chart
When working with current default charts, you can define baselines by values on the numeric axis or the category axis.

Procedure
1. Click the chart object.
2. Depending on your chart and the type of baseline you want to add, in the Properties pane, under Chart Annotations, double-click the Numeric Baselines or Category Baselines property.
3. Click the new button and choose the type of baseline from the list.
4. Specify the necessary criteria for the baseline position.
5. Under Baseline Properties, type a label for the baseline and specify the line style.
   Tip: To remove the baseline from the legend, delete the label.
6. If you add more than one baseline, specify their order using the up and down arrows.
   The new baselines appear in the Markers, notes, baselines, and trendlines box.
7. To change the label that appears next to the baseline, under the Markers, notes, baselines, and trendlines box, double-click the placeholder text next to the baseline icon and type your text.
8. To change the line style, select the chart and in the Properties pane, under Chart Annotations, double-click the Baselines, Numeric Baselines, or Category Baselines property.
9. To delete a baseline, select the baseline icon and click the delete button.

Related concepts:

“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Add a Baseline to a Legacy Chart
When working with legacy charts, you can define baselines by the values on the numeric axis.
Procedure
1. Click the chart object.
2. In the Properties pane, under Chart Annotations, double-click the Baselines property.
3. Click the new button and choose the type of baseline from the list.
4. Specify the necessary criteria for the baseline position.
   If you use a calculation to determine the baseline, define an expression.
5. If you add more than one baseline, specify their order using the up and down arrows.
6. Click OK.
   A baseline icon appears in the Markers, notes, and baselines box.
7. To define the line style, click the baseline icon and set the Line Styles property.
8. To define the text style, click the baseline text and make changes in the Properties pane.
9. To delete a baseline, click its baseline icon and click the delete button.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Adding Markers to Charts
Markers are symbols that you add to a chart to designate points of significance that can help you analyze or understand the data.

For example, you might want to add a marker to designate the time when a significant event happened, such as the date when a new product was launched.

You can add a marker to combination, progressive column, Pareto, scatter, and bubble charts.

When you define the position of the marker, you define the X- and Y-axes coordinates for the placement of the symbol. You can position the marker in the chart using the following options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric Value</td>
<td>Uses a static numeric value.</td>
</tr>
<tr>
<td>Query Calculation</td>
<td>Uses a query calculation from the same query or from a different query. For more information, see “Using Relational Calculations” on page 248.</td>
</tr>
<tr>
<td>Layout Calculation</td>
<td>Uses a layout calculation. For more information, see “Using Relational Calculations” on page 248.</td>
</tr>
<tr>
<td>Option</td>
<td>Position</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Statistical Minimum</td>
<td>Uses the following expression:</td>
</tr>
<tr>
<td></td>
<td>[25\text{th percentile value} - 1.5 \ast (75\text{th percentile value} - 25\text{th percentile value})]</td>
</tr>
<tr>
<td></td>
<td>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical minimum is -5 [2.5 -1.5(5) = -5].</td>
</tr>
<tr>
<td></td>
<td>Statistical minimum uses percentiles to determine values, and might not always appear in the chart if the values are off the axis.</td>
</tr>
<tr>
<td>Statistical Maximum</td>
<td>Uses the following expression:</td>
</tr>
<tr>
<td></td>
<td>[75\text{th percentile value} + 1.5 \ast (75\text{th percentile value} - 25\text{th percentile value})]</td>
</tr>
<tr>
<td></td>
<td>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical maximum is 15 [7.5 +1.5(5) = 15]. Statistical maximum uses percentiles to determine values, so the marker might not always appear on the chart if its value is off the axis.</td>
</tr>
<tr>
<td>Mean</td>
<td>Uses the statistical mean plus or minus a number of standard deviations based on all charted data values on the specified axis.</td>
</tr>
<tr>
<td>Percentile</td>
<td>Uses a specified percentile.</td>
</tr>
<tr>
<td>Percent on Axis</td>
<td>Uses a percentage of the maximum axis value.</td>
</tr>
</tbody>
</table>

If you apply more than one marker, you can specify the order in which they should be drawn when the report runs. If you have two markers with the same coordinates, the first one in the list is drawn first and the next one is drawn on top of the first.

**Add a Marker to a Current Default Chart**

Markers are symbols that you add to a chart to designate points of significance that can help you analyze or understand the data.

**Note:** Current default scatter charts have a different set of steps.

**Procedure**

1. Select the chart object.
2. In the Properties pane, under Chart Annotations, double-click the Markers property.
3. Click the new button \[\text{新增}^{\text{新增}}\], and then set the marker properties.
4. In the Based on box, select the chart object that will determine the marker position.
5. In the Numeric position box, specify how you want to define the position on the numeric (Y) axis.
6. In the **Category position** box, specify how you want to define the position on the category (X) axis.

7. In the **Marker label** box, type the label that you want to give to the marker.

8. In the **Marker size** box, specify the size of the marker symbol.

9. To specify the color, fill, and shape of the marker symbol, click the **Color and style** box.

10. If you add more than one marker, specify their order by using the up and down arrows.
    The new markers appear in the **Markers, notes, and baselines** box.

    **Tip:** To delete a marker, click the marker icon and text and click the delete button.

11. Run the report.

**Related concepts:**

"**Current Default Charts and Legacy Charts**" on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Add a Marker to a Current Default Scatter Chart**

Markers are symbols that you add to a chart to designate points of significance that can help you analyze or understand the data.

**Procedure**

1. Select the scatter chart object.
2. In the **Properties** pane, under **Chart Annotations**, double-click the **Markers** property.
3. Click the new button, and then set the marker properties.
4. In the **X-axis position** box, specify how you want to define the position on the X-axis.
5. In the **Y-axis position** box, specify how you want to define the position on the Y-axis.
6. In the **Marker label** box, type the label that you want to give to the marker.
7. In the **Marker size** box, specify the size of the marker symbol.
8. To specify the color, fill, and shape of the marker symbol, click the **Marker color and style** box.
9. If you add more than one marker, specify their order using the up and down arrows.
    The new markers appear in the **Markers, notes, and baselines** box.

    **Tip:** To delete a marker, click the marker icon and text and click the delete button.

**Related concepts:**

"**Current Default Charts and Legacy Charts**" on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Add a Marker to a Legacy Chart**

Markers are symbols that you add to a chart to designate points of significance that can help you analyze or understand the data.
Procedure

1. Select the chart object.
2. In the Properties pane, under Chart Annotations, double-click the Markers property.
3. Click the new button, and then set the Numeric position type, Marker label, Marker shape, Numeric value, and Color properties.
4. If you add more than one marker, specify their order by using the up and down arrows.

   The new markers appear in the Markers, notes, and baselines box.

   Tip: To delete a marker, click the marker icon or marker label and click the delete button.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Display Trendlines in Current Default Charts

You can display or hide trendlines in the current default bar, line, area, bubble, or scatter charts.

Trendlines, also known as lines of best fit or regression lines, graphically illustrate trends in data series and are commonly used when charting predictions. A trendline is typically a line or curve that connects or passes through two or more points in the series, showing a trend.

You can specify the following types of trendlines:

- **Linear**
  Use a linear trendline when your data increases or decreases along a straight line at a constant rate. For example, if your chart displays a steady increase in revenue by product line over time, you could use a linear trendline.

- **Exponential**
  Use an exponential trendline when your data values increase or decrease exponentially, or at an increasingly higher or lower rate. For example, if your chart displays an exponential increase in revenue by product line over time, you could use an exponential trendline.

- **Polynomial**
  Use a polynomial trendline when your data values both increase and decrease. For example, if your chart displays both increases and decreases in revenue by product line over time, you could use a polynomial trendline.

- **Logarithm** or **Natural Logarithm**
  Use a logarithmic trendline when your data values increase or decrease rapidly and then level out. For example, if your chart displays a rapid decrease in revenue by product line over time and then a plateau, you could use a logarithmic trendline.

- **Power**
  Use a power trendline when your data values increase or decrease in a curve at a steady rate. For example, if your chart displays a steady increase in revenue by product line over time and your data points will fit a curved line, you could use a power trendline.
• **Moving Average**

Use a moving average trendline when your data values fluctuate and you want to smooth out the exceptions to see trends. For example, if your chart displays wild fluctuations in revenue by product line over time, but you know that some data points are exceptions, you could use a moving average trendline.

If you are not sure which trendline type to use, try each type to see which one best fits most of your data points. For example, a linear trendline will not fit most points on a scatter chart with widely spread data points.

Trendlines do not support color transparency.

**Procedure**

1. Select the chart object.
2. In the Properties pane, under Chart Annotations, double-click the Trendlines property.
3. Click the new button and click a trendline type.
4. Define the trendline by specifying the following options. The options that are available depend on the type of trendline you chose.
   - To set the order, or degree, of a polynomial trendline, in the Order box, type a value between 2 and 6.
   - To set the number of periods to go back in a moving average trendline, in the Periods box, type a value.
   - If you have more than one series on your chart, in the Based on list, click the data you want to use for the trendline.
   - Click either One trendline for all series items or A trendline for each series item.
   - To customize the style of the trendlines, click Line Styles and customize the line color, weight, and style.
   - To customize the trendline labels in the legend, click Label and choose None, Automatic, or Custom.
   - To display the trendline equation, click Show equation, and then click Show in legend or Show on chart, and then click Position to define the equation’s position on the chart.
   - To display the R-squared value of the trendline, click Show R-squared value, and then click Show in legend or click Show on chart and then click Position to define the position of the value on the chart.

**Related concepts:**

"Current Default Charts and Legacy Charts" on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Display Regression Lines in Legacy Scatter or Bubble Charts**

When using legacy charts, you can display or hide regression lines on scatter charts and bubble charts. Regression lines, also known as lines of best fit or trend lines, graphically illustrate trends in data series. Regression lines are commonly used when charting predictions. A regression line is typically a line or curve that connects or passes through two or more points in the series, showing a trend. Legacy scatter and bubble regression lines are calculated using a numerical algorithm that isn’t guaranteed to be optimal. Regressions may or may not match current default chart regressions.
You can specify the following types of regression lines to determine the position and slope of the line:

- **Linear**
  Use a linear regression line when your data increases or decreases at a consistent rate. For example, if your chart displays a steady increase in revenue by product line over time, you could use a linear regression line.

- **Common Log or Natural Log**
  Use a logarithmic regression line when your data increases or decreases rapidly and then levels out. For example, if your chart displays a rapid decrease in revenue by product line over time and then a plateau, you could use a logarithmic regression line.

- **Exponential**
  Use an exponential regression line when your data increases or decreases exponentially, or at steadily increasing or decreasing rate. For example, if your chart displays an exponential increase in revenue by product line over time, you could use an exponential regression line.

- **Polynomial Fit**
  Use a polynomial fit regression line when your data both increases and decreases. For example, if your chart displays both increases and decreases in revenue by product line over time, you could use a polynomial fit regression line.

If you are not sure which regression line type to use, try each type to see which one best fits most of your data points. For example, a linear regression line is not going to fit most points on a scatter chart with widely spread data points.

**Procedure**

1. Select the scatter or bubble chart.
2. In the **Properties** pane, under **Chart Annotations**, set the **Regression Line** property to **Show**.
3. Click the regression line icon in the report layout.
4. In the **Properties** pane, under **General**, set the **Line Styles**, **Regression Type**, and **Number of Regression Lines** properties.

**Related concepts:**

“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Showing data values in current default charts**

You can show the data labels or data values within the chart so that the data values are clearer.

For example, you can display the data values above each column in a column chart to show the exact height of each column.

**Procedure**

1. For a bar, column, line, or area chart, under **Series**, select the chart type icon.
2. For a bubble, scatter, Pareto, or progressive chart, click the chart.
3. In the **Properties** pane, under **Chart Labels**, double-click the **Show Values** property.
4. For bar, column, line, area, Pareto, or progressive charts, to specify the data label format, in the **Values** list, select what values to display.
   - **None** does not display data values.
   - **Absolute** displays the absolute value of the data.
   - **Cumulative** displays the cumulative value of the data.

5. For bubble or scatter charts, to specify the data label format, in the **Show** box, select whether to show values or values and labels for the category, series, and measure.

6. For bar, column, line, area, Pareto, or progressive charts, to show lines that point from data labels to data marker they apply to, select the **Show leader lines** check box.

7. For bubble or scatter charts, to show lines that point from data labels to data marker they apply to, select the **Leader lines** check box.

8. To specify where values and labels are to be rendered in the chart, click the **Value Location** list and choose the location that you want.

9. To specify how to display the labels if their positions on the chart overlap, in the **Collision Mode** list, click one of the following modes:
   - **None** specifies that labels appear in default positions and might overlap.
   - **Normal** (for pie and donut charts) specifies that labels are placed just above their corresponding data markers or chart objects. There is no collision detection, so labels can overlap.
   - **Coarse Stagger** specifies that labels are placed close to their data markers and staggered so that they do not overlap. This collision mode takes less time to render than **Fine Stagger** but might result in labels that are farther away from their corresponding data markers.
   - **Fine Stagger** specifies that labels are staggered so that they do not overlap. The labels are as close to the data markers as possible without overlapping. This collision mode takes more time to render than **Coarse Stagger** but might result in labels that are closer to their corresponding data markers.

**Showing data values in current default pie and donut charts**

You can show the data labels or data values within the chart so that the data values are more clear.

For example, in a pie chart, show the data values within each pie slice, so that you know the exact size of each pie slice.

**Procedure**

1. Select the chart object.
2. In the **Properties** pane, double-click the **Show values** property.
3. To display data labels for each slice of the pie, under **Show**, select the **Slice names** check box.
4. To show lines pointing from data labels to the slices they apply to, select the **Show leader lines** check box.
5. To specify the data label format, in the **Values** list, select what values to display.
   - **Hide** does not display data values.
   - **Absolute** displays the absolute value of the data.
   - **Percentage** displays the slice’s percentage of the whole pie.
• **Absolute and Percentage** displays the slice’s percentage of the whole pie as an absolute value.

6. In the **Position** list, select the placement of data labels.

**Showing data values in other current default charts**

You can show the data labels or data values within the chart so that the data values are more clear.

For example, you can display the data values above each bubble in a bubble chart to show the exact value of each bubble.

You can show data values for the following chart types: Pareto, progressive, scatter, and bubble.

**Procedure**

1. Select the **chart object**.
2. In the **Properties** pane, under **Chart Labels**, double-click the **Show Values** property.
3. For each of the categories, series, and measures, select whether to show only values, values and labels, or neither.
4. To show lines pointing from data labels to the data marker they apply to, select the **Leader lines** check box.
5. To specify how to display the labels if their positions on the chart overlap, in the **Collision Mode** list, click one of the following modes:
   - **None** specifies that labels appear in default positions and might overlap.
   - **Normal** (for pie and donut charts) specifies that labels are placed just above their corresponding data markers or chart objects. There is no collision detection, so labels can overlap.
   - **Coarse Stagger** specifies that labels are placed close to their data markers and staggered so that they do not overlap. This collision mode takes less time to render than **Fine Stagger** but might result in labels being farther away from their corresponding data markers.
   - **Fine Stagger** specifies that labels are staggered so that they do not overlap. The labels are as close to the data markers as possible without overlapping. This collision mode takes more time to render than **Coarse Stagger** but might result in labels being closer to their corresponding data markers.

**Related concepts:**

“Current Default Charts and Legacy Charts” on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Define Query Context When Customizing Legend Entries, Legend Titles, or Axis Labels**

You want to use a revenue expression as a chart legend title. If you get an error message saying that the query context of a layout object cannot be determined, you must define the property list for the item to which it refers. You must first add the desired data item to the query before you can define its property list.

**Procedure**

1. Open the chart to customize.
2. Pause the pointer over the query explorer button and click the query.
3. From the Source tab, drag the desired item to the Data Items window to add it to the query.

4. Pause the pointer over the page explorer button and click the chart page.

5. From the Source tab, drag the desired item to the layout object.
6. In the Properties pane, under Data, double-click the Properties property.
7. Select the check box for the data item to define.

Related tasks:
“Specify a List of Data Items for an Object” on page 284
Specify the list of data items for an object when you must reference a data item that is in a query but is not in the layout.

Summarize Small Slices, Bars, or Columns in Current Default Charts
You can summarize the smaller slices or bars in charts to avoid having many tiny slices or bars. For example, if your pie chart shows revenue by product and 10 of your products have less than 1% of the pie, you can summarize these 10 slices into one larger slice and name that slice Other.

Similarly, you can summarize small items in a column, bar, area, and line chart.

You cannot summarize small slices or items in charts that have matrix edges or in charts that have multiple numeric axes.

Procedure
1. Select the chart object.
2. In the Properties pane, under General, double-click the Summarize Small Slices or Summarize Small Items property.
3. To summarize small slices or items up to a maximum number, select the Maximum number of slices or Maximum number of items check box and type the maximum number.
4. To summarize all the slices or bars that are smaller than a specific value, select the Summarize slices smaller than a value or Summarize items smaller than a value check box, type a value that represents the upper size limit, and choose whether the value is a percentage or absolute.
5. Under Small slice summarization type or Small item summarization type, choose whether to summarize the slices or items as a Total or Average.

Note: Averages are calculated using only the summarized items in the report.
6. Under Slice label or Item label, type a label for the one large slice, bar, area, or line that summarizes the smaller ones.

Related concepts:
“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Customizing Lines and Data Points in a Line Chart
You can customize the lines in a line chart to show only lines, only data points, or both. Data points represent series values for each category on the Y-axis. You can show special data markers that represent statistically significant values, such as open, high, low, and close values.
You can also add markers at other positions on the chart. For more information, see “Add a Marker to a Current Default Chart” on page 132.

You can also change the shape of the line that connects the data points to one of the following:

- **Point to Point** shows data points connected by straight lines.
- **Step at Point** shows data points connected by steps that start and end at the data points.
- **Step Between Points** shows data points connected by steps that start and end between the data points.
- **Smooth** shows data points connected by smooth curves.

Related tasks:
“Add a Marker to a Current Default Chart” on page 132

Markers are symbols that you add to a chart to designate points of significance that can help you analyze or understand the data.

**Customize Lines and Data Points in a Current Default Line Chart**

You can change the color and shape or the line or data markers in a line chart. You can also show or hide data labels, the line, and data markers.

You cannot show value markers when the **Data points** option is selected or when the configuration of the line chart is stacked or 100 percent stacked.

**Procedure**

1. Select the chart object.
2. In the chart area, under **Series**, click the line chart icon to view the line properties.
3. To select whether to show only the line, the line and data markers, or only the data markers, in the **Properties** pane, select an option from the **Line and Markers** list.
4. To show special data markers, double-click the **Value Markers** property. Select the check box for the special data markers that you want to add and for each marker, specify the color and shape.
   The options you specify in **Value Markers** overwrite the options for **Line and Markers**.
5. To change the shape of the line that connects data markers, select an option from the **Line Shape** list.
6. To show data labels for the data points, double-click the **Data Labels** property.
7. To change the color of the lines, double-click the **Palette** property.

**Related concepts:**
“Current Default Charts and Legacy Charts” on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

“Customizing the Color Palette of a Chart” on page 108

You can use the chart palette to control the colors or patterns used in the columns, lines, data markers or areas in a chart. For example, if “Telephone” is the first in a data series of order methods, and you want it to appear in blue, use the palette to make the first item in the series blue.

**Customize Lines and Data Points in a Legacy Line Chart**

You can change the color and shape or the line or data markers in a line chart. You can also show or hide data labels, the line, and data markers.
You cannot show value markers under the following conditions:

- The **Data points** option is selected.
- The configuration of the line chart is stacked or 100 percent stacked.
- The **Show line** property is set to **No**, creating a point chart.

**Procedure**

1. Select the chart object.
2. In the chart area, under **Series**, click the line chart icon to view the line properties.
3. To select whether to show only the line, the line and data markers, or only the data markers, in the **Properties** pane, under **General**, select an option from the **Line Type** list.
4. To show or hide the line, select an option from the **Show line** list.
5. To show data points or value markers, double-click the **Show Data Points** property:
   - To show or format data points, click **Data points** and specify the point shape and point size.
     The point color is defined by the chart palette.
   - To show or format value markers, click **Value markers** and specify the markers to show. For each marker, specify the shape and marker size. Click **Color** to change the marker color.

**Tip:** To remove all data points and value markers, in the **Show Data Points** dialog box, select **None**.

**Related concepts:**

- "Current Default Charts and Legacy Charts" on page 79
- "Customizing the Color Palette of a Chart" on page 108

You can use the chart palette to control the colors or patterns used in the columns, lines, data markers or areas in a chart. For example, if "Telephone" is the first in a data series of order methods, and you want it to appear in blue, use the palette to make the first item in the series blue.

**Customize a Current Default Combination Chart**

Combination charts show data series using two or more types of charts - area, bar, and line. The different charts are overlaid on top of each other. You can customize the order in which the charts appear along with the type of charts and their configurations.

You can also customize which numeric axes to show and which chart to show on each axes.

**Procedure**

1. Select the combination chart object.
2. In the **Properties** pane, under **General**, double-click the **Combinations** property.
3. Under **Numeric axes**, select which axes to show.
4. Under **Combinations**, add or remove data series.
5. If you want to change the order in which the series appear, use the up and down arrows.
The series appear in the order they are listed. Each chart appears in the
foreground of any previous charts.

6. If you want to change the configuration type of the series, such as changing
clustered bars to stacked bars, under **Combinations**, select the series, click the
edit button and select the type.

7. If you use the same data series for multiple charts and want to synchronize the
data marker colors, under **Color & Background**, set the **Series Color** property
to **Match**.

**Related concepts:**
“Current Default Charts and Legacy Charts” on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Create a Matrix of Current Default Charts**

You can view a complex chart that includes nested series or categories into a
matrix, or crosstab, that shows multiple small charts arranged in rows and
columns.

The charts in the rows represent the outer nested levels of the series and categories
and, the charts in the columns represent the categories. Each data item in the outer
nested levels of the series and categories becomes a separate chart. The numeric
scale of all the charts is the same so that you can easily compare.

When working with pie, gauge, and bullet charts, if you include a data item in the
categories, you automatically create a matrix of charts. One chart appears for each
data item in the category. When working with progressive and bullet charts, if you
include a data item in the series, you also automatically create a matrix of charts.

For example, the following column chart shows the revenue for each year (in the
categories or X-axis) for all the regions and product lines (in the series or Y-axis).
This chart is very complex and difficult to understand.

![Complex column chart](image)
The following chart contains the same series and categories. However, when converted to a matrix of charts, the information is much easier to analyze. The columns show charts for each year and the rows show charts for each the product lines. The bars represent the revenue for each region.

**Figure 13. Matrix chart**

**Procedure**

1. Select the chart object.
2. In the Properties pane, under General, double-click the Matrix rows and columns property.
3. Select the Show outer nested series as matrix rows or Show outer nested categories as matrix columns check boxes (or both).
4. In the Matrix levels box, select the number of nested levels to include in the matrix columns or rows.

   The remaining nested levels are represented in the chart bodies in the matrix if the chart supports additional categories. The pie, gauge, and bullet charts do not support additional categories. The bullet and progressive column charts do not support additional series.

   For example, in the above chart, Regions are nested under Product Line. In the matrix chart, a Matrix level of 1 is specified. Therefore, Product Line appears as the matrix rows (series) and Regions appear within the chart bodies.

5. If you want labels for the matrix rows and columns to appear on each chart, select the Show row labels or Show column labels check box.
6. In the Labels location list, select where the nested labels should appear on each chart.
The default position is on the left for rows and on the bottom for columns.

7. If the labels are too long, click **Truncation** and specify where to truncate text or select the **Size fonts automatically** check box to resize the text to fit.

8. To change the font, color, and data format of the labels, click **Style**.

9. To show the title in the row or column axis, select the **Show row title** or **Show column title** check box.

10. If you want to hide or show the axes, select the axis object, and in the **Properties** pane, under **Miscellaneous**, set the **Show in Matrix** property.

11. If your matrix chart includes only rows or columns, and you want them to wrap, select the **Wrap rows or columns if possible** check box.

12. If you want to show the axes and axis labels for each small chart in the matrix, select the **Repeat row and column axis labels** check box.

   When this option is cleared, axes and axis labels appears only along the outer edge of the matrix.

**Related concepts:**

"Current Default Charts and Legacy Charts" on page 79

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Customize a Current Default Gauge Chart**

When you create a gauge chart, you can choose from a variety of chart templates, which offer different shape, axis, and border options.

You can customize the following aspects of your gauge chart. As you modify properties, the chart preview shows you what your chart will look like.

![Gauge chart diagram](image)

Unless indicated in the user interface, all sizes are a percentage of the maximum allowed.

If your gauge chart includes a border, long axis labels, such as 250,000,000, may overlap the gauge border and be difficult to read. To avoid this problem, customize the data format of your gauge chart measure and reduce the scale so that less zeros are displayed. Or change the gauge border color or size or remove the border.

**Procedure**

1. Click the gauge chart object.
2. If you want to change the shape of the gauge, modify the start and end angles of the gauge axes and border, as follows:
   a. In the Properties pane, under General, double-click the Gauge Axes property and specify the start and end angles and the direction of the axes.
   b. Double-click the Gauge Border property and specify the style and the start and end angles of the border.
      For example, a border start angle of 0 degrees and end angle of 180 degrees produces a semi-circular gauge.

3. If you want to add an additional axes to a gauge chart, do the following:
   a. In the Properties pane, under General, double-click the Gauge Axes property.
   b. Click the new button \( \text{new} \) and specify the start and end angles and the direction of the new axis.

4. If you want to change the size, shape, and color of the center pivot point, double-click the Gauge Pivot property and specify the style.

5. If you want to change the indicators in the gauge axis, click the Gauge Axis object in the chart and do the following:
   a. To change the needle, under Axes, double-click the Gauge Needle property and specify the style.
   b. To change, add, or remove color bands that indicate the data range positions, under Color & Background, double-click the Gauge Axis Colors property and specify the color palette.
   c. If you want to change the size or thickness of the gauge axis and color bands, under Axes, specify a percentage size for the Gauge Axis Inner Radius and Gauge Axis Outer Radius properties.
   d. If you want to change the gridlines, under General, double-click the Gridlines or Minor Gridlines properties and specify the style.
   e. If you want to change the appearance of the gauge axis line, under General, double-click Axis Line.

Related concepts:
- “Current Default Charts and Legacy Charts” on page 79
- IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.
- “Gauge Charts” on page 94

Customize a Legacy Gauge Chart

You can customize the colors and threshold boundaries of the gauge axis areas and whether they use numbers instead of percentages. You can also change the colors of the gauge face and outline and hide gauge labels. By default, the legacy gauge chart uses a band divided into thirds that runs from green to red.

Procedure

1. Click the gauge chart object.
2. If you want to customize the color of the gauge face or outline, in the General section of the Properties pane, double-click the Face Color or Dial Outline Color property.
3. If you want to hide the gauge labels, in the Chart Labels section of the Properties pane, set the Gauge Labels property to Hide.
4. If you want to customize the colors and boundaries of the gauge areas, do the following:
   a. In the **Color & Background** section of the **Properties** pane, double-click the **Gauge Palette** property.
   b. To change the color of a boundary area, under **Palette**, select the color, click **Color**, specify the color properties, and click **OK**.
   c. To change the value of a boundary area, under **Palette**, select the boundary value, and type a new value.
   d. To add a new boundary area, click the new button.
   e. To choose a pre-defined threshold style, click the **Palette** drop-down menu. You can choose whether to use discrete colors or continuous colors.
      To use a number instead of a percentage as a boundary, clear the **Percentage** check box, and then type a number in the **Numeric boundary** box.

   Related concepts:
   - “Current Default Charts and Legacy Charts” on page 79
     IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.
   - “Gauge Charts” on page 94
     Gauge charts, also known as dial charts or speedometer charts, use needles to show information as a reading on a dial.

### Define Color by Value in Current Default Scatter and Bubble Charts

In a scatter or bubble chart you can specify that data points or bubbles appear in different colors based on an additional measure or fact. The color of the points or bubbles adds an additional aspect to the chart to assist you in finding relationships in large amounts of data.

For example, the following bubble chart shows the relationship between the unit cost and unit sale price. The size of the bubbles shows the gross profit and the color of the bubbles shows whether the quantity sold is above (yellow) or below (green) 1,000,000 units.
In a bubble chart, you can combine the color by value and bubble size to create a more meaningful chart.

**Procedure**

1. Drag a measure to the **Color** drop zone under **Measures** and then select the measure.
2. In the **Properties** pane, under **Color & Background**, double-click the **Color by Value** property.
3. To define the colors by percentages instead of actual values, select the **Percentage** check box. For example, if the **Percentage** check box is selected and your values range from 25 (red) to 50 (green), then the bottom 25 percent of values will be red, the top 50 percent of values will be green, and the values between 25 and 50 percent will be an interpolated color, such as yellow.
4. If you want to use a preset color palette, click **Palette**, and select the palette that you want to use.
5. If you want to customize a palette color or boundary value, select the palette entry in the **Palette** box and specify the color, transparency, and type a new boundary value.
6. To add a new palette entry, select the palette entry below which you want to add the new entry, click the new palette entry button ![Plus](image), and click **Color**.
7. If your chart has lines or markers, under **Style**, set the marker shape, line style, and line weight.
8. To choose a color and transparency for missing or null values, under **Missing Values**, click **Color** and type a value in the **Transparency** box.
Specify the Bubble Size in a Current Default Bubble Chart

In a bubble chart, you use a measure or fact to determine the size of the bubbles.

You can use the Bubble Measure in the Measures drop zone of the chart to specify bubble size. You can then specify the size range of the bubbles in the chart. You can also specify what value the smallest bubbles represent. For example, your bubble measure is revenue, and you set the minimum and maximum bubble size to 5 pt and 20 pt, respectively. You set the smallest bubble to represent the value zero. All the bubbles on your chart will be between 5 and 20 pts and any bubbles between 0 and 5 pts on this scale will appear at 5 pt.

You can combine bubble size with colors by value to create a chart that shows multiple dimensions.

Procedure

1. Select the chart object.
2. In Properties pane, under General, double-click the Bubble Size property.
3. Under Smallest bubble, select what value the smallest bubbles represent:
   - To show data items with a value of zero at the minimum bubble size, click Zero.

   **Tip:** This setting is consistent with Microsoft Excel 2003.

   - To show data items with a value of zero at the minimum bubble size and show negative bubbles as hollow, select the Zero. Negatives shown as hollow check box.

   **Tip:** This setting is consistent with Microsoft Excel 2007.

   - To set the minimum bubble size to the minimum data value, click Minimum data value, and type a size for the minimum and maximum bubble size.

   **Note:** The minimum data value can be positive or negative.

Set the Position of the First Slice in a Current Default Pie Chart

You can specify the angle in a pie chart where the first pie slice starts. You can also change the direction of the slices so they appear clockwise or counter-clockwise.
The default starting position of the first slice is 90, which displays the first slice beginning at the 12 o'clock position. By default, the slices appear clockwise around the pie, so a starting position of 90 displays the first slice at the 12 o'clock position, a starting position of 180 displays the first slice at the 9 o'clock position, and so on.

**Procedure**

1. Select the pie chart object.
2. In the Properties pane, under General, beside the First Slice Angle property, enter the angle at which you want the first slice to appear.
3. To change the directions in which slices appear, change the Slice Direction property.

**Related concepts:**

"Current Default Charts and Legacy Charts" on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

"Pie Charts" on page 86
Pie charts are useful for highlighting proportions.

**Create a Donut Chart from a Current Default Pie Chart**

You can add a hole to the middle of your pie chart to create a donut chart. You can then display something in the hole, such as a company logo, a calculation, or the legend.

**Procedure**

1. Select the pie chart object.
2. In the Properties pane, under General, set the Hole size (%) property to the percentage of the pie that you want the hole to take up.

   **Tip:** You can select a value from the Hole size (%) list or type a value that is not listed.

**Related concepts:**

"Current Default Charts and Legacy Charts" on page 79
IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

"Pie Charts" on page 86
Pie charts are useful for highlighting proportions.

**Pull Out Pie Slices in a Current Default Pie Chart**

You can highlight pie slices by pulling them out from the rest of the pie. For example, the following chart shows revenue by product line with slices of less than 1,000,000,000 in revenue pulled out by 25%.
Procedure

1. Select the pie chart object.
2. In the Properties pane, under General, double-click the Exploded Slices property.
3. In the Exploded Slice dialog box, click the new button.
4. In the Exploded amount box, type the percentage by which the slice should appear away from the rest of the pie.
   - 0% displays the pie slice in the pie; 100% displays the pie slice as far out from the rest of the pie as possible.
5. Define which slice to pull out:
   - To select a slice to pull out using its slice index, under Exploded slice, click Slice number and type the slice number.
     - The slice number refers to the order of the slice in the legend. In the example above, Camping Equipment has a slice number of 1 and Personal Accessories has a slice number of 5.
   - To select a slice to pull out using a calculation, under Exploded slice, click Expression and click the ellipsis (...) beside the Expression box to define an expression.
     - In the example above, the expression that defines which slices to pull out is as follows: [Query1].[Revenue] < 100000000.
       - The slice definitions appear in the Exploded Slices list.
6. To pull out other slices, repeat steps 3 to 5.

Figure 15. Exploding slices in a pie chart
Define the Cumulation Line in a Current Default Pareto Chart

The cumulation line on a Pareto chart displays the percentage of the accumulated total of all the columns or bars. For example, if your chart displays revenue by product line by year, the cumulation line at the second year's column would be the total revenue of the first and second years.

You can customize the appearance of the cumulation line and its data points.

Procedure
1. Select the Pareto chart.
2. To hide or show the cumulation line, in the Properties pane, under Chart Annotations, set the Cumulation Line property.
3. To customize the cumulation line, click the cumulation line icon, and in the Properties pane, under General, set the following properties:
   - Line Styles defines the cumulation line color, style, and weight.
   - Cumulation Label defines the label in the legend for the cumulation line. You can use the default label from the data source or type a custom label. The color and marker shape of the cumulation line still appears in the legend when this property is set to None.
   - Data Points defines whether to show or hide data points along the cumulation line, whether to show or hide data point borders, the color of data point borders, and the data point size and shape.
   - Data Labels specifies whether to show or hide the labels for the data points along the cumulation line.

Define the Cumulation Line in a Legacy Pareto Chart

The cumulation line on a Pareto chart displays the percentage of the accumulated total of all the columns or bars. For example, if your chart displays revenue by product line by year, the cumulation line at the second year's column would be the total revenue of the first and second years.

You can customize the appearance of the cumulation line and its data points.

Procedure
1. Select the Pareto chart.
2. To hide or show the cumulation line, in the Properties pane, under Chart Annotations, set the Cumulative Line property to Show.
3. To customize the cumulation line, click the cumulation line icon and in the Properties pane, under General, set the following properties:
   - **Cumulation Axis** displays or hides the cumulation line's axis on the right of the Pareto chart.
   - **Cumulation Label** displays or hides the cumulation line label in the legend. The color and marker shape of the cumulation line still appears in the legend when this property is set to No.
   - **Line Styles** defines the cumulation line color, style, and weight.
   - **Marker Size (pt)** defines the size of the markers along the cumulation line in points.
   - **Marker Shape** defines the shape of the markers along the cumulation line.
   - **Values** specifies whether to show or hide the values for the markers along the cumulation line.
   - **Value Location** defines the location of the marker values.

**Related concepts:**
- "Current Default Charts and Legacy Charts” on page 79
- "Pareto Charts” on page 95

IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

Pareto charts help you to improve processes by identifying the primary causes of an event. They rank categories from the most frequent to the least frequent. These charts are frequently used for quality control data, so that you can identify and reduce the primary cause of problems.

**Insert a Microchart into a Crosstab**

You can use microcharts to improve the visualization of data in crosstabs.

**Procedure**

1. Select a crosstab row or column.
2. From the right-click menu, click **Insert Chart for Row Data** or **Insert Chart for Column Data**.
3. From the **Insert Chart** dialog box, select a chart and click **OK**.
4. Specify the data to plot in the microchart.
   - The chart automatically plots the data in the specified rows or columns. You can change this if necessary.

**Customize a Current Default Bullet Chart**

After you create a bullet chart, you can customize the shape, color, and size of the bullet and target indicators.

By default, the bullet chart includes three gray colored regions in the background. You can edit the colored regions (Properties pane, Colored Regions).

**Procedure**

1. Select the bullet chart object.
2. To change the shape, color, and size of the bullet or target, do the following:
   a. In the Properties pane, under General, double-click the Bullet Indicators property.
   b. Under Bullet, specify how to show the bullet.
The **Bar width** setting specifies the width of the bullet bar as a percentage of the space available. For example, if you specify 50 percent, the bar uses half of the space available. If you specify 100 percent, the bar uses all the available space.

c. Under **Target**, specify how to show the target.

3. To change the chart orientation, in the **Properties** pane, under **General**, set the **Chart Orientation** property.

**Related concepts:**

- “Current Default Charts and Legacy Charts” on page 79
- IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.
- “Bullet Charts” on page 93

Bullet charts are a variation of bar charts. They compare a featured measure (the bullet) to a targeted measure (the target). They also relate the compared measures against colored regions in the background that provide additional qualitative measurements, such as good, satisfactory, and poor.

**Related tasks:**

- “Add Colored Regions in a Current Default Chart” on page 117

You can define colored regions in the body of a chart. For example, you can divide the background of a scatter chart into quadrants and color each quadrant.

### Changing the number of hotspots in a chart

To improve performance, you can limit the number of hotspots that are generated for Report Studio charts.

**About this task**

A hotspot in a chart appears when you pause a pointer over it. For example, a hotspot on a drill-down symbol or a tooltip gives details about the column, line, or pie slice. The browser response time increases with the number of hotspots. When charts with many members are generated, the hotspots can become an additional burden for the system resources, which can freeze the browser.

When you limit the number of hotspots, priority is given to items such as axis labels and legend labels before individual graphical elements such as bars, pie slices, and so on. Depending on the number of items in a chart and the setting for maximum number of hotspots, some axis items may have hotspots while other axis items and all graphical elements do not, or all axis items and some graphical elements may have hotspots while other graphical elements do not.

The maximum hotspot setting in Report Studio overrides the default set by the administrator. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

**Procedure**

1. Select the **chart object**.
2. In the **Properties** pane, under **Miscellaneous**, specify a number for the **Maximum Hotspots** property.

### Create a Drill-up and Drill-down Chart

If you use a dimensionally-modeled data source, you can create a chart that allows you to drill down to lower-level data or drill up to higher-level data.
Drilling up and down allows you to view more general or more detailed information on your data within a predefined dimensional hierarchy.

This is an example of a dimensional hierarchy:

Years - Year - Quarter - Month

**Before you begin**

Before you begin, ensure that you are using a dimensionally-modeled data source.

**Procedure**

1. Open a chart that uses a dimensionally-modeled data source.
2. From the Data menu, click **Drill Behavior**.
3. On the Basic tab, under **Report output drill capabilities**, select the **Allow drill-up and drill-down** check box.
   
   By default, the system determines which items can be drilled on based on the dimensional structure.

   On the Basic tab, you can make drilling unavailable for any data item by selecting the item in either the **Disable drill-up for** box or the **Disable drill-down for** box.

   On the Advanced tab, you can change the drill-up or drill-down behavior for any parameter by selecting the parameter and then choosing one of the desired behaviors.

**Results**

The chart generates links for any item that can be drilled down on.

You can drill down or drill up by right-clicking and choosing the action from the context menu. The menu items are unavailable if an item cannot be drilled up or down on.

**Related tasks:**

“Create a Drill-up and Drill-down Report” on page 308

You can link groups of data items from different queries so that when you drill up or drill down in one query, the data item also drills up or drills down in the linked queries.

**Example - Creating Drill-through Access in a Legacy Chart**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a chart that shows the revenue for each product line and allows the reader to drill through from the revenue chart to view the product details for any item selected. You create a drill-through report to link two reports containing related information. You can then access related or more detailed information in one report by selecting a value in the chart. You create two reports: a target list report that contains the details for the item and a source report that contains the chart that shows the product line revenue.
Create the target report

This example has three parts: creating the target report, creating the source report, and defining the drill behavior.

Procedure

1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click List and click OK.
4. From the Source tab, expand Sales and Marketing (query), Sales (query), and Product.
5. Double-click the following data items to add them to the list:
   - Product line
   - Product type
   - Product name
   - Introduction date
   - Product image
   
   Now you must create a filter to use as a drill-through parameter. A drill-through parameter begins and ends with a question mark (?) symbol.

6. Click the filters button.
7. Click the add button and type the following in the Expression Definition window:
   
   [Sales (query)].[Product].[Product line]=?p_PL?

8. Save the report as Product Line Details.

Create the Source Report

This example has three parts: creating the target report, creating the source report, and defining the drill behavior.

Procedure

1. Create a new report.
2. In the New dialog box, click Chart and click OK.
3. In the Chart group pane, click Column.
4. In the Chart type pane, click Column.
5. Click OK.
6. From the Source tab, expand Sales and Marketing (query) and then Sales (query).
7. Expand Sales fact and drag Revenue to the Measure (Y-axis) drop zone.
8. Expand **Order method** and drag **Order method** to the **Series** drop zone.
9. Expand **Product** and drag **Product Line** to the **Categories (X-axis)** drop zone.

**Define the Drill Behavior**

This example has three parts: creating the target report, creating the source report, and defining the drill behavior.

**Procedure**

1. From the **Data** menu, click **Drill Behavior**.
2. On the **Basic** tab, under **Report output drill capabilities**, select the **Allow this report to be a package-based drill-through source** check box and click **OK**.
3. Right-click the chart object and click **Drill-Through Definitions**.
4. Click **New Drill Through**.
5. Under **Report**, click the ellipsis (...) button.
6. Select the **Product Line Details** report you created and click **Open**.
8. Under **Format**, click **HTML**.
9. Click the edit button...
   Any existing drill-through parameters appear. You see the parameter you created for **Product Line Details**.
10. For item **p_PL**, under **Method**, click **Pass data item value**, and under **Value**, click **Product line**.
11. Save the chart as **Product Revenue**.
12. Run the report.

**Results**

The chart shows the product lines as clickable links. When you click a product line, the second report runs for that product line.

---

**Example - Creating and Customizing a Metrics Range Legacy Chart**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are asked to create a chart that shows revenue compared to planned revenue by quarter by product line. You also want to highlight revenues that were below target.

This example has three parts: creating a metrics chart, customizing it, and customizing the legend labels.

**Related concepts:**

- "Current Default Charts and Legacy Charts" on page 79
- IBM Cognos Report Studio has a new default chart technology as of version 10.1.0.

**Create a Metrics Range Chart**

This example has three parts: creating a metrics range chart, customizing it, and customizing the chart legend labels.

**Procedure**

1. Open IBM Cognos Report Studio with the **GO Data Warehouse (query)** package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click Chart and click OK.
4. In the Chart group pane, click Metrics Range.
5. In the Chart type pane, click Column Chart with Range Indicators and click OK.

6. From the Source tab, expand Sales and Marketing (query), Sales (query), and Sales fact.
7. Drag Revenue to the Actual (Y-axis) drop zone.
8. Drag Planned Revenue to the Target (marker) drop zone.
9. Expand Time dimension, and drag Quarter to the Categories (X-axis) drop zone.

10. From the Toolbox tab, drag a query calculation to the Tolerance (marker) drop zone.
11. In the Create Calculation dialog box, type Tolerance.
12. In the Expression Definition box, type [Planned revenue] * 0.1 and click OK.
13. Run the report and then pause the pointer over each bar to see the revenue for that quarter.

Customize the Chart
This example has three parts: creating the metrics range chart, customizing it, and customizing the legend labels.

Procedure
1. Select the chart body.
2. In the Properties pane, under Target Markers, ensure that the Performance Pattern property is set to On Target.
3. Double-click the Marker Color property, click Lime and click OK.
4. Set the Upper Range Skew (%) property to 0%.
   This option removes the upper tolerance line. In this example, you do not need to see the upper tolerance line because the revenue is always below the planned revenue target.
5. Ensure that the Target Range (%) property is set to 50%.
   This option sets the size of the range around the planned revenue target. The percentage you choose is the percentage of the tolerance measure. In this example, a 50% target range would display 50% of the tolerance calculation you created, which is 50% of 10% of planned revenue.
6. Double-click the Target Color property, click Red and click OK.
7. Double-click the Target Marker Border Color property, click Green and click OK.
8. Run the report.

Customize the Chart Legend Labels
This example has three parts: creating the metrics range chart, customizing it, and customizing the legend labels.

Procedure
1. Under the chart legend select Marker.
2. In the Properties pane, under Text Source, double-click the Text property.
3. Replace the text in the Text dialog box with Planned revenue and click OK.
4. Under the chart legend select Tolerance.
5. In the Properties pane, under Text Source, double-click the Text property.
6. Replace the text in the Text dialog box with Tolerance: 10% of planned revenue and click OK.
7. Under the chart legend, select Range.
8. In the Properties pane, under Text Source, double-click the Text property.
9. Replace the text in the Text dialog box with Planned revenue range = 50% and click OK.
10. Run the report.

Results

The report now shows customized legend labels for the markers, tolerances, and ranges as shown below:

![Image of chart with customized legend labels](image)

**Figure 16.** Metrics range chart with customized legend labels for the markers, tolerances, and ranges

---

**Adding a Metric Studio Diagram to a Report**

You can add IBM Cognos Metric Studio impact or custom diagrams in your reports. Impact diagrams show cause-and-effect relationships between metrics. Custom diagrams allow you to monitor your metrics using a predefined visual representation, such as a process diagram or strategy map.

The diagrams are added as images within your report.

You can add a diagram in the following ways:

- as a static image with the Metric Studio Diagram object
- as a dynamic image from a metrics package

You can also add range indicator charts.
You must first create custom diagrams in Metric Studio before you can add them in an IBM Cognos Report Studio report. Impact diagrams are automatically created in the metrics package. For information about creating Metric Studio diagrams, see the Metric Studio User Guide.

Add a Static Metric Studio Diagram to a Report
You insert the image using the diagram identifier from Metric Studio. The identifier is converted to an image URL and the image appears in your report output.

Because the diagram is a static image, any changes to the metrics will not appear in the diagram image within your report.

Procedure
1. From the Toolbox tab, drag a Metric Studio Diagram object to the report.
2. Select the Metric Studio diagram object.
3. In the Properties pane, paste the diagram identifier in the Diagram Identifier box.
   Tip: Copy the identifier from Metric Studio (Diagrams tab, View the Diagram Identifier button in the Actions column).
4. In the Description box, type a description for the diagram.
5. If you want to make the diagram image accessible, in the Alternate Text box, type a description for the image.

Related tasks:
"Add Alternate Text to Images and Charts" on page 470
You can add alternate text for images, maps, and charts to make your reports accessible. When a screen reader encounters one of these objects, it reads the alternate text that you added to the object.

Add a Data-driven Metric Studio Diagram to a Report
If you use a metrics package (which is a relational data source), you can insert a Metric Studio diagram directly from the Toolbox tab. Because you add the diagram from the metrics store, any changes to the metrics will appear in the diagram image when you re-run your report.

If you add diagrams from the Metric History table of the metrics package, one diagram is added for each metric per time history. As a result, your report could contain a large number of diagrams. To improve the performance of your report, consider adding a filter to limit the time period.

Procedure
1. Open IBM Cognos Report Studio with a metrics package.
2. From the Toolbox tab, insert a diagram query item to the report.
3. If you want to make the diagram image accessible, select the image object for the diagram, and in the Alternate Text box, type a description for the image.

Related tasks:
"Add Alternate Text to Images and Charts" on page 470
You can add alternate text for images, maps, and charts to make your reports accessible. When a screen reader encounters one of these objects, it reads the alternate text that you added to the object.
Chapter 7. Maps

IBM Cognos Report Studio provides a set of maps that you can use to represent tabular data in a spatial context. For example, on a map of the world, countries and regions can be colored to represent the level of revenue.

Maps are not supported for reports run in Microsoft Excel format.

The Report Studio Map Object

Maps are most often used to show geographical areas, but they can be used to show other spatial information, such as a floor plan in a building, seats in an airplane, or parts of the human body.

Maps are similar to crosstabs in the way their data is organized. The display is different, but maps show the intersection of data the same ways as crosstabs; for example, you can see the revenue for golf equipment in Canada.

A map in IBM Cognos Business Intelligence consists of a collection of layers. Each layer contains different information and acts like a transparency that is placed on top of the map. Each layer adds more information to the map. For example a map of the world may contain information related to countries or regions on one layer and information related to cities on another level.

IBM Cognos Report Studio maps provide the following three types of layers:

• Region layer
  Specifies the regions on a map to be differentiated according to values in the data source. For example, to show the revenue level for each country and region on a map of the world, choose Country and Region as the region layer and then specify that the color of each country and region is based on the revenue value for that country and region. Areas can be set up for drilling through to other reports.

• Point layer
  Specifies the points to be placed on a map. The color and size of the points is based on the data that you select. For example, you choose to show cities as points on a map and set the color of each point by revenue and the size of each point by profit. Points can be set up for drilling through to other reports.
• Display layer
  You can show or hide items such as grid lines or capital cities. This layer is
determined in the map file and not in the data source.

Note: If you intend to create CSV or XML output from your map, use only a point
layer or a region layer. CSV and XML do not support the simultaneous use of both
layers in the same map. Only one layer will be rendered in the output.

Parts of Map Reports

The following shows the parts of a map as they appear in the IBM Cognos Report
Studio interface.

Example - Create a Map Report

You are a report author at The Sample Outdoors Company, which sells sporting
equipment. You are asked to show how revenue for the company is distributed
throughout the world. This information can be shown in tabular format using a list
report, but a map will create a more meaningful presentation. You decide to create
a report that contains a map of the world showing the distribution of revenue by
country and region.

Procedure
1. Open IBM Cognos Report Studio with the GO Data Warehouse (query)
   package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click Map and click OK.
4. In the **Choose Map** dialog box, in the **Maps** pane, expand the **World** folder and click **World**.

5. In the **Region layers** box, click **Countries and Regions + Territories**.

6. In the **Point layers** box, click **None**.

7. In the **Display layers** box, click **Oceans**.

   **Tip:** You can select or deselect multiple display layers by Ctrl+clicking.

   **Tip:** You can return to the **Choose Map** dialog box at any time by double-clicking the map background.

8. Drag the following data items to the map:
   - **Revenue** (in **Sales fact**) to the **Color** drop zone.
   - **Retailer country and region** (in **Retailer site**) to the **Location** drop zone.

9. Run the report.

**Results**

![Image of World map](image)

*Figure 17. World map*

**Set Map Properties**

When you select an element in a map, you can view its properties in the **Properties** pane. Some properties are dependent on the existence of other properties.

If you are familiar with using conditional variables, you can customize the map to change appearance or provide information in response to expressions or conditions.

When you change a map property, you usually do not see the change until you run the report. Changes to the properties of labels and titles are reflected immediately.

These are some of the properties you can change in maps. These properties are available when you select the map object, unless specified otherwise in the **Action to perform in the Properties pane** column.
<table>
<thead>
<tr>
<th>Goal</th>
<th>Action to perform in the Properties pane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide or show the title, subtitle, footer, or axis title</td>
<td>Under <strong>Chart Titles</strong>, set the <strong>Title, Subtitle, Footer, or Axis Title</strong> property.</td>
</tr>
<tr>
<td>Hide or show the legend</td>
<td>Under <strong>Chart Annotations</strong>, set the <strong>Legend</strong> property.</td>
</tr>
<tr>
<td>Hide or show region and point layer values and labels on the map</td>
<td>Select the region or point layer. Under <strong>Chart Labels</strong>, double-click the <strong>Show Values</strong> property and select the options that you want. To hide overlapping labels, select the <strong>Hide overlapping labels</strong> check box.</td>
</tr>
<tr>
<td>Hide or show display layer labels on the map</td>
<td>Select the display layer. Under <strong>Chart Labels</strong>, double-click the <strong>Show Labels</strong> property and select the option that you want. To hide overlapping labels, select the <strong>Hide overlapping labels</strong> check box.</td>
</tr>
<tr>
<td>Hide or show the border around the legend</td>
<td>Select the legend icon. Under <strong>Box</strong>, set the <strong>Borders</strong> property.</td>
</tr>
<tr>
<td>Change the border around the map object</td>
<td>Under <strong>Box</strong>, set the <strong>Border</strong> property.</td>
</tr>
<tr>
<td>Hide or show the tooltips</td>
<td>Under <strong>Chart Labels</strong>, set the <strong>Tooltips</strong> property.</td>
</tr>
<tr>
<td><strong>Note:</strong> Some versions of Acrobat Reader do not support tooltips.</td>
<td></td>
</tr>
<tr>
<td>Change the amount of white space around the map</td>
<td>Under <strong>Box</strong>, set the <strong>Padding</strong> or <strong>Margin</strong> property.</td>
</tr>
<tr>
<td>Change the default colors for all map elements</td>
<td>Under <strong>Color &amp; Background</strong>, set the <strong>Background Color, Foreground Color, or Fill Effects</strong> property.</td>
</tr>
<tr>
<td>Change the font and the alignment of text</td>
<td>Under <strong>Font &amp; Text</strong>, set the <strong>Font</strong> or <strong>Relative Alignment</strong> property. <strong>Tip:</strong> The 2011 Quarterly Sales Forecast sample report in the GO Sales (analysis) package includes text alignment. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.</td>
</tr>
<tr>
<td>Resize the map</td>
<td>Under <strong>Positioning</strong>, set the <strong>Size &amp; Overflow</strong> property.</td>
</tr>
<tr>
<td>Change the font for the legend</td>
<td>Select the legend icon. Under <strong>Font &amp; Text</strong>, set the <strong>Font</strong> property.</td>
</tr>
<tr>
<td>Change the format of values in the legend</td>
<td>Select the value in the region or point layer. Under <strong>Data</strong>, set the <strong>Data Format</strong> property.</td>
</tr>
<tr>
<td>Ignore data with no features</td>
<td>Under <strong>Data</strong>, set the <strong>Ignore Data with No Features</strong> property.</td>
</tr>
<tr>
<td>Specify the size of points</td>
<td>In the point layer, click the measure in the <strong>Size</strong> drop zone and set the <strong>Minimum Size</strong> and <strong>Maximum Size</strong> properties.</td>
</tr>
<tr>
<td>Add titles, subtitles, footers, or axis titles</td>
<td>Set the <strong>Title, Subtitle, Footer, or Axis Title</strong> property.</td>
</tr>
</tbody>
</table>
Procedure
1. Select the map object or map element to change:
   - To change general properties, such as size and color, click the map object.
   - To change specific map elements, such as a layer or title, click the element itself.

   Tip: To cancel a selection, press the Esc key.
2. In the Properties pane, set the property value.
   An ellipsis (...) button indicates that a dialog box provides further options.

   Note: You may have to scroll to see all the properties.

Example - Define Data Values for the Region Layer
The map that you created in the previous topic is not yet linked to a data source. You will now specify the data values from your data source that will determine the color of each region.

Procedure
1. From the Source tab, expand Sales and Marketing (query), Sales (query), and Sales fact.
2. Drag Revenue to the Color drop zone.
3. Expand Employee by organization.
4. Drag Country or Region to the Location drop zone.
5. Run the report.

Results

Figure 18. Map showing revenue by country or region
### Match Data Values to Names in the Map File

If the **Ignore Data with No Features** property is set to **No**, then each object that is called from the data source must have a matching label in the specified layer of the map file. For example, if your data source has a country and region named United States and the layer in the map file labels the same country and region USA, then there is a mismatch that must be corrected. IBM Cognos Report Studio only makes you aware of a mismatch if each object in your data source does not have a corresponding label in the map file. If there are extra labels in the map file that do not have a match in the data source, the report will run without an error message.

A mismatch between your data and the map file must be corrected by the report author. It cannot be corrected by a consumer of the map report at run time. There are two ways to correct a mismatch between your data and the labels in the map files. You can use IBM Cognos Map Manager to edit the labels in the layers of the map file, or you can use the dictionary property to create an alias for each mismatched object. When you use the dictionary property, it resolves the mismatch only for a single report, and is not shared with other reports. If you intend to continue using a map with the same data source, it is best to edit the map in Map Manager so that the labels match the objects in your data source.

For information about using Map Manager, see the Map Manager *Installation and User Guide*.

**Procedure**

1. Select the map object.
   - The title bar of the Properties pane now shows the word Map.
2. In the General section of the Properties pane, double-click the Dictionary property.
3. Click the new button.
4. In the Dictionary Entry dialog box, click Search.
5. In the Search string box, type a word or part of a word for which to search.
   - For example, if you are searching for United States, type in part or all of the name.
6. In the Search map layer box, click the layer to search and click Search.
7. In the Matching features box, click the label to which to match your data source and click OK.
8. In the Alias box, type the name as it appears in the data source and click OK.
   - For example, if the country and region in your data source is named USA, type USA as the alias.

**Note:** To find out the name for the objects in your data source, run a list report. For example, you can run a list report to show the names of all the countries and regions in your data source. For more information see Chapter 4, “Lists,” on page 63.

### Define Data Values for the Point Layer

The Point layer in a map is used to visually represent data for point locations, such as cities or sales outlets. Both the color and size of points can be based on data from your data source.
Procedure

1. Open the Choose Map dialog box:
   - When you create a new map report, this dialog box appears automatically.
   - If you are already in a map report, double-click the map background.

2. In the Point Layers box, select the layer containing the points to show on the map.
   For example, on a map of the world, you may want the points to represent cities.

3. From the Source tab, drag a data item to the Color drop zone in the Point Layer.
   For example, to have the color of the point based on revenue, drag Revenue from the Source tab to the Color drop zone.

4. From the Source tab, drag a data item to the Size drop zone in the Point Layer.

5. From the Source tab, drag a data item to the Location drop zone in the Point Layer.
   The object must be supported in the map file as a point location. For example, in the World sample map, city is supported as a point location but country or region is not.

6. If you need to refine the location, drag an object to the Refine Location drop zone.
   Use this drop zone when there is more than one location with the same name. For example, if you try to run a report with cities in the point layer, and there is more than one city in your data source with the same name, the report does not run. An error message indicates that there are cities in your data source with duplicate names. You can differentiate the cities by using the data object Region to refine the location.

Add Colors to the Region or Point Layer

You can add colors for regions or points and specify values to determine when those colors are shown.

Procedure

1. In the report, click the Region Layer or Point Layer.

2. In the Color & Background section of the Properties pane, double-click the Palette property.

3. Click the new button and click Color.
   A new color is added to the list of colors.

4. To view the palette colors as a continuous spectrum in which colors blend into one another, click Continuous Colors.

5. With the new color selected, click Color in the right pane of the dialog box and select a color.

6. Change the percentage boundaries for the colors.

   Tip: To specify absolute values rather than percentages, clear the Percentage check box.
Add a Legend Title to a Map

There are legend titles for the entire legend, for the color of the regions, for the color of the points, and for the size of the points.

Procedure

1. If the legend title is not showing, click the legend icon.
2. In the General section of the Properties pane, set the Legend Title to Show.
3. In the report, double-click the legend title and type the title.
4. By default, the legend titles are taken from the object you have selected from the data source. To change the legend title for the region color, point color, or point size, do one of the following:
   - From the Toolbox tab, drag a text or calculation object to the Color Legend Title drop zone in the Region Layer or Point Layer, or the Size Legend Title drop zone in the Point Layer.
   - Double-click the Color Legend Title or Size Legend Title drop zone, to change from the default legend title, then double-click the drop zone again. In the Text dialog box, type text for the legend title.

Add a Note to a Map

You can add one or more notes, determine their position in the map report, and specify borders around notes.

Procedure

1. Select the map object.
2. In the Chart Annotations section of the Properties pane, double-click the Notes property.
3. Click the new button, and click OK twice.
   A note icon with the words New Note appears in the report.
4. Click New Note next to the note icon.
5. In the Text Source section of the Properties pane, double-click the Text property.
6. Type the text to appear in the note and click OK.
7. In the report, click the note icon.
8. In the Positioning section of the Properties pane, type values to specify the bottom position, left position, height, and width for the note.
   The location of the note is defined by the number of pixels.
9. Set the Note Border property to specify a border around the note.
10. Run the report to view the note.
    If necessary, change the position again.

Drill Through to Another Report From a Map

You can link regions or points on a map to another report. For example, on a map of the world, you can specify that when you click on China, a map of China opens.
Procedure

1. Open the target report.
2. From the Data menu, click Filters.
3. On the Detail Filters tab, click the add button.
4. In the Available Components box, click the Source or Data Items tab to select the data item to use.
   For example, to open the target report when Canada is clicked in the source report, expand Countries and Regions and double-click Country and Region.
5. In the Expression Definition box, type an operator after the data item or select an operator from the Functions tab and then enter a value.
   For example, to open the report when Canada is clicked in the source report, the expression would be as follows:
   [Country and Region]='Canada', where [Country and Region] is the name of the data item in the package.
6. Save the target report.
7. Open the source report.
8. Select the region layer or the point layer.
9. In the Data section of the Properties pane, double-click the Map Drills property.
10. In the Map Drills dialog box, click the new button.
11. In the Search string box, type the name of the feature to use for the drill-through link.
    For example, to open a report when you click on Canada in the map, search on all or part of the word Canada.
12. Click Starts with to search for features that start with the search string or click Contains to search for feature names that include the search string.
13. To include a parent layer's name in the results of your search, select a layer in the Include parent layer list.
    For example, if you searched for Oslo and you included the parent layer of Countries and Regions + Territories, the search result would be Oslo (Norway).
14. Click Search.
15. Select a feature from Matching features box and click OK
16. In the Drill-Through Definitions dialog box, click the new button.
17. Click the ellipsis (...) button beside the Report box, and select the target report.
18. Run the report.

Results

When you click the selected feature in the source report, the target report will open.

Edit a Map

With IBM Cognos Map Manager you can edit the labels in maps to be consistent with the object names in your database.
Administrators and modelers use a Microsoft Windows operating system utility named Map Manager to import maps and update labels for maps in IBM Cognos Report Studio. For map features such as country or region and city names, administrators and modelers can define alternative names to provide multilingual versions of text that appears on the map.

For instructions, see the Map Manager Installation and User Guide.

**Additional Maps**

IBM Cognos supplies a set of standard maps that can be used directly with IBM Cognos Report Studio as well as IBM Cognos Map Manager.

If you already have proprietary geographic data, you can also create your own custom .GST files and then import these into Map Manager. To create custom .GST files, you must use an application that produces MapInfo files, such as MapInfo Professional.

**Location Intelligence**

Sophisticated mapping functionality, known as location intelligence, can be used for a broad range of business applications that extend the mapping capability of IBM Cognos Business Intelligence. MapInfo provides solutions that can be directly integrated with IBM Cognos BI. These include the ability to dynamically create geographic filters and custom areas for aggregating data for ad-hoc analysis. Examples of business applications of location intelligence are listed in the following table.

<table>
<thead>
<tr>
<th>Business application</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target marketing</td>
<td>Learn who your best clients are and find more like them.</td>
</tr>
<tr>
<td>Network optimization and site location analysis</td>
<td>Put stores near your customers and look for gaps in geographical coverage.</td>
</tr>
<tr>
<td>Routing and work force optimization</td>
<td>Reduce the number of trucks you need and make your drivers more efficient.</td>
</tr>
<tr>
<td>e-government</td>
<td>Provide citizens with self-service opportunities.</td>
</tr>
<tr>
<td>Sales territory creation</td>
<td>Create balanced sales territories.</td>
</tr>
<tr>
<td>Economic development</td>
<td>Plan the development of your community.</td>
</tr>
<tr>
<td>Communications network planning</td>
<td>Avoid costly mistakes by putting cell towers in the right locations. Identify the locations of clients in your service area.</td>
</tr>
</tbody>
</table>

You can contact MapInfo for both data and location intelligence solutions through their Web site: [http://www.mapinfo.com](http://www.mapinfo.com)
Chapter 8. Active Reports

You can use IBM Cognos Report Studio to create active reports. IBM Cognos Active Report is a report output type that provides a highly interactive and easy-to-use managed report. Active reports are built for business users, allowing them to explore their data and derive additional insight.

Active reports make business intelligence easier for the casual user. Report authors build reports targeted at their users' needs, keeping the user experience simple and engaging. Active reports can be consumed by users who are offline, making them an ideal solution for remote users such as the sales force.

Active reports are an extension of the traditional IBM Cognos report. You can leverage existing reports and convert them to active reports by adding interactive behavior, providing end users with an easy-to-consume interface.

Like existing IBM Cognos reports, you can execute active reports from IBM Cognos Connection as well as schedule and burst them to users.

You build active reports with the same objects that you use to build other report types. However, there are objects that are specific to active reports. These objects fall into two categories:

- active report controls
- active report variables

For tips about building active reports, see the Business analytics proven practices page (http://www.ibm.com/developerworks/analytics/practices.html).

Note: Cognos Active Report does not support JavaScript.

Active Report Controls

You use active report controls to create the layout of an active report as well as filter, sort, and navigate through data in the report.

Layout

The following controls are used to build the layout of an active report:

- Tab controls, which are used for grouping similar report items.
- Decks of cards, which are used for layering report items.
- Hiding or showing list columns, which allows users to control the data they see by using check boxes.
- Row numbers, which show the number of rows of data in a control.

Filtering and Sorting

To help report authors deliver the content in the most consumable way possible, IBM Cognos Report Studio provides several new filtering controls:

- List and drop-down list controls
- Interactions with charts
For example, clicking a category in a chart filters the data in a list.

- Radial buttons
- Check boxes
- Toggle buttons
- Push button controls

**Navigation**

To help users navigate through data in a report, report authors can add the following controls:

- Iterators
- Sliders

**Data Containers**

Data containers, such as lists, crosstabs, charts, visualizations, and repeater tables, are also considered controls in an active report, as you can add interactive behavior to them.

**Related concepts:**

“Adding Controls to a Report” on page 185

IBM Cognos Active Report offers several active report controls. For many controls, two different versions are available, a static version and a data-driven version.

---

**Active Report Variables**

Active report variables work in conjunction with active report controls to add interactivity to a report.

Actions performed on a control, such as selecting an item in a control or selecting a control itself, can set the value of a variable. In turn, controls can respond to changes in the value of a variable, such as filtering data in a control.

**Related tasks:**

“Create and Manage Active Report Variables” on page 184

Create active report variables to allow controls in a report to interact with each other. Active report variables pass information from one control to another control.

---

**Convert an Existing Report to an Active Report**

You can leverage existing reports by converting them to active reports.

**Procedure**

1. Open the report that you want to convert.
2. From the **File** menu, click **Convert to Active Report**.

**Results**

IBM Cognos Active Report menu items, objects, and properties become available in IBM Cognos Report Studio. Objects in the report that are not supported in active reports, such as prompt controls in report pages, are removed.
Specify Active Report Properties

You can specify properties for active reports, such as the text that appears in the browser window title bar, the maximum number of rows that a report can contain, and an access code that users must enter to view a report.

Procedure

1. From the File menu, click Active Report Properties.
2. To specify the title that will appear in the browser window title bar when the report is viewed, under Window Title, type the title that you want.
3. If you want to specify the window title in other languages, do the following:
   - Under Window Title, click the ellipsis button (...).
   - Click the add button .
   - Select the languages that you want.
   - In the Language column, click one of the languages that you selected.
   - Click the edit button and type the window title for that language.
4. In the Window Startup Size box, choose among the available options to specify the size of the browser window when an active report is viewed.
5. In the Maximum query rows box, specify the maximum number of rows of data that the report can contain.
   If the number of rows of data returned when the report is run exceeds this value, an error message appears, and users will not be able to view the report.
6. To specify an access code that users will have to enter to view the report, enter the code in the Access code box.
   You can use any Unicode character in the access code, and there is no limit to the number of characters an access code can contain.
   The access code will not be hidden or treated like a password when entered (* will not appear as the code is entered). Users will have three attempts to enter the correct code, after which the active report will shut down. To try again, users have to reopen report.

   Note: The access code is a tool to help prevent the unauthorized consumption of an active report within an organization. It is not an enterprise grade security solution.
7. If you do not want to compress the MHT file that is produced when the active report is run, clear the Compress the active report output check box.
   Compressing the output reduces the size of MHT files so that active reports can support more data and load faster in browsers and mobile devices.
8. To support pinch-to-zoom gestures when the active report is viewed on a mobile device, select the Scalable check box.
9. To reuse the Window Startup Size and Maximum query rows properties when you create other active reports, select the Use as default for new active reports check box.

Extensible visualizations

You can use extensible client-side visualizations in IBM Cognos Active Report to visualize data.
Visualizations must be imported into IBM Cognos Administration. The Visualization Gallery is empty until your administrator imports the visualizations and makes them available to you. Your administrator can set permissions to restrict access to only certain report authors.

The visualizations come from the site [AnalyticsZone](https://www.analyticszone.com). For more information about administration and visualizations, see the IBM Cognos Business Intelligence Administration and Security Guide.

Because visualizations are extensible, they can be customized by an author with the necessary skills. Because they are client-side, they can be run when not connected to the web. You can also use visualizations on mobile devices.

Some of the visualizations that you can add to your active reports are treemaps, heatmaps, packed bubble visualizations, and network diagrams.

Extensible visualizations support all Active Report interactivity. For example, you can use Active Report controls to dynamically filter visualizations to view only the data that you want to analyze. You can also set properties that are available for each visualization type. These properties are determined by the visualization definition. Typically, you can set such properties as width, height, animation effect, and fixed-axis scaling.

For more information about visualizations and their uses, go to [Many Eyes®](http://www-958.ibm.com).

Attention: If you use Microsoft Internet Explorer to view visualizations, you must have Microsoft Silverlight 5 installed on your computer. Firefox or iPad users do not need Silverlight.

**Adding a visualization to an active report**

You can add a visualization to an active report to efficiently represent your data.

**Before you begin**

Your administrator must make the visualizations available in the Visualization Gallery.

If you use Microsoft Internet Explorer to view visualizations, you must have Microsoft Silverlight 5 installed on your computer. Firefox or iPad users do not need Silverlight.

**About this task**

You add data to a visualization by inserting data items in drop zones. Visualizations will not run until all of the required drop zones contain data items. When all the required drop zones contain data items, additional drop zones appear for extra categories or values. You can use extra categories and extra values to filter data or set variables. Extra items are not visible in the visualization.

When working with dimensional data sources, if the visualization does not contain a measure against which to plot data, you must provide one. For example, if you add a single member to the values drop zone, such as '2012', the default measure drop zone appears in which you must insert a measure.
Attention: The supported properties for a visualization are determined by the visualization definition.

Procedure
1. Create a new active report or open an existing active report.
2. From the Toolbox tab, drag a Visualization icon to the work area.
3. From the Visualization Gallery, select a visualization and click OK.
4. From the Source tab, drag items to the appropriate location in the data container.
5. Run the visualization.

Adding a treemap visualization
You can use a treemap visualization in IBM Cognos Active Report to identify patterns and exceptions.

Treemaps show relationships among large numbers of components by using size and color coding in a set of nested rectangles.

A treemap that is colored by category identifies the level 1 category by color. The sizes of the rectangles represent the values. In a treemap that is colored by value, the sizes of the rectangles represent one of the values and the color represents a second set of values.

Attention: The supported properties for a visualization are determined by the visualization definition.
Before you begin

Your administrator must make the visualizations available in the Visualization Gallery.

If you use Microsoft Internet Explorer to view visualizations, you must have Microsoft Silverlight 5 installed on your computer. Firefox or iPad users do not need Silverlight.

About this task

You add data to a visualization by inserting data items in drop zones. Visualizations will not run until all of the required drop zones contain data items. When all the required drop zones contain data items, additional drop zones appear for extra categories or values. You can use extra categories and extra values to filter data or set variables. Extra items are not visible in the visualization.

When working with dimensional data sources, if the visualization does not contain a measure against which to plot data, you must provide one. For example, if you add a single member to the values drop zone, such as ‘2012’, the default measure drop zone appears in which you must insert a measure.

**Attention:** The supported properties for a visualization are determined by the visualization definition.
Procedure

1. From the Toolbox tab, drag a Visualization icon to the work area.
2. From the Visualization Gallery, select a Treemap icon:
   - If you are plotting a single measure, select a treemap colored by category.
   - If you are plotting two measures, select a treemap colored by value.
3. Click OK.
4. From the Source tab, drag items to the appropriate location in the data container:
   a. Under Values, drag items to define the size and color of the rectangles to size and color.
   b. Under Categories, drag items to the different levels. Each level represents the category within which the subsequent levels are nested. For example, the levels might be Year, Product Line, and Product Type.
5. Add any active controls that you want and run the visualization.

Adding a heatmap visualization

Heatmaps in IBM Cognos Active Report use colors to represent the individual values that are contained in a matrix.

Similar to treemaps, you can use heatmaps to identify patterns and exceptions. Unlike treemaps, the sizes of the rectangles are not proportionate to values.

![Heatmap Visualization](image)

Figure 20. A heatmap visualization

Before you begin

Your administrator must make the visualizations available in the Visualization Gallery.
If you use Microsoft Internet Explorer to view visualizations, you must have Microsoft Silverlight 5 installed on your computer. Firefox or iPad users do not need Silverlight.

**About this task**

You add data to a visualization by inserting data items in drop zones.
Visualizations will not run until all of the required drop zones contain data items.
When all the required drop zones contain data items, additional drop zones appear for extra categories or values. You can use extra categories and extra values to filter data or set variables. Extra items are not visible in the visualization.

When working with dimensional data sources, if the visualization does not contain a measure against which to plot data, you must provide one. For example, if you add a single member to the values drop zone, such as ‘2012’, the default measure drop zone appears in which you must insert a measure.

**Attention:** The supported properties for a visualization are determined by the visualization definition.

**Procedure**

1. From the **Toolbox** tab, drag a **Visualization** icon to the work area.
2. From the Visualization Gallery, select the **Heatmap** icon and click **OK**.
3. From the **Source** tab, drag items to the appropriate location in the data container:
   a. Under **Values**, drag a measure or a single value to define the color shading of the rectangles.
   b. Under **Categories**, drag items to the Y Category and X Category.

   **Tip:** Ensure that the items in the Y Category and X Category come from different dimensions.
4. Add any active controls that you want and run the visualization.

**Creating a packed bubble visualization**

A packed bubble visualization in IBM Cognos Active Report is similar to a bubble chart in which the bubbles are tightly packed rather than spread over a grid. You can use a packed bubble visualization to display a large amount of data in a small space.
Before you begin

Your administrator must make the visualizations available in the Visualization Gallery.

If you use Microsoft Internet Explorer to view visualizations, you must have Microsoft Silverlight 5 installed on your computer. Firefox or iPad users do not need Silverlight.

About this task

You add data to a visualization by inserting data items in drop zones. Visualizations will not run until all of the required drop zones contain data items. When all the required drop zones contain data items, additional drop zones appear for extra categories or values. You can use extra categories and extra values to filter data or set variables. Extra items are not visible in the visualization.

When you work with dimensional data sources, if the visualization does not contain a measure against which to plot data, you must provide one. For example, if you add a single member to the values drop zone, such as '2012', the default measure drop zone appears in which you must insert a measure.

Attention: The supported properties for a visualization are determined by the visualization definition.

Procedure

1. From the Toolbox tab, drag a Visualization icon to the work area.
2. From the Visualization Gallery, select the Packed Bubble icon and click OK.
3. From the **Source** tab, drag items to the appropriate location in the data container:
   a. Drag a measure or a single value to the data container, under **Values**.
   b. Under **Categories**, drag an item to plot as the bubbles.
   c. Under **Series**, drag an item to apply as a category. Each item in the category gets a different color from the color palette.

4. Add any active controls that you want and run the visualization.

**Adding a network diagram**

You can create a network diagram in IBM Cognos Active Report. The nodes data container defines the items that you want to link together. The links data container defines relationship between the items that you want to link together.

Before you begin

Network node drop zones only take numeric data items. Not all data sources have the appropriate data to create a network diagram.

Your administrator must make the visualizations available in the Visualization Gallery.

If you use Microsoft Internet Explorer to view visualizations, you must have Microsoft Silverlight 5 installed on your computer. Firefox or iPad users do not need Silverlight.

**About this task**

You add data to a visualization by inserting data items in drop zones. Visualizations will not run until all of the required drop zones contain data items. When all the required drop zones contain data items, additional drop zones appear for extra categories or values. You can use extra categories and extra values to filter data or set variables. Extra items are not visible in the visualization.
When working with dimensional data sources, if the visualization does not contain a measure against which to plot data, you must provide one. For example, if you add a single member to the values drop zone, such as ‘2012’, the default measure drop zone appears in which you must insert a measure.

**Attention:** The supported properties for a visualization are determined by the visualization definition.

**Procedure**

1. From the **Toolbox** tab, drag a **Visualization** icon to the work area.
2. From the Visualization Gallery, select the Network diagram icon and click **OK**.
3. From the **Source** tab, drag items to the appropriate location in the data container:
   a. In the **Nodes1** data container, drag items to **Node (Numeric)** and **Name**. The nodes data container defines the items that you want to link together, such as Employees, defined by an employee code.
   b. In the **Links1** data container, drag items to **From node (Numeric)** and **To node (Numeric)**. The links data container defines relationship between the items that you want to link together, such as Employees and Managers.
4. Add any active controls that you want and run the visualization.

**Related tasks:**

“Example - Network diagram visualization” on page 209

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a visualization in IBM Cognos Active Report that shows the hierarchies of employee by region.

**Resizing a visualization**

You can change the size of a visualization in IBM Cognos Active Report.

**Procedure**

1. Click and drag the lower right corner of the visualization.
2. To maintain the aspect ratio, press the Shift key while you resize the visualization.
   
   You can view the changing height and width of the resized visualization in the **Properties** pane, under **Positioning**.

**Changing the nesting order**

In IBM Cognos Active Report, you can change the nesting order so that extra items are ordered properly in the query. If the items are not ordered properly in the query, the report might not run.

Extra items are data items that are used for filtering or setting variables. These items are not visible in the visualization. By default, extra categories are nested below other categories, so as not to affect sorting. However, if the extra categories are from the same dimension as a category, they must be in the correct hierarchical order: Years, Quarters, and Months. You can change the nesting order.

For example, you have Quarters and Months as categories. You then add Years as an extra category. You add an Active Report control and connect to Years as a filter. You might have to change the nesting order of the data items for the report to run.
**Procedure**

1. Select the data container that contains the data set in which you want to change the nesting order.
2. From the **Properties** pane, under **General**, click **Change Nesting Order**.
3. In the Change Nesting Order window, select the extra category and use the arrow icon to reorder it.

   **Tip:** You can click **Show Data Container** to view the new nesting order when the report is run.

**Populating a visualization with data from the visualization definition**

In IBM Cognos Active Report, you can populate a visualization with data that is specified in the visualization definition instead of adding data items from a package.

**About this task**

You can use data that is specified in the visualization definition to test a visualization before you add your own data. Using data from the visualization definition is also a way to bring external data into a visualization.

**Procedure**

1. Select the visualization.
2. In the Properties pane, double-click the **Use Embedded Data** property.
3. Select the check box of the data container for which you want to use data that is defined in the visualization specification.

**Results**

The data container is removed from the visualization. The data container is not required when the visualization is using data from the visualization definition.

**Specifying the range values that are displayed**

Specify the values that you want to display in a visualization in IBM Cognos Active Report. For example, you can specify the range of values that is displayed when users select values in a control that is connected to a visualization.

**About this task**

You can specify the range of values and categorical data that is displayed in visualizations.

To control the range of values that is displayed, the following choices are available to calculate the minimum and maximum values of the range.

- **All data**
  - With this choice, minimum and maximum values are calculated to span the values of all Active Report controls that are connected to the visualization.

- **Filtered data**
  - With this choice, minimum and maximum values are calculated from values that users select in Active Report controls that are connected to the visualization.

- **Specified values**
With this choice, you specify the minimum and maximum values that you want to display in the visualization.

**Note**: The minimum and maximum values that are displayed in the visualization might differ from the values that you specify. For example, if you specify a range of 0 to 195, IBM Cognos BI might display a range of 0 to 200.

To specify the categorical data that is displayed, the following choices are available.

- **All data**
  - With this choice, all categorical data that spans the values of all Active Report controls that are connected to the visualization are displayed.

- **Filtered data**
  - With this choice, only the filtered data is shown in the legend and on axes.

The choice that you make for displaying categorical data can affect elements in the visualization, such as the legend and axes. For example, when all data is selected, all legend items are always shown in the visualization, and the color of each legend item stays the same in the visualization. When a category is inserted on an axis, space is reserved on the axis for all data. When filtered data is selected, the color of each legend item can change when users select different values in a control that is linked to the visualization. When a category is inserted on an axis, the axis shows only the values that are selected in the control.

**Procedure**

1. To specify the categorical data that is displayed, follow these steps.
   a. In the visualization data container, select the category that you want.
   b. In the Properties pane, click the **Category Range** property and select the choice that you want.

2. To specify the range of values that is displayed, follow these steps.
   a. In the visualization data container, click the data item in the **Values** section that represents the values in the visualization. For example, in a packed bubble visualization, click the data item that is inserted in the **Size** box.
   b. In the Properties pane, double-click the **Value Range** property.
   c. In the **Range type** box, click one of the available choices.
   d. If you clicked **All data** or **Filtered data** and you want the range of values to include zero, select the **Range includes zero** check box.
   e. If you clicked **Specified**, type the minimum and maximum values of the range in the **Minimum** and **Maximum** boxes.

**Updating visualizations**

Update visualizations in IBM Cognos Active Report when a change was made to the visualization definition.

**About this task**

If any part of a visualization is invalidated by changes that were made to the visualization definition, the invalidated part is removed.

If a new property or field was added to the visualization definition, you must delete and recreate the visualization in the report.
If you do not update visualizations when the visualization definition is changed, the old visualization definition is used when you run the report.

**Procedure**
1. Open the report that contains the visualizations that you want to update.
2. Click **Tools > Update Visualizations**.

**Adding Interactivity to a Report**
You create an active report by adding interactive behavior to controls in the report. Adding interactive behavior links controls to each other, allowing an action performed on a control to change the state of another control.

**Create and Manage Active Report Variables**
Create active report variables to allow controls in a report to interact with each other. Active report variables pass information from one control to another control.

**About this task**
You can also create active report variables when you define connections between controls. However, you may want to modify these variables, such as changing the name or defining default values.

In addition, active report variables allow active reports inserted in a workspace in IBM Cognos Workspace to interact with each other. For example, selecting a product line in one active report causes another active report to show the data for the selected product line. In addition, variables allow check boxes and sliders in a workspace in Cognos Workspace to drive active reports in the workspace.

To work in Cognos Workspace, active report variables must meet the following criteria:
- The variables must be public.
  - When you make a public variable, the variable can be consumed in other IBM Cognos products.
- For active reports to listen to each other in a workspace, each report must have a public variable with the same name.
- For a Cognos Workspace slider to drive an active report, a range variable must be defined in the active report.
  - The data item of the slider and the range variable in the active report must have the same name.
  - Two variables are used to define the minimum and maximum values of the range variable. The two variables are used to define the minimum and maximum values of the slider.

**Procedure**
1. Click the **Active Report Variables** tab.
2. Create a variable by following these steps:
   a. Click **Create a New Variable**.
   b. In the **Name** box, type the name of the variable.
   c. Click the **Data Type** drop-down menu and click the type of variable that you want to create.
d. To make the variable public, select the **Public (External)** check box.

e. If you want to specify a default value for the variable, click **Set Default Value(s)** and define or specify the values that you want.

   To define your own default values, in the **Variable Values Source** list, click **Enter Value(s)**, and then click the new icon.

   To specify default values that come from a data item in a control, in the **Variable Values Source** list, click **Select Value(s)**. Specify the control and the data item from which you want to select default values, and then select the values that you want from the **Values** box.

   **Tip:** Define multiple default values when you want more than one item selected in a control that allows multiple selections, such as a check box group.

3. To modify an existing variable, right-click the variable and click **Edit Variable**.

4. To add a default value, click **Set Default Value(s)**.

   To change a default value, click the name of the current default value specified for the variable.

5. Create a range variable by following these steps:
   a. Create the variables that will define the minimum and maximum values of the Cognos Workspace slider.
      The variables must have the number data type.

      **Tip:** Range variables are automatically set as public variables. Do not make the variables that are used to define the minimum and maximum values of the slider public.
   b. Click **Manage Ranges**.
   c. Click the new icon.
   d. In the **Name** box, type the name of the range variable.
   e. Click the **Minimum Variable** list and select the variable that will be used to set the minimum value of the slider.
   f. Click the **Maximum Variable** list and select the variable that will be used to set the maximum value of the slider, and then click **OK** twice.

   **Note:** If you later delete the minimum or maximum variable, the range variable is also deleted.

**Related tasks:**

"Define a Connection Between Controls" on page 192

Define a connection between controls to link them together. When you link controls, an action performed on one control affects the behavior of the other control.

**Adding Controls to a Report**

IBM Cognos Active Report offers several active report controls. For many controls, two different versions are available, a static version and a data-driven version.

Use the static version of a control when you want to manually define the values that the control can contain. Use the data-driven version of a control when you want the values that the control can contain to come from a data item. Controls are available on the **Toolbox** tab.
Tip: To view only toolbox items that are specific to active reports, right-click anywhere in the Toolbox tab and click Active Report Toolbox Items.

When you add a control to an active report, you can convert the control to another type of control that is compatible by right-clicking the control and selecting Convert Control. For example, you can convert a data drop-down list control to a data iterator control.

**Variable Text Items**

Use variable text items to insert active report variables in a report. Variable text items are useful to show the value of a variable when an item in another control is selected. For example, you can use a variable text item as a title for a list that is linked to a drop-down list control. When an item is selected from the drop-down list, the item appears as the title of the list.

**Row numbers**

The active report row number tool adds row numbers that reflect the visible row number in the report. This is different from the report row number tool, which reflects the server calculated row number.

**Repeater tables**

Use static repeater tables and data repeater tables to build a repeating structure of content. Repeater table controls allow you to create button bars or grids of custom content to control the content that appears in other controls. In data repeater table controls, the content that appears in the repeater table is driven by a data item that you insert in the control.

The following image shows a static repeater table when it is inserted in an active report.

![Figure 23. Static repeater table control](image)

The data repeater table is the repeater table data container that is available for all report types. The static repeater table is available in the Toolbox tab only when you are authoring an active report.

**Decks and Data Decks**

Use decks and data decks to show different objects and different data respectively based on a selection in another control. For example, in a deck, clicking a radio button in a radio button group control shows a list object while clicking a different radio button shows a chart object.

The following image shows a deck control when it is inserted in an active report.
Decks are composed of cards. In static decks, you define the number of cards in the deck and you insert the objects that you want in each card. This allows you to create cards that can contain different objects, such as pie chart in one card and a crosstab in another card. In data decks, the cards are defined by a data container or data items inserted in the deck, and a data item determines which card appears.

**Tab Controls and Data Tab Controls**

Use tab controls and data tab controls to define multiple pages for the same area of a report. In data tab controls, the tabs are driven by a data item that you insert in the control.

The following image shows a tab control when it is inserted in an active report.

![Figure 25. Tab control](image)

**Button Bars and Data Button Bars**

Use button bars and data button bars to add a group of push buttons. In data button bars, the buttons are driven by a data item that you insert in the control.

The following image shows a button bar control when it is inserted in an active report.

![Figure 26. Button bar control](image)

In reports, users can click only one button at a time.

**Toggle Button Bars and Data Toggle Button Bars**

Use toggle button bars and data toggle button bars to add a group of buttons that change appearance when pressed. In data toggle button bars, the buttons are driven by a data item that you insert in the control.

The following image shows a toggle button bar control when it is inserted in an active report.

![Figure 27. Toggle button bar control](image)

In reports, users can click one or more buttons simultaneously.
Radio Button Groups and Data Radio Button Groups

Use radio button groups and data radio button groups to group a set of buttons that have a common purpose. In data radio button groups, the radio buttons are driven by a data item that you insert in the control.

The following image shows a radio button group control when it is inserted in an active report.

![Radio Button Group Control](image1)

*Figure 28. Radio button group control*

In reports, users can click only one radio button at a time.

Check Box Groups and Data Check Box Groups

Use check box groups and data check box groups to group a set of check boxes. In data check box groups, the check boxes are driven by a data item that you insert in the control.

The following image shows a check box group control when it is inserted in an active report.

![Check Box Group Control](image2)

*Figure 29. Check box group control*

In reports, users can select one or more check boxes simultaneously.

Drop-Down Lists and Data Drop-Down Lists

Use drop-down lists and data drop-down lists to provide a list of items that users can choose from. In data drop-down lists, the lists are driven by a data item that you insert in the control.

The following image shows a drop-down list control when it is inserted in an active report.

![Drop-Down List Control](image3)

*Figure 30. Drop-down list control*

In reports, users can select only one item at a time.

List Boxes and Data List Boxes

Use list boxes and data list boxes to provide a list of items that users can choose from. In data list boxes, the lists are driven by a data item that you insert in the control.

The following image shows a list box control when it is inserted in an active report.
In reports, users can select one or more items in a list box.

**Iterators and data iterators**

Use these controls to allow users to navigate or filter values in a control by clicking buttons such as previous and next. In data iterators, the iterations are driven by a data item that you insert in the control.

The following image shows an iterator control when it is inserted in an active report.

![Figure 32. Iterator control](image.png)

**Discrete values sliders, continuous values sliders, and data discrete values sliders**

These controls allow users to navigate or filter data using a slider bar. Discrete and data discrete values sliders allow users to slide between individual values, such as months. In data discrete values sliders, the values are driven by a data item that you insert in the control. With continuous values sliders, users slide through numeric values between minimum and maximum values that you define, such as 0 and 100.

The following image shows a discrete value slider control when it is inserted in an active report.

![Figure 33. Discrete values slider control](image.png)

**Buttons**

Use buttons to add individual push buttons to a report.

The following image shows a button control when it is inserted in an active report.
Data Containers

Lists, crosstabs, charts, maps, visualizations, and repeater tables are also controls in an active report, as you can add interactive behavior to them.

Add Data to a Control

For every control that you insert in a report, you must add the data that you want to appear in the control.

Add data to a data-driven control

For data-driven controls, you insert data items from the package.

Procedure

1. On the Source tab, drag query subjects or query items to the control.
2. To add an image to the data in the control, drag the appropriate data item to the Icon box.
3. If the data that will appear in the control depends on other data that you do not want to show, drag the appropriate data item to the Extra Items box.
   For example, you add Product type to a data list box control and you want to filter the product types that appear in the control based on a product line selected in another control. To filter the data, you must drag Product line to the Extra Items box.

   Tip: IBM Cognos Report Studio can automatically copy data items from other controls when you create connections between controls.
4. If the control is a data deck and you want to create a default card that will appear when no other card matches the variable value passed to the deck, in the Properties pane, click Yes for the Default Card property. Then click the No Data Contents tab for the deck and insert the objects that you want to appear on the default card.

Related tasks:

“Define a Connection Between Controls” on page 192
Define a connection between controls to link them together. When you link controls, an action performed on one control affects the behavior of the other control.

Add data to a static control

For static value driven controls, you manually define the data in a data table.

Procedure

1. In the report, click the definition icon of the control.

   Tip: The definition icon is visible only when the visual aid Show Interactive Object Controls is enabled.
2. To add a new object, such as a card in a deck or a button in a button bar, under Data Table, click the new button.
3. In **Data Table**, define the values that you want to appear in the control. Values are organized by data item. In addition to the default data item (Label) and values provided, you can create your own data items and values. For example, to add an icon to each value, you must create a data item for the icons.

- To define a new data item, click the new button ![new](image) and type the name of the data item that you want to create.
- Click inside each table row and type the name of the value.
- To add translations for the values, to support users in multiple languages, click the ellipsis (...) button, choose the languages that you want to support by clicking the add button ![add](image), and type the translated text for each language added.
- If you created new data items, in the **Definition** box, click the **Label** drop-down list and select the data item that you want to use in the control.
- To add an icon to each value, select the **Icon** check box, click the drop-down list, and select the data item that contains the icons that you want to use. You cannot add an icon to values in static decks and discrete values sliders. To add an icon to values in a static repeater table, insert the **Image** object from the **Toolbox**. Then specify the data item for the icons by referencing the data item in the **Properties** pane.

4. Optional: If you want to copy all data values to another static control, click the copy control definition icon ![copy](image). After you have copied the data values of the control, click the definition icon of another static control, and then click the paste control definition icon ![paste](image) to copy the data values.

   **Tip:** You can also drag data items from one control and drop them in another control. This action moves the data item from one control to another control instead of copying the item. You can drag and drop data items only to static controls that can contain objects, such as a deck, tab control, and repeater table. In addition, the data table of the target control must contain a data item with the same name as the data item you are moving, such as Label.

5. If the control is a deck, click the left and right arrows to navigate to each card of the deck and insert the objects that you want to appear on each card. To create a default card that will appear when no other card matches the variable value passed to the deck, in the **Properties** pane, click **Yes** for the **Default Card** property. Then click the **No Data Contents** tab ![tab](image) for the deck and insert the objects that you want to appear on the default card.

**Related tasks:**

* [Referencing data items defined in a static control](#)

   If you insert an object in a static control, such as an image in a static repeater table, you can use a data item in the control to populate the object.

**Referencing data items defined in a static control**

If you insert an object in a static control, such as an image in a static repeater table, you can use a data item in the control to populate the object.
Before you begin

You must create the data item in the data table of the control before you can reference the item.

About this task

You can reference data items only for text items, images, hyperlinks, and HTML items inserted in decks, tab controls, and repeater tables.

Procedure

1. In the static control, click the object for which you want to reference a data item.
2. In the Properties pane, click the Source Type property and choose Control Data Item Value. The Control Data Item Value property appears below the Source Type property.
3. Click the Control Data Item Value property and select the data item that you want to reference.

   Tip: If the object is a hyperlink, you can specify the data item to be used to populate the hyperlink text as well as the data item to be used to populate the URL of each link.

Related tasks:

"Add data to a static control" on page 190

For static value driven controls, you manually define the data in a data table.

Define a Connection Between Controls

Define a connection between controls to link them together. When you link controls, an action performed on one control affects the behavior of the other control.

Procedure

1. Click the create a new connection icon of a control that you added to the report.

   Tip: The create a new connection icon is visible only when the visual aid Show Interactive Object Controls is enabled.
2. In the Source Control drop-down list, select the source control for which you want to define a connection.
3. Click the Data Item drop-down list under the source control and select the data item that will drive the behavior of the target control.
4. In the Target Control drop-down list, select the target control for which you want to define a connection.
   If the target control is a visualization, select a visualization data container.
5. Click the Data Item drop-down list under the target control and select the data item that will determine what appears in the target control.
   If the data item that you want to use is not in the query of the target control, and the source control is a data-driven control, select Copy Data Item from Source to copy the source data item specified in the previous step to the target control query. If the source control is a static control, select Use Source Definition.
If the target control is a data container, the copied data item appears shaded in the container to indicate that the column visibility property for the item is set to **Hidden**.

6. Click the **Behavior** drop-down list between the source and target controls and specify the type of relationship that you want to define between the two controls:
   - To select data in the target control based on what will be selected in the source control, click **Select**.
   - To filter data in the target control based on what will be selected in the source control, select **Filter**.

7. Below the **Behavior** area, click the active variable drop-down list and choose to create a new variable for the connection or use an existing variable.

   **Tip:** If there are no variables defined in the report, you cannot click the active variable drop-down list.

   a. If you create a new variable, type a name for the variable and specify its data type. You can also specify its default values.
   b. If you choose to use an existing variable, choose the variable that you want. You can also specify its default values.

8. Click **Connect**.

**Results**

When a connection is defined, IBM Cognos Report Studio creates the appropriate conditions for the selection behavior and reaction behavior of the source and target controls.

**Tip:** The interactive behavior icon of a control changes to indicate that a connection was defined. The **Active Report Controls** tab shows the controls and variables inserted in a report and their relationships to one another. You can click a control in this tab to quickly locate the control in the report as well as set default values for variables.

**Related tasks:**

- "Show or Hide a Column or Row" on page 197
  You can show or hide a column in a list or a column or row in a crosstab when the report is viewed.

- "Add data to a data-driven control" on page 190
  For data-driven controls, you insert data items from the package.

- "Add data to a static control" on page 190
  For static value driven controls, you manually define the data in a data table.

**Specify the Selection Behavior of a Control**

Specify the selection behavior of a control to determine what interactive behavior should occur when the control is clicked or when an object in the control, such as a button, check box, or data item, is selected.

Specifying the selection behavior of a control sets the value of active report variables defined in the report. The active report variable values are then used to specify the reaction behavior of controls. For example, you add two drop-down list controls to a report and you create a connection between them. The first drop-down list shows product lines and the second drop-down list shows product...
types. When a product line is selected in the first drop-down list box, you want to pass that information to the second drop-down list box so that it shows only the product types of the selected product line.

You can specify a selection behavior for any control, except for decks and data decks.

IBM Cognos Report Studio automatically sets active report variable values when you define a connection between controls. However, you may need to set the value of other variables to specify additional interactive behavior.

**Procedure**

1. Click the interactive behavior icon of the source control.

   **Tip:** The interactive behavior icon is visible only when the visual aid **Show Interactive Object Controls** is enabled.

   The **Report Behavior** box shows the relationship between the source and target controls when you defined a connection between the controls. You can specify default values for the variable associated with the control by clicking **Set Default Value(s)**.

2. In the **Behavior on Select** box, click inside the **Set Variable Values** box and then click the edit button that appears.

3. For each active report variable that you want to set as part of the selection behavior of the control, click the **Value** drop-down list and select one of the following choices:
   - If you want to clear the current value of the variable, click **Set to Empty**.
     
     **Tip:** Clearing the current value of a variable is useful when you have cascading controls. For example, if the parent control is reset to a default value, you want to reset the child control to a default value as well.
   - If you want to specify your own value, click **Type in a value** and type the value that you want to use.
   - If you want the variable values to be driven by a data item in the report, click the data item that you want to use.
   - If you do not want the variable to be used for the control, click **Do not set**.

4. To create a new active report variable, click **Create a New Variable**.

**Related tasks:**

- **“Specify the Reaction Behavior of a Control”**
  
  Specify the reaction behavior of a control to determine how a control will react when another control, or an object in the control, is selected.

**Specify the Reaction Behavior of a Control**

Specify the reaction behavior of a control to determine how a control will react when another control, or an object in the control, is selected.

You can specify the following reaction behaviors:

- Select an item in a control.
  
  For example, you specify a default value for a variable, and the variable drives a drop-down list control. When the report is run, you want the default value to appear in the control.
You can specify this reaction behavior for any control, except for single button controls.

- Filter data in a control.
  For example, you want to filter the product types in a drop-down list control based on the product line selected in another drop-down list control.
  You can specify this reaction behavior for drop-down lists, data drop-down lists, list boxes, data list boxes, radio button groups, data radio button groups, check box groups, data check box groups, lists, and crosstabs.

- Enable a control.
  For example, you want to make a control interactive only when a selection is first made in another control.
  You can specify this reaction behavior for any control, except for decks, data decks, tabs, data tabs, lists, crosstabs, charts, and maps.

In addition to these behaviors, you can specify the item to appear when nothing is selected in a list control, hide or show a column, and whether to allow data to be sorted in a list.

IBM Cognos Report Studio automatically sets the reaction behavior when you define a connection between controls. However, you may want to change the reaction behavior or specify additional interactive behavior.

**Procedure**

1. Click the interactive behavior icon of the control.
2. In the Reaction Behavior box, click the reaction behavior box that you want, such as Container Select, and click the edit button that appears.
   If a condition has not yet been defined, two dialog boxes appear. The second dialog box prompts you to define the condition expression that you want to use to specify the reaction behavior.
3. In the Item box on the left, define the left side of the condition expression.
   - To use a data item, click the Data Item radio button and then click the item that you want to use.
     The data items that are available in the drop-down list depend on the data items that you inserted into the control.
   - To use a static item, click the Value radio button, type the value that you want to use, and then specify the value's data type.
     The value must be defined as a value of the selected active report variable in the next step.
   - If you want to set the value of an active report variable to null, click the Empty Variable radio button.
4. Click the Operator drop-down list and select the operator that you want to use.
5. In the Item box on the right, define the right side of the condition expression.
   - To use a value in an active report variable, click the Active Report Variable radio button and then click the variable that you want to use.
     The variables that are available in the drop-down list depend on the variables that you defined in the report.
   - To use a static item, click the Value radio button, type the value that you want to use, and then specify the value's data type.
   - To use null as the value, click the Empty Data Item radio button.
Use null as the value to specify the reaction behavior when no data exists for the value selected in the source control.

6. If you want to apply the condition when no item is selected in the source control, clear the Drop this condition if the variable is empty check box.

Clearing the check box can improve performance and prompt users to perform an action. For example, a report contains a list with many rows of data, and the list is filtered by another control. When the check box is cleared, no data will appear in the list until an item is selected in the other control.

Related concepts:
“Recommendation - use master detail relationships when authoring decks” on page 197
When authoring deck controls in active reports, use master detail relationships to improve performance.

Related tasks:
“Specify the Selection Behavior of a Control” on page 193
Specify the selection behavior of a control to determine what interactive behavior should occur when the control is clicked or when an object in the control, such as a button, check box, or data item, is selected.

“Set the No Value List Item”
Set the no value list item to show a specific item in a control when no value is selected.

“Show or Hide a Column or Row” on page 197
You can show or hide a column in a list or a column or row in a crosstab when the report is viewed.

“Sort Data in a List” on page 197
You can specify whether users can sort data in a list in the report output.

Set the No Value List Item
Set the no value list item to show a specific item in a control when no value is selected.

For example, a report contains a drop-down list control that contains product lines. You create a no value list item named (All) to show all product lines when the report is initially viewed.

Procedure
1. Select the control.
2. In the Properties pane, for the No Value List Item property, click Show.
   A text item representing the no value list item appears in the control.
3. In the Properties pane, double-click the Label property.
4. In the Default text box, type the name of the no value list item.
5. To specify the text for other languages, do the following:
   • Click the add button.
   • Select the languages that you want.
   • In the Language column, click one of the languages that you selected.
   • Click the edit button and type the name of the no value list item for that language.
   • Repeat for all remaining languages.
6. In the Properties pane, click the Position in List property and specify where you want the no value list item to appear in the control.

Show or Hide a Column or Row

You can show or hide a column in a list or a column or row in a crosstab when the report is viewed.

Procedure

1. Click the title of the column or row that you want to show or hide.
2. If the data container is a crosstab, in the Properties pane, click the Visible property and click Yes or No.
3. If the data container is a list, in the Properties pane, double-click the Column Visibility property.
4. To make the list column visible when the report is viewed, click the Visible radio button.
5. To hide the list column when the report is viewed, click the Hidden radio button.
6. To make the list column visible based on a condition, click the Visible based on condition radio button, click the edit condition button, and specify the condition.
   • If you want to specify a value to use for the condition, click the Value radio button, type the value, and in the Data Type drop-down list, select the value’s data type.
   • If you want to use null as the value for the condition, click the Empty Variable radio button instead.
   • Click the Operator drop-down list and select the operator that you want to use for the condition.
   • In the Active Report Variable drop-down list, select the active report variable that you want to use for the condition.
   • If you want to apply the condition when no item is selected in a control that is connected to the list, clear the Drop this condition if the variable is empty check box.

Sort Data in a List

You can specify whether users can sort data in a list in the report output.

Procedure

1. Select the column that you want users to be able to sort.
2. In the Properties pane, for the Allow Sorting property, click Yes.

Recommendation - use master detail relationships when authoring decks

When authoring deck controls in active reports, use master detail relationships to improve performance.

Decks offer greater flexibility in active reports by allowing you to show different objects in each card of a deck. In data decks, the number of cards that are created depends on data items inserted in the deck. As a result, a data deck can contain many cards, which may reduce performance when the report is run and viewed. For example, if a data deck contains Product line and Years, there can be up to 20
cards in the deck (five different product lines multiplied by four different years). In
addition, filtering data in a data deck can affect performance when there are many
rows of data in the deck.

To improve performance, use a master detail relationship to filter data in a data
container inserted in a data deck control. A master detail relationship defined
between a data deck and a data container inserted in the deck results in a specific
number of cards generated for the deck, which serves as a way to filter the data in
the data container. For example, a data deck contains a list object that has many
rows of data, and you want to filter the list by product line. Creating a master
detail relationship between the deck and the list using Product line produces five
cards in the deck, one for each product line. When you filter by product line, the
appropriate card appears in the data deck.

In addition, specify Select as the behavior instead of Filter when defining the
connection between the data deck control and the control you want to use to filter
the data in the deck.

Tip: If the data container in the data deck is a chart, you must use a master detail
relationship if you want to filter data in the chart.

**Summarizing Data in the Active Report Output**

You can add an output summary to a list or crosstab control that reflects the data
that appears in the control when an active report is viewed. In visualizations, you
can specify how the data is summarized when users select multiple values in a
control that is connected to a visualization.

For example, the data in a list is driven by selecting an item in a data drop-down
list control. An output summary specified for the list is recalculated when a
different item is selected in the drop-down list.

**Procedure**

1. In a list or crosstab, follow these steps to add an output summary.
   a. Click the column or row for which you want to add an output summary.
   b. In the toolbar, click the summarize button and then, under **Summarize in Output**, choose the summary that you want to add to the active report output.

2. In a visualization, follow these steps to specify how to summarize data when
   users select multiple values in a control.
   a. In the visualization data container, in the **Values** section, click the box that
      represents the data values in the visualization. For example, in a heatmap,
      click the **Color** box.
   b. In the Properties pane, click the **Rollup Method** property and select the
      aggregation method that you want to use to summarize data in the
      visualization.

**Run or Save an Active Report**

You can run an active report in HTML format or save it as a MIME HTML (MHT)
file so that you can send the report to your consumers. The MHT file is viewable
in the Microsoft Internet Explorer or Mozilla Firefox browsers.
To assist you in resolving problems when you are authoring an active report, a run option that shows the state of active report variables in the report as you click various controls is available. By default, this option is enabled.

If you are using Microsoft Internet Explorer 6.0, you cannot open an active report in MHT format as a file. You can view the report only by using a URL.

**Before you begin**

To view an active report in MHT format in Mozilla Firefox, you must first download an UnMHT add-on.

**Procedure**

1. To disable the run option that can show the state of active report variables when the report is viewed, from the Run menu, click Run Options, and clear the Enable right-click debug menu check box.
2. To run a report, from the Run menu, click Run Active Report.
3. To save a report, from the Run menu, click Download Active Report, and when prompted, choose to save the report as an MHT file.

**Results**

When running a report with the Enable right-click debug menu check box selected, right-clicking in the report output window allows you to choose to view the active report variables in the report, reset variables, or print the report.

**Tip:** The right-click debug menu is not available if you run the report from IBM Cognos Connection.

**Related tasks:**

- “Drill-through links in active reports do not work” on page 506

When viewing an active report that contains drill-through links in Microsoft Internet Explorer 8 and later, the links do not work.

---

**Example - Create an Active Report Showing Revenue Data for Core Products**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create an active report showing revenue data for the core products Camping Equipment and Golf Equipment.

The revenue data is specific to countries and regions in the Americas. The report will allow users to filter data by product or by country or region.

To create this report, you will need to add two data drop-down list controls and a list container. The first data drop-down list control will contain the core products that users can select. The second data drop-down list control will contain countries and regions that users can select. The list will show revenue data for each core product and country or region. When users click different products or different countries and regions, the data in the list will be filtered based on the selections made.

**Create the active report**

Create a new active report and add the objects that you need to build the core products revenue report.
Procedure
1. Open IBM Cognos Report Studio with the GO Data Warehouse (analysis) package.
2. In the Welcome page, click Create new.
3. In the New dialog box, click Active Report.
4. Double-click the report title and type Core Products Revenue
5. Click the Toolbox tab and drag the Table object to the report.
6. Clear the Maximize width check box and then click OK to create a table with two columns and one row.
7. Drag the Data Drop-Down List object to each cell in the table.
8. Drag the List object to the report so that it appears under the table.

Add data to the data drop-down list controls
This task adds the required data items to the two drop-down list controls in the report.

Procedure
1. Right-click the first data drop-down list control and click Go to Query. Query1 opens in the Query Explorer.
2. Click the Source tab and do the following:
   • Expand the Sales and Marketing (analysis) folder and the Sales namespace.
   • Expand the Products dimension and then the Products hierarchy.
   • Drag Product line to the Data Items area.
3. Click Product line and in the Properties pane, do the following:
   • Click the Name property, delete the existing name, and type Core products
   • Double-click the Set Definition property.
   • Click the new button and then click Exclude.
   • In the Available members box, expand Sales and Marketing (analysis), Sales, and Products.
   • Drag the Mountaineering Equipment, Personal Accessories, and Outdoor Protection members to the Members box.
   The Core products member set now contains only Camping Equipment and Golf Equipment.
4. Click OK twice.
5. Pause the pointer over the page explorer button and click Page1.
6. Click the Data Items tab and drag Core products from Query1 to the drop-down list box in the control.
7. Click the control and in the Properties pane, for the No Value List Item property, click Show.
The Properties pane for the no value list item property appears.
8. Double-click the **Label** property, type **Core products** in the **Default text** box, and click **OK**.

9. Right-click the second data drop-down list control and click **Go to Query**. Query2 opens in the Query Explorer.

10. Click the **Toolbox** tab and drag **Data Item** to the **Data Items** area.

11. In the **Expression Definition** box, type

   children ([[Sales].[Retailers].[Retailers].[Region]–>[Retailers].[710])

   **Tip:** This expression returns the children of the **Americas** member. The MUN for **Americas** is used instead of the member name.

12. In the **Properties** pane, click the **Name** property, delete the name and type **Countries and regions**.

13. Pause the pointer over the page explorer button and click **Page1**.

14. Click the **Data Items** tab and drag **Countries and regions** from Query2 to the drop-down list box in the control.

15. Click the control and in the **Properties** pane, for the **No Value List Item** property, click **Show**.

   The **Properties** pane for the no value list item appears.

16. Double-click the **Label** property, type **Countries and regions** in the **Default text** box, and click **OK**.

### Add data to the list

This task adds the required data items to the list.

**Procedure**

1. Pause the pointer over the query explorer button and click **Query1**.

2. Copy the **Core products** data item.

3. Pause the pointer over the query explorer button and click **Query3**.

4. Paste the **Core products** data item into Query3.

5. Repeat steps 1 to 4 to copy the **Countries and regions** data item from Query2 to Query3.

6. Pause the pointer over the page explorer button and click **Page1**.

7. Click the **Data Items** tab and drag **Core products** and **Countries and regions** from Query3 to the list.

8. Click the **Source** tab and do the following:

   * Expand the **Products** dimension and then the **Products** hierarchy.
   * Drag **Product type** to the list, inserting it between **Core products** and **Countries and regions**.
   * Expand **Sales fact**.
   * Drag **Revenue** and **Planned revenue** to the list, to the right of **Countries and regions**.
Add interactivity to the report

This task defines connections between the various controls and sets an active report variable so that the controls can interact with each other when users view the report.

Procedure

1. Click the interactive behavior icon of the first data drop-down list control.
2. Click Create a New Connection.
3. Click the Target Control drop-down list and click List1.
4. Ensure that the following information appears in the dialog box and click Connect.

   • In the Behavior drop-down list, Filter is selected.
   • In the Data Item drop-down list for both controls, Core products is selected.
   • In the Active Report Variable box, type Core products Variable 1.
5. In the Control drop-down list, click Data Drop-Down List2.
6. Click Create a New Connection.
7. Click the Target Control drop-down list and click List1.
8. Ensure that the following information appears in the dialog box and click Connect.

   • In the Behavior drop-down list, Filter is selected.
   • In the Data Item drop-down list for both controls, Countries and regions is selected.
   • In the Active Report Variable box, type Countries and regions Variable 1.
9. In the Control drop-down list, click Data Drop-Down List1.
10. In the Behavior on Select section, click the edit button.
11. Set Countries and regions Variable 1 to Set to Empty and click OK twice.

   When users select a different product in the first data drop-down list control, the second data drop-down list value will reset to the no value list item.

Specifying default values for the data drop-down list controls

This task specifies default values for the two drop-down list controls in the report.

Procedure

1. Click the Active Report Variables tab.
2. Expand Core products Variable 1 and click Set Default Values.
3. In the Values box, double-click Camping Equipment and click OK.
4. Expand Countries and regions Variable 1 and click Set Default Values.
5. In the Values box, double-click United States and click OK.

Revenue data for core products report output

When initially viewed, the revenue data for core products active report shows a list containing data for all core products in all countries and regions.
Selecting a product from the core product drop-down list or selecting a country or region from the country and region drop-down list filters the list to the selections made.

The following image shows the data that appears in the list when Camping Equipment and United States are selected in the two drop-down lists.

![Figure 35. Revenue data for core products report](image)

**Example - Sales analysis active report**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create an active report analyzing sales by region and by product line.

You will use a list to show sales by region, with a slider control to allow users to move between different regions. Users will have the option to show or hide the slider control by respectively selecting or clearing a check box.

You will use a pie chart to show sales by product line. Users will be able to click product lines in the chart legend to highlight sales in the chart.

**Creating a new active report**

Create a new active report and add the objects that you need to build the sales analysis report.

**Procedure**

1. Open IBM Cognos Report Studio with the **GO Sales (query)** package.
2. In the **Welcome** page, click **Create new**.
3. In the **New** window, double-click **Active Report**.
4. Double-click the report title and type **Sales Analysis**
5. Click the **Toolbox** tab, drag **Table** to the report, specify two columns and one row, and click **OK**.
6. Drag **Check Box Group** to the first table cell.
7. Select the second and third check boxes and delete them.
   Only one check box is required for this report.
8. Drag **Deck** to the second table cell.
9. Drag **Data Discrete Values Slider** to the deck, in the space under **Card 1**.
   The slider control becomes the first card in the deck.
10. Select the slider control and in the Properties pane, click the Track Length property, change the length of the slider track to 500 px, and click OK.
11. Optional: To animate the movement of the slider when users view the report, set the Animate property to Yes.
12. Optional: To update active report variables associated with the slider when users move the slider thumb, set the Update Variables Live property to Yes. The report will be updated live when the thumb is moved. If set to No, the report is updated only when the thumb is stopped and released.
13. Drag Block below the table.
14. Drag List below the block.
15. Insert another block below the list.
16. Click the second block and in the Properties pane, double-click the Padding property.
17. Specify 20 px for the top of the block and click OK.
18. Drag Data Deck below the block.
19. Drag Chart to the data deck.
   a. In the left column, click Pie, Donut.
   b. In the right column, click Exploded Pie with 3-D Effects and Flat Bevel.
      Tip: This pie chart type is available only when the Use legacy chart authoring check box (Tools > Options > Advanced tab) is cleared. To find this specific pie chart type, hover over each choice to view its tooltip.
   c. Click OK.

Results

The active report now contains all of the objects that you need to build the sales analysis report. In the following tasks, you will add data to each control and define connections between controls so that they can interact with each other.

Adding data to the check box group control

This task defines the data that will appear in the check box group control. The data will appear as the check box label.

Procedure

1. Click the Check Box Definitions icon.
2. Under Data Table, click Check Box Label 1 and then click the ellipsis button.
3. In the Default text box, type Show Region and click OK twice.

Adding a default card to the deck and data to the data discrete values slider

This task adds a default card to the deck and defines the data that will appear in the data discrete values slider control.

No content will be added to default card so that the slider will be hidden when the check box in the check box group control is cleared. The slider control is in the first card of the deck.
Procedure

1. Click the container selector (three orange dots) of the deck so that the entire deck is selected.

   **Tip:** The container selector is visible only when the visual aid **Show Container Selectors** is enabled.

2. In the **Properties** pane, set the **Default Card** property to **Yes**.
   
   A **No Data Contents** tab is created for the deck.

3. Click the **Deck** tab.

4. Click the **Source** tab and do the following:
   
   a. Expand **Sales (query) > Retailers**.
   
   b. Drag **Region** to the **Label** box of the slider.

**Defining a connection between the check box group and the deck**

This task defines a connection between the check box group control and the first deck control that you inserted in the report. The connection will allow users to show or hide the slider control when the check box is selected or cleared.

Procedure

1. In the check box group control, click the **Create a New Connection** icon.

2. Ensure that the following information is specified:
   
   - **Deck1** appears in the **Target Control** list.
   
   - **Select** appears in the **Behavior** list.
   
   - **Label** appears in the **Data Item** list for the source control.

3. Under **Target Control**, click the **Data Item** list and select **Use Source Definition**.
   
   The data value defined for the check box group, Show Region, is copied to the deck control.

4. Optional: In the **Active Report Variable** box, change the default name to a more meaningful name.

5. Click **Connect**.

**Adding data to the list object**

This task adds data items and an active report summary to the list object.

Procedure

1. Click the **Source** tab and drag the following data items to the list:
   
   - **Sales (query) > Retailers > Region**.
   
   - **Sales (query) > Products > Product line**.
   
   - **Sales (query) > Sales > Quantity**.
   
   - **Sales (query) > Sales > Revenue**.
   
   - **Sales (query) > Sales > Planned revenue**.

2. Pause the pointer over the **Query Explorer** icon and click **Query**.
3. Click the Toolbox tab and drag Data Item to the Data Items pane. The Data Item Expression window appears.

4. Copy and paste the following expression into the Expression Definition box and click OK:
   
   ```
   ([Revenue] - [Planned revenue]) / abs([Planned revenue])
   ```
   
   This data item calculates the percentage difference between revenue and planned revenue.

5. In the Properties pane of the data item, modify the following properties:
   a. In the Name property, delete the default name and type % Difference.
   b. Change Aggregate Function to Calculated.

6. Pause the pointer over the Page Explorer icon and click Page1.

7. Click the Data Items tab and drag % Difference from Query2 to the end of the list.

8. Click the % Difference column body in the list.

9. In the Properties pane, double-click the Data Format property and do the following:
   a. Click the Format type list and select Percent.
   b. In the list of properties, click the No. of Decimal Places, select 2, and click OK.

10. With the % Difference column still selected, in the toolbar, click the Summarize icon and under Summarize in Output, click Total.

### Defining a connection between the slider and the list object

This task defines a connection between the slider control and the list object. The connection will allow users to filter data in the list as the slider thumb is moved.

**Procedure**

1. In the slider control, click the Create a New Connection icon.

2. Specify the following information:
   a. In the Target Control list, click List1.
   b. In the Behavior list, click Filter.
   c. Ensure that Region appears in the Data Item list for both the source and target controls.

3. Optional: In the Active Report Variable box, change the default name of the variable that will be created when the connection is defined to a more meaningful name.

4. Click Connect.

### Adding a title to the list

This task adds a title above the list. The title will change according to the region selected in the slider.

**Procedure**

1. Click the Toolbox tab and drag Text Item to the block that you inserted above the list.
2. Type Sales by Region; add a blank space, and then click OK.

3. Select the text item in the report and in the Properties pane, modify the following properties:
   a. For the Foreground Color property, specify blue and click OK.
   b. For the Font property, specify 12 pt as the size and bold as the weight and click OK.

4. From the Toolbox tab, drag Variable Text Item to the right of the text item.

5. Select the name of the variable that you created when you defined the connection between the slider and the list.

6. Repeat step 3 for the variable text item.

---

**Adding data to the data deck and chart**

This task adds data items to the data deck and to the pie chart.

**Procedure**

1. Click the Source tab and drag the following data item to the Values box of the data deck:
   Sales (query) > Products > Product line

2. Drag the following data items to the chart:
   - Sales (query) > Sales > Quantity to the Default measure box.
   - Sales (query) > Products > Product line to the Series (pie slices) box.
   - Sales (query) > Retailers > Region to the Categories (pies) box.

3. With the pie chart selected, in the Properties pane, click the Exploded Slices property and do the following:
   a. Select the default value and click the Delete icon.
   b. Click the New icon.
   c. Change Exploded amount to 25.
   d. In the Exploded slice section, click Expression and then click the ellipsis.
   e. In the Expression Definition box, type the following, and click OK three times.
      \[ \text{[Query3].[Product line]} = \text{[Query4].[Product line]} \]

4. In the Properties pane, click the Clickable Regions property, select the Legend Labels check box, and click OK.

   By default, the Intersections check box is selected. By making the legend labels clickable, users will be able to click different pie slices in the chart or different product lines in the legend.

5. Optional: If you do not want to use the pie chart default colors, specify the colors that you want by clicking the available properties in the Color & Background property group.

   For example, you can change the color palette for the chart by clicking the Palette property.

---

**Defining a connection between the chart and the data deck**

This task defines a connection between the pie chart and the data deck. The connection will create an active report variable that will be used to update the title that will be inserted above the chart. The title is described in the next topic.
Procedure

1. In the pie chart, click the Create a New Connection icon.

2. Specify the following information:
   a. In the Target Control list, click Data Deck1.
   b. Ensure that Select appears in the Behavior list.
   c. Ensure that Product line appears in the Data Item list for both the source and target controls.

3. Optional: In the Active Report Variable box, change the default name of the variable that will be created when the connection is defined to a more meaningful name.

4. Set Camping Equipment as the default value for the variable by doing the following:
   a. In the Active Report Variable box, click Set Default Value(s).
   b. In the Values box, double-click Camping Equipment and then click OK.

5. Click Connect.

Adding a title to the chart

This task adds a title above the pie chart. The title will change according to the product line selected in the chart legend.

Procedure

1. Click the Toolbox tab and drag Text Item to the block that you inserted above the chart.

2. Type Sales by Product Line:, add a blank space, and then click OK.

3. Select the text item in the report and in the Properties pane, modify the following properties:
   a. For the Foreground Color property, specify blue and click OK.
   b. For the Font property, specify 12 pt as the size and bold as the weight and click OK.

4. From the Toolbox tab, drag Variable Text Item to the right of the text item.

5. Select the name of the variable that you created when you defined the connection between the chart and the data deck.

6. Repeat step 3 for the variable text item.

Sales analysis report output

When initially viewed, the Sales Analysis active report shows a Show Region check box that is cleared, a list, and a pie chart with Camping Equipment selected in the chart legend.

Selecting the Show Region check box shows the region slider. Moving the slider thumb filters the list to the specified region. Clicking a product line in the pie chart legend highlights the product line in the chart.

The following image shows the Sales Analysis report with the Show Region check box selected, the slider thumb moved to Northern Europe, and with Camping Equipment selected in the chart legend.
Example - Network diagram visualization

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a visualization in IBM Cognos Active Report that shows the hierarchies of employee by region.

You will use a network diagram visualization to show employees by region, with a data drop-down list to allow users to move between different regions.

Adding a new visualization in IBM Cognos Active Report

Create a visualization and add the objects that you require to build the network diagram.

Procedure

1. Open IBM Cognos Report Studio with the GO Sales (query) package.
2. In the Welcome page, click Create new.
3. In the New window, double-click Active Report.
4. Double-click the report title and type Employees by Region.
5. Click the Toolbox tab, drag a Visualization icon to the work area.
6. From the Visualization Gallery, select a Network diagram icon.
7. Click OK.

Figure 36. Sales Analysis active report
8. From the Toolbox tab, drag a Data Drop-Down List to the report.

Results

The visualization now contains the objects that you require to build the network diagram. In the following tasks, you will add data to each control and define connections between controls so that they can interact with each other.

Adding data to the nodes data container

In IBM Cognos Active Report, the nodes data container defines the items that you want to link together.

Procedure

Click the Source tab, and complete the following steps:

1. Expand Sales (query) and then Sales staff.
2. Drag Sales staff code to the Nodes1 data container, under Node (numeric).
3. Drag Staff name to the Nodes1 data container, under Name.
4. Drag Sales region to the Nodes1 data container, under Extra categories.

Adding data to the links data container

In IBM Cognos Active Report, the links data container defines relationship between the items that you want to link together.

Procedure

Click the Source tab, and complete the following steps:

1. Expand Sales (query) and then Sales staff.
2. Drag Manager code to the Links1 container, under From node (numeric).
3. Drag Sales staff code to the Links1 data container, under to node (numeric).
4. Drag Sales region to the Links1 data container, under Extra categories.

Defining a connection between the data drop-down list and the visualization object

This task in IBM Cognos Active Report defines a connection between the data drop-down control and the visualization object. The connection will allow users to filter data in the list as different regions are selected.

Procedure

1. From the Source tab, expand Sales (query) and drag Sales region to the drop-down list box of the data-drop-down list control.
2. In the data-drop-down control, click the Create a New Connection icon.
3. Complete the following steps:
   a. In the Source Control list, click Sales region.
   b. In the Behavior list, click Filter.
   c. In the Target Control list, leave the default as Links1.
4. Click Connect.

5. In the data-drop-down control, click the Create a New Connection icon again.

6. Complete the following steps:
   a. In the Source Control list, click Sales region.
   b. In the Behavior list, click Filter.
   c. From the variable drop-down list, click Reuse an existing variable.
   d. In the Target Control list, select Nodes1.

7. Click Connect.
8. Select the Links1 data container.

9. In the content pane, click the Active Report Controls tab.
10. In the Active Report Controls pane, expand and click Data Drop-Down List1 and click Set Default Value(s).
11. In the Default Variable Values window, under Values, select Northern Europe and use the arrow icon to move it to Selected values.
12. Click OK.
13. Run the report.

Network diagram visualization output

When initially viewed, the network diagram visualization in IBM Cognos Active Report shows the employee network for the Americas region.

Select different regions from the drop-down list and the employee network for that region is displayed.

The following image shows the Employees by Region report with the drop-down list selected for the Northern Europe sales region.

Figure 37. Network diagram visualization
Chapter 9. Relational Reporting Style

The relational reporting style is recommended for relational data sources. Relational data is best represented by lists. This data is organized in IBM Cognos Report Studio by query items.

In relational reporting, you summarize data by using headers and footers in lists, summary functions, and within detail aggregation. You focus data in relational reporting with summary or detail filters.

IBM Cognos Query Studio is also available to work with relational data.

Add Relational Data to a Report

Select the data items to appear in the report.

**Note:** We recommend using relational data in the relational reporting style. However, if you are using dimensional data, see “Add Dimensional Data to a Report” on page 253.

If you frequently use items from different query subjects or dimensions in the same reports, ask your modeler to organize these items into a folder or model query subject and then to republish the relevant package. For example, if you use the product code item in sales reports, the modeler can create a folder that contains the product code item and the sales items you need.

**Procedure**

1. From the **Source** tab, drag each query item to the report.

   **Note:** A flashing black bar indicates where you can drop an item. Items inserted in the report appear on the **Data Items** tab.

   Other ways to select data items are to double-click each item or to right-click each item and click **Insert**.

2. If you want to remove a data item from the report, select it and click the delete button. To remove the data item from the report but keep it on the **Data Items** tab, click the cut button instead.

3. For more information about a query item, right-click the item in the **Source** tab and click **Properties**. For example, when working with relational data sources, the **Usage** property identifies the intended use for the data represented by the data item. You can also obtain more information by clicking **Lineage**.

Group Relational Data

Group data items in a list report to remove duplicate values. For example, you have a report that shows all the products purchased and their product type. You group the Product type column so that each product type cell spans the products purchased cells.
You can specify whether grouping a column automatically makes it the first column in the list by selecting **Group and automatic summary behavior for lists** in the IBM Cognos Report Studio options. With this option selected, if the list contains any fact or measure columns, footers that show the summary values for the columns are automatically created for each group. Grouping a second column makes it the second column. When you ungroup a column, that column becomes the first column after the grouped column.

After a column is grouped, you can move it elsewhere in the report.

You can also group data items in repeaters and in page sets.

In addition to grouping, you can specify the sort order of data items. In lists, you can sort data items within groups as well as ungrouped items. In crosstabs, you can sort row and column items. For more information, see “Perform Advanced Layout Sorting” on page 235.

**Tip:** The Succession Report sample report in the GO Data Warehouse (analysis) package includes grouping. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Procedure**

1. Click the column on which to group.
   
   You can click either the column title or one of the column cells.
   
   **Tip:** To perform multiple groupings at once, use Ctrl+click or Shift+click.
   
2. From the **Structure** menu, click **Group/Ungroup**.

**Results**

A symbol appears indicating that the column is grouped. You can control when its values appear by setting the group span, and the column appears in the list of headers and footers that you can add to the report.

**Note:** Grouping all columns in a list is equivalent to creating a single-edge crosstab.

**Perform Advanced Grouping**

You can perform more advanced groupings in a list to get the results you want.

For example, you can

- view the entire grouping structure and make changes to the grouping order
- group a data item that appears in a query but not in the layout
- specify a list of properties for a group

You can also perform advanced sorting.

**Procedure**

1. Click a column in the list.
2. Click the select ancestor button in the title bar of the **Properties** pane and click **List**.
Tip: You can also click the container selector (three orange dots) of the list to select it.

3. In the **Properties** pane, double-click the **Grouping & Sorting** property.

4. To change the grouping order, in the **Groups** pane, click the grouped item in the **Groups** folder and drag it to the grouping hierarchy.

5. To create a new grouping, in the **Data Items** pane, click the data item and drag it to the **Groups** folder in the **Groups** pane.

6. To specify a list of data items for a group, do the following:
   - Click the group.
     
     Tip: Click **Overall** to specify a list of data items for the overall group. The overall group is the grouping of all data into one group. You can specify a list of data items for the overall group to add overall totals to the list.

   - Click the properties button.
   - Select the appropriate check boxes.

### Creating relational custom groups

Create custom groups to classify existing data items into groups that are meaningful to you.

You can use custom groups to convert numeric values into broader categories. For example, you can break down sales results into low, medium, and high categories. Or you can reduce the number of values into a smaller, more meaningful groups. For example, you can change a list of employees into separate teams or departments.

When you create custom groups in a list, a new column is added to the report, with the name *data item* (Custom). You can use this new column to group or sort the report. If you want to show only the new groups, you can delete the original column.

### Procedure

1. Click the column that you want to group on and, from the **Data** menu, click **Calculate > Define Custom Groups**.
   
   Tip: You can click either the column heading or one of the column cells.

2. Click the **New Custom Group** icon.

3. To group by values that you select, click **New Select Values Groups**, type a **New Group Name**, select the required values from the **Values** box, and move the values to the **Selected Values** box.

4. To group by a range of values, click **New Range Group**, type a **New Group Name**, and enter the **From** and **To** values.

5. If you do not want a group name to appear for remaining values, select **Do not show remaining values**. Clicking this option will produce empty cells for remaining values.

6. If you want the group name for remaining values to match each value, select **Use each remaining value as a group name**.

7. If you want to specify your own group name for all remaining values, select **Group remaining values into a single group** and type the name that you want.
8. If you are adding custom groups in a list, under New data item name, select either the default data item name or type a new data item name.

**Set the Group Span for a Column**

When columns are grouped, you can choose how often to show column names by changing the group spanning.

For example, when country or region and city are both grouped, you can choose to show the country or region name each time
- the country or region changes, by spanning Country or Region by Country or Region
- the city changes, by spanning Country or Region by City
- there is a new record, by specifying no spanning

Spanning one grouped column by another column is helpful if the second column contains many items.

**Procedure**

1. Click the column that will span the other column(s).
2. In the Properties pane, set the Group Span property to the column to span.

**Example - Remove Identical Values in a List**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a report that lists all product orders organized by order number. To make the report easy to read, you group the Order number column so that only one instance of each order number appears. However, because each order contains different products, the same order date appears for each product. You decide to show the order date only when the order number changes.

**Procedure**

1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click List and click OK.
4. In the Source tab, expand Sales and Marketing (query) and Sales (query).
5. Expand Sales order and add Order number by double-clicking it.
6. Expand Time dimension and add Date by double-clicking it.
7. Expand Product and add Product name by double-clicking it.
8. Expand Sales fact and add Quantity and Revenue by double-clicking them.
9. Group the Order number column.
10. Click the Date column.
11. In the Properties pane, set the Group Span property to Order number.

**Results**

When you run the report, the first row of the Order date column appears for each Order number row.
Example - Repeat a Column Value in a List

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a report that lists all products sold by the company, organized by the product line and product type. To make the report easier to read, you group the Product line and Product type columns so that only one instance of each column appears. However, because some product lines contain many different product types and products, you decide to show the product line for each product type.

Procedure

1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click List and click OK.
4. In the Source tab, expand Sales and Marketing (query).
5. Expand Sales (query) and Product and add the following data items by double-clicking them:
   - Product line
   - Product type
   - Product name
6. Expand Sales fact and add Product cost by double-clicking it.
7. Group the Product line and Product type columns.
8. Click the Product line column.
9. In the Properties pane, set the Group Span property to Product type.
10. Click any part of the list and then click the select ancestor button in the title bar of the Properties pane.

11. Click List.

Tip: You can also click the container selector (three orange dots) of the list to select it.

12. In the Properties pane, double-click the Font property.

13. In the Size box, click 8pt.

Results

When you run the report, the product line appears whenever the product type changes.

![Product Line and Product Type]({{image_url}})

Figure 38. List showing product line, product type, product name, and product cost

Divide Data into Sections

Create sections in a report to show a data item as the heading of a section. When you run the report, separate sections appear for each value.

Creating sections is similar to creating headers by grouping on a data item. The difference is that section headers appear outside the list, crosstab, chart, or repeater. In addition, you can group data items only in lists.

When working with dimensional data, you can also create page layers to show values on a separate page for each member.
**Procedure**

1. Click the column to make a section heading.
2. Do one of the following:

   • From the **Structure** menu, click **Section**.
     
     If the column is in a list or repeater, this menu option will create sections without creating a master detail relationship. This can improve performance when running the report.
     
     If the column is in a crosstab or chart, this menu option will create sections using a master detail relationship.

   • From the **Structure** menu, click **Section Using Master/Detail**.
     
     This menu option creates sections using a master detail relationship.

3. To add or remove section headers and footers, from the **Structure** menu, click **Headers & Footers**, click **List Headers & Footers**, and select or clear the appropriate check boxes.

   The section header and footer check boxes appear under **Overall header**.

**Remove Sections**

You can remove sections and reinsert the data item that was used to create the sections in the data container.

**Procedure**

1. From the **View** menu, click **Page Structure**.
2. Expand the page containing the data container with the section.
3. Expand **Page Body** until you see the data container in which you added the section.

   The data container is nested in the **List Column Body** object of the **List** object that is created when you add a section.

4. Drag the data container to **Page Body**.

   The data container appears as a node of **Page Body**.

5. Delete the **List** object.
6. From the **View** menu, click **Page Design**.

7. From the **Data Items** tab, drag the data item that was used as a section header back into the data container.

8. If you created sections in a list without creating a master detail relationship, click the data item, and then click the group/ungroup button to ungroup the item.

---

**Summarizing Data Relationally**

Summarize data in your reports to obtain totals, averages, and so on.

In a report, you can add both detail and summary aggregation. Detail aggregation, which is supported only for relational data sources, specifies how a data item is totaled at the lowest level in a report. In lists, detail aggregation specifies how the values that appear in the rows are totaled. In crosstabs, detail aggregation specifies how the values in the cells are totaled. For example, detail aggregation for a
measure like Revenue might be Total in both lists and crosstabs. In the following list report, this means that the values you see for the Revenue column represent the total revenue for each product type.

<table>
<thead>
<tr>
<th>Product Line</th>
<th>Product Type</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Equipment</td>
<td>Cooking Gear</td>
<td>272,835.984.18</td>
</tr>
<tr>
<td></td>
<td>Lantern</td>
<td>126,925.860.64</td>
</tr>
<tr>
<td></td>
<td>Packs</td>
<td>351,080.462.64</td>
</tr>
<tr>
<td></td>
<td>Sleeping Bags</td>
<td>309,172.886.85</td>
</tr>
<tr>
<td></td>
<td>Tents</td>
<td>528,221.720.02</td>
</tr>
<tr>
<td>Golf Equipment</td>
<td>Golf Accessories</td>
<td>51,514.342.60</td>
</tr>
<tr>
<td></td>
<td>Iron</td>
<td>254,514.337.99</td>
</tr>
<tr>
<td></td>
<td>Putters</td>
<td>106,184.271.37</td>
</tr>
<tr>
<td></td>
<td>Woods</td>
<td>313,998.414.85</td>
</tr>
</tbody>
</table>

*Figure 39. A list showing product line, product type, and revenue*

Summaries, which are supported for all data sources, specify how data items are totaled in the headers and footers of a list and in the total rows and columns of a crosstab. For list reports, these summaries only summarize the data that is visible on that page of the report.

You can specify detail and summary aggregation in different ways by using any of the following:

- aggregation properties that are specified in the model
- the **Auto Group & Summarize** property
- the summarize button in the toolbar
- aggregation properties for a data item
- summary functions in data item expressions
- the solve order of calculations

**Limitation**

If a summary is applied to a report that contains binary large object (BLOB) data, such as images or multimedia objects, you cannot also perform grouping or sorting.

**The Type of Data**

How data is aggregated also depends on the type of data that you are aggregating. Aggregation rules are applied differently to facts, identifiers, and attributes. For example, if you aggregate a data item that represents part numbers, the only aggregate rules that apply are count, count distinct, count non-zero, maximum, and minimum. For information about how to determine the type of data that a data item represents, see “Add Relational Data to a Report” on page 213 and “Add Dimensional Data to a Report” on page 253.

**Add a Simple Summary**

You can add simple summaries to the groups in a report by using the summarize button in the toolbar. This button provides a subset of the summary functions
available in IBM Cognos Report Studio. For list reports, a **Custom** option is also available so that you can add your own summary function in the expression of the data item.

The summarize button sets the rollup aggregate property for the data item to the selected summary, and places the data item into an appropriate footer. A footer is created for each group as well as an overall footer, unless they already exist.

In lists, the summary appears as a footer. If the column to which you added a summary is grouped, group and overall summaries appear. In crosstabs and charts, the summary appears as a node.

To change a summary, select it and, in the **Properties** pane, under **Data Item**, click **Rollup Aggregate Function** and choose a different function.

In crosstabs, you can add multiple summaries at the same level. For example, you have a crosstab with Product line as rows, Order year as columns, and Revenue as the measure. For Product line, you can add the **Total** summary as a header, which will total all revenue for each order year. You can then add the **Average** summary as a footer, which will give the average revenue of all product lines for each order year.

For information about adding a rolling or moving average, see "Rolling and Moving Averages" on page 272.

**Procedure**

1. Click the column to which to add a summary.
2. Click the summarize button and click a summary type.
3. To change the summary label, do the following:
   - Click the label.
   - In the **Properties** pane, under **Text Source**, set the **Source Type** property to the source type to define the label.
     For example, set it as **Data Item Value** to produce a dynamic label for the summary based on data item values.
   - Set the property below **Source Type** to specify the label.
     This property depends on the source type you chose. For example, if you chose **Data Item Value** as the source type, set the **Data Item Value** property to the data item to use to define the label.

**Set the Auto Group & Summarize Property**

Set the **Auto Group & Summarize** query property to specify whether IBM Cognos Report Studio should group non-fact data items (identifiers and attributes) and apply aggregate functions to aggregate fact data items in lists.

If you are using an OLAP data source, data is always summarized regardless of how this property is set.

**Procedure**

1. Pause the pointer over the query explorer button and click a query.
2. In the **Properties** pane, set the **Auto Group & Summarize** property:
To group non-aggregate fact data items and apply aggregate functions to aggregate fact data items in lists, set this property to **Yes**.

To render detail rows, set this property to **No**.

**Specify Detail or Summary Aggregation in the Model**

When working with relational or dimensionally-modeled relational (DMR) data sources, you can use the aggregation properties specified for the query item in the IBM Cognos Framework Manager model instead of specifying detail or summary aggregation in the report. The model specifies the default summary function for each query item.

Aggregation functions are mapped between data sources, Framework Manager, and IBM Cognos Report Studio.

**Specify Aggregation Properties for a Data Item**

When working with relational data sources and list reports, for each data item in a report, you can specify detail and summary aggregation properties to manage summaries without having to create complex data item expressions.

**Procedure**

1. Click the data item for which to set detail or summary aggregation.
2. In the **Properties** pane, set the **Aggregate Function** or the **Rollup Aggregate Function** property to a function.

**Use Summary Functions in Data Item Expressions**

You can use summary functions in data item expressions. The summary functions in the expression editor that have the same name as the summary functions available through the **Aggregate Function** and **Rollup Aggregate Function** properties operate the same way. For example, in a relational report, setting the **Aggregate Function** property to **Total** is the same as changing the expression of the data item to `total ([Revenue])`.

In general, report maintenance is easier if the **Aggregate Function** and **Rollup Aggregate Function** properties are used rather than adding summary functions to data item expressions. Use summary functions in expressions if the required summary is not supported as an aggregate or rollup aggregate property or if the complexity of the expression cannot be managed using the data item properties. Add a summary function to a data item expression if one of the following conditions applies:

- The underlying data source is relational, and you want to use database vendor-specific summary functions.
- You want to use summary functions that require more than one parameter, such as percentile.
- You require summary expressions that are not available in the aggregation properties, such as a for clause. You should use for clauses only in relational style reports (list reports).

For example, your report uses Product line, Product type, and Quantity. You want to calculate the percentage of the parent quantity that is represented by each product type. This requires you to compare the total quantity for each product type to the total quantity for the parent product line. In other words, your
expression requires summaries at different group levels. You use summary functions, such as aggregate, in the data item expression with a for clause to specify the group level as follows:

\[ \text{Quantity} / \text{aggregate ([Quantity] for [Product line])} \]

In some cases, you may need a summary that is different from the default. For example, if the default summary for Quantity is total, you can calculate the average as follows:

\[ \text{Quantity} / \text{average (aggregate ([Quantity]) for [Product line])} \]

The inner summary gives you the expected Quantity values at the Product Type level, which are then averaged to the Product Line level. This two-stage aggregation is required for OLAP data sources and recommended for relational data sources also.
Why is this necessary? Here’s what would happen if you simply averaged Quantity, as follows:

\[
\text{[Quantity]} / \text{average}\{\text{[Quantity]} \text{ for [Product line]}}\]

<table>
<thead>
<tr>
<th>Product line</th>
<th>Product type</th>
<th>Quantity</th>
<th>% of Product Line Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Equipment</td>
<td>Camping Gear</td>
<td>13,400,351</td>
<td>2,538,352%</td>
</tr>
<tr>
<td></td>
<td>Lanterns</td>
<td>4,020,755</td>
<td>914,306%</td>
</tr>
<tr>
<td></td>
<td>Packs</td>
<td>2,756,540</td>
<td>522,150%</td>
</tr>
<tr>
<td></td>
<td>Sleeping Bags</td>
<td>3,153,213</td>
<td>597,269%</td>
</tr>
<tr>
<td></td>
<td>Tents</td>
<td>3,164,265</td>
<td>599,352%</td>
</tr>
</tbody>
</table>

Camping Equipment - Summary: 27,301,149 5,174,500%

<table>
<thead>
<tr>
<th>Product line</th>
<th>Product type</th>
<th>Quantity</th>
<th>% of Product Line Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Golf Equipment</td>
<td>Golf Accessories</td>
<td>3,119,747</td>
<td>1,295,327%</td>
</tr>
<tr>
<td></td>
<td>Irons</td>
<td>351,445</td>
<td>151,256%</td>
</tr>
<tr>
<td></td>
<td>Putters</td>
<td>1,264,570</td>
<td>496,259%</td>
</tr>
<tr>
<td></td>
<td>Woods</td>
<td>317,939</td>
<td>122,827%</td>
</tr>
</tbody>
</table>

Golf Equipment - Summary: 5,413,701 1,975,760%

The denominator is now computed as the average of the detail Quantity rows in the underlying fact table. Since these values are much smaller than the pre-summarized Quantity values you see in the report, the resulting percentage is far too big, and not what we wanted.

**Tip:** The Total Revenue by Country or Region sample report in the GO Data Warehouse (query) package includes a total summary function. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Procedure**
1. Click a data item.
2. In the **Properties** pane, double-click the **Expression** property and add summary functions to the expression for the data item.

**Related concepts:**
- “Limitations When Using Clauses in Summary Functions” on page 532
A summary function that uses a `for` clause may yield unexpected results. These may include error messages, warnings, incorrect numbers, and more or fewer than expected rows, columns, or chart points and lines.

**Summary Functions**
This section describes the summary functions that are available in IBM Cognos Report Studio. Some functions, such as **Custom**, are available only when you click the summarize button on the toolbar. Additional summary functions are available in the expression editor.

**Automatic Summary**
Depending on the type of data item, applies the function **None**, **Summarize**, or **Calculated** based on the context in which the data item appears.
Calculated is applied if the data item expression
- contains a summary function
- is an if then else or case expression that contains a reference to at least a
  modeled measure in its condition
- contains a reference to a model calculation or to a measure that has the Regular
  Aggregate property set to a value other than Unsupported
- contains a reference to at least one data item that has the Rollup Aggregate
  Function property set to a value other than None

If the underlying data source is relational and if the data item expression contains
no summary functions and a reference to at least one fact that has the Regular
Aggregate property set to a value other than Unsupported, Summarize is applied.

In all other contexts, None is applied.

For relational and dimensionally modeled relational (DMR) data sources, if this
function is applied to a data item expression that has the average function,
weighted averages are computed based on the lowest level detail values in the
data source.

Note: This is the default function.

**Average**

Adds all existing values and then divides by the count of existing values.

**Calculated**

Specifies that all the terms within the expression for a data item are aggregated
according to their own rollup rules, and then the results of those aggregations are
computed within the overall expression.

For example, a list contains the data item Quantity with the Aggregation property
set to Total. You add a query calculation named Quantity Calculated. You define
its expression as Quantity + 100 and you set its Aggregation property to
Calculated. When you run the report, the values for Quantity Calculated are
computed by first adding all values for Quantity and then adding 100 to each
value.

Note: In crosstabs, this function overrides any solve orders that are specified.

**Count**

Counts all existing values.

If the underlying data source is dimensionally modeled relational (DMR), Count
behaves as follows if it is specified in the Aggregate Function or Rollup Aggregate
Function property for a data item.

<table>
<thead>
<tr>
<th>Object</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Count distinct is used. A warning appears when you validate the report.</td>
</tr>
<tr>
<td>Member set</td>
<td>Count distinct is used. A warning appears when you validate the report.</td>
</tr>
</tbody>
</table>
If you add the count summary to a non-fact column in a list and then group the column, the column will not be grouped when you run the report. To resolve this issue, group the column first before adding the count summary.

**Count Distinct**

Returns the total number of unique non-null records.

If the underlying data source is dimensionally modeled relational (DMR), count distinct is supported for levels, member sets, attributes, and measures when it is specified in the Aggregate Function or Rollup Aggregate Function property for a data item.

**Custom**

Summarizes data based on an expression that you define.

**Maximum**

Selects the largest existing value.

**Median**

Returns the median value of the selected data item.

**Minimum**

Selects the smallest existing value.

**None**

Does not aggregate values.

If the underlying data source is relational, the data item is grouped when the query property Auto Group & Summarize is set to Yes.

**Not Applicable**

This function is available only for the Aggregate Function property. It specifies that the aggregate attribute is ignored. For example, the calculation will be applied after the data is aggregated.

This function differs from the Calculated function, which applies aggregation to the rows and then performs the calculation.

For example, for the average ( [Quantity] ) + 5 expression, when the Aggregate Function property is set to Not Applicable, five is added to the average of the single Quantity value. When the function is set to Calculated, five is added to the average of a set of numbers.
This setting should not be applied to a simple model reference.

This setting is relevant only for reports that are upgraded from IBM Cognos ReportNet 1.x.

**Standard Deviation**

Returns the standard deviation of the selected data item.

**Note:** From a mathematical perspective, this function is not useful for small numbers of items and is not supported if the query property *Auto Group & Summarize* is set to *Yes*.

**Summarize**

Aggregates data based on model or data type information. This function can be thought of as a calculate and then aggregate rule. Data is aggregated using the following rules:

If the underlying data source type is relational and the data item or calculation is a reference to a single fact query item in the model that has the Regular Aggregate property set to *Sum*, *Maximum*, *Minimum*, *Average*, or *Count*, aggregate data using this function. Otherwise, data is aggregated according to the data type of the data item as follows:

- **Total** is applied for numeric and interval values.
- **Maximum** is applied for date, time, and date-time values.
- **Count** is applied for everything else.

The underlying data source type can be dimensional and the data item or calculation a reference to a single item in the model. In this case, if the Regular Aggregate property of the model item is not *unknown*, the summary function that corresponds to the Regular Aggregate property is applied. If the Regular Aggregate property is *unknown*, set the function to *Calculated*.

For example, a list contains the data item Quantity with the Aggregation property set to *Average*. You add a query calculation named Quantity Summarize. You define its expression as *Quantity + 100* and you set its Aggregation property to *Summarize*. When you run the report, the values for Quantity Summarize are computed by first adding 100 to each value for Quantity and then calculating the total. The total is calculated because Quantity contains numeric values and the expression for Quantity Summarize is not a simple data item reference. If Quantity Summarize is defined as Quantity, the function *Average* is applied to each value.

**Total**

Adds all existing values.

**Variance**

Returns the variance of the selected data item.

**Note:** From a mathematical perspective, this function is not useful for small numbers of items and is not supported if the query property *Auto Group & Summarize* is set to *Yes*. 
Mapping Aggregation Functions From Data Sources to Framework Manager and Report Studio

The summary functions available in IBM Cognos Framework Manager and IBM Cognos Report Studio reflect summary functions supported by relational and dimensional data sources. The following table shows how summary functions in data sources are mapped to Framework Manager and Report Studio.

<table>
<thead>
<tr>
<th>Data source</th>
<th>Framework Manager</th>
<th>Report Studio</th>
</tr>
</thead>
<tbody>
<tr>
<td>None specified, or none</td>
<td>Unsupported</td>
<td>None</td>
</tr>
<tr>
<td>average (avg)</td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td>count</td>
<td>Count</td>
<td>Count</td>
</tr>
<tr>
<td>count distinct</td>
<td>Count Distinct</td>
<td>Count distinct</td>
</tr>
<tr>
<td>maximum (max)</td>
<td>Maximum</td>
<td>Maximum</td>
</tr>
<tr>
<td>minimum (min)</td>
<td>Minimum</td>
<td>Minimum</td>
</tr>
<tr>
<td>median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td>standard deviation (stddev)</td>
<td>Standard Deviation</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>sum</td>
<td>Sum</td>
<td>Total</td>
</tr>
<tr>
<td>variance (var)</td>
<td>Variance</td>
<td>Variance</td>
</tr>
<tr>
<td>count non zero</td>
<td>Count Non-Zero</td>
<td>Automatic</td>
</tr>
<tr>
<td>average non zero</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
<tr>
<td>external</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
<tr>
<td>any</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
<tr>
<td>calculated</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
<tr>
<td>unknown</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
<tr>
<td>first_period</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
<tr>
<td>last_period</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
</tbody>
</table>

The following summary functions exist only in either Framework Manager or Report Studio. There is no corresponding summary function in data sources.

<table>
<thead>
<tr>
<th>Framework Manager</th>
<th>Report Studio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Automatic</td>
</tr>
<tr>
<td>Calculated</td>
<td>Calculated</td>
</tr>
<tr>
<td>No corresponding summary function</td>
<td>Summarize</td>
</tr>
<tr>
<td>No corresponding summary function</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

Focusing Relational Data

You can limit the data in your report with filters or query parameters. The method you choose should depend on the capabilities of the data source, the performance requirements, the persistence of the dataset, and how complex you want the report to be.

In IBM Cognos Report Studio, summary and detail filters are best for list reports in the relational reporting style. Summary filters are applied to summaries and detail filters are applied to details. These filters remove rows and columns from your reports.
You can also use prompts, or prompt controls, to allow report users to provide input for the filter.

**Create a Detail or Summary Filter**

Add a filter expression to focus a report and minimize processing time by excluding unwanted data. For example, you can filter data to show only customers who placed purchase orders that were valued at over one thousand dollars during the past year. When you run the report, you see only the filtered data.

Detail filters are applied only to detail data items from the data source, and not to summary data items. Examples of detail data items include customer names, product numbers, and order numbers. Detail filters are Boolean expressions that apply one or more conditions and evaluate to either true or false. If the conditions are true for a row of data, the row is retrieved from the data source and appears in your report. If the conditions are false, the row is not retrieved and does not appear in your report. You can specify whether detail filters are applied before or after summaries using the Application property.

**Tip:** The 2011 Sales Summary sample report in the GO Sales (analysis) package includes a detail filter. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Summary filters are applied to summary data items in a report. They are also Boolean expressions, however they are used to exclude specific groups of data items based on the group summaries. They are based on summary calculations.

You can combine multiple conditions using AND, OR, and NOT operators. By default, multiple conditions are combined with an AND operator, which means that all conditions must be met for the filter to take effect.

For relational data sources, if a filter expression is used in multiple reports or by different report authors, ask your modeler to create the filter as a standalone object in the model and include it in the relevant package. For information about creating filters in the package, see the IBM Cognos Framework Manager User Guide.

If you are filtering a multiple-fact query, you should understand the differences between using a detail filter and summary filter.

If you filter values that are floating-point numbers, you may encounter imprecise filtering results due to rounding or data truncation. Floating-point numbers may appear differently in the report than how they are stored in the data source. To ensure accurate results, your filter criteria should account for any rounding issues.

If you are working with a Microsoft SQL Server 2005 Analysis Services (SSAS) data source, we recommend that you not use OR filters in summaries. Doing so causes error characters (--) to appear in summaries instead of values.

**Procedure**

1. To add a filter that was already created in the package, from the Source tab, drag the filter to the report.

   You must drag the filter to a data container object that is linked to a query, such as a list, crosstab, or chart.
2. To create your own filter, from the Data menu, click Filters, and click Edit Filters.
   To create a simple filter condition based on only one data item, click Create Custom Filter.
   If more than one query is defined in the report, you must first select a data container object that is linked to a query.

   **Tip:** You can also define filters in Query Explorer. Pause the pointer over the query explorer button and click a query.

3. In the Filters dialog box, decide what type of filter to create:
   - To add a filter that will apply to detail values, click the Detail Filters tab.
   - To add a filter that will apply to summary values, click the Summary Filters tab.

4. Click the add button.

5. In the Create Filter dialog box, decide what conditions to include in the filter:
   - To create a simple filter condition based on only one data item, click Custom based on data item, select a data item from the query, and click OK.
   - To combine multiple filter conditions, click Combined, and click OK.
   - To create a filter condition that uses advanced calculations, click Advanced, and click OK.

6. If you create a simple filter condition based on one data item, do the following:
   a. In the Filter Condition window, under Condition, select the type of condition that you want.
   b. If you want to parameterize the filter so that users are prompted when they run the report, select the Prompt for values when report is run in viewer check box.
   c. To choose specific data item values for the filter condition, in the Values box, click Specific values and choose the values that you want to filter on.

      **Tip:** You can search for values by typing a string in the Keywords box. Click the arrow beside Search and select the search parameter that you want to use. If you select Use Tab as keyword delimiter, you can add a tab in the Keywords box only by copying and pasting a tab (or the string you want to search on with a tab in it), from another application, such as Notepad.

   d. To filter values based on a pattern, in the Values box, click Starts with, Ends with, Contains, or Matches SQL pattern, and type the pattern in the Value box.

      With the Matches SQL pattern choice, you can filter values by using standard SQL syntax to create a SQL supported pattern. You can use a % (percent) symbol to substitute for zero or more characters, or _ (underscore) to substitute for one single character. For example, the pattern '%a_b_c%' returns every value that contains the sequence 'a, any character, b, any character, c' anywhere in the string.

      If you search for patterns that include the '%' or '_' characters, specify an escape character in the Escape character (optional) box. The value must be a single character. When specified, each occurrence of the escape character...
causes the next character to be treated as text and not as a special character. For example, if you specify `\` as the escape character, entering \TEST\% in the Value box searches for the string 'TEST%' rather than a string that begins with 'TEST' followed by any other characters. Similarly, entering \TEST\\% searches for strings that begin with 'TEST\%'.

You can specify whether the filter condition is case sensitive or not. A case-sensitive filter is applied only when the data source supports case-sensitive filtering.

**Note:** When a filter based on a pattern is converted into a prompt, users are prompted to specify a pattern when the report is run.

e. If you want the filter to include null values, select the Include missing values (NULL) check box.

f. Click OK.

7. If you create a Combined filter, do the following:
   a. In the Create Filter dialog box, select the data item for the first filter condition and click OK.
   b. In the Filter Conditions dialog box, specify the details for the first condition and click OK.
   c. Click the new condition button and, in the Create Filter dialog box, select the data item for the second filter condition and click OK.
   d. In the Filter Conditions dialog box, specify the details for the second condition and click OK.
   e. In the Combined Filter dialog box, use the AND, OR, and NOT operators to combine the filter conditions. Select the conditions that you want to combine, and click an operator. To change an operator, double-click the operator to toggle among available ones or select the operator and then select a different one from the list.

f. Click OK.

8. If you create an Advanced filter condition, in the Filter Expression dialog box, define the filter expression:
   a. To add data items that are not shown in the report, on the Source tab, double-click data items.
   b. To add data items that are in the report but not necessarily in the model (such as calculations), on the Data Items tab, double-click data items.
   c. To add data items from a specific query in the report, on the Queries tab, double-click data items.
   d. To add functions, summaries, and operators, on the Functions tab, double-click elements. You can insert only functions that return a Boolean value (true or false). For example, you cannot insert the function topCount because it returns a set of data. Filter expressions must resolve to a Boolean in order to be valid.
   e. To add a value that is derived from a parameter, on the Parameters tab, double-click the parameter. Parameters define prompts, drill-through reports, and master detail relationships.
f. To add a macro, on the macros tab, double-click the parameter maps, session parameters, or macro functions you want to add to the macro expression.

   **Tip:** Use **Insert macro block** to get you started with creating the macro expression. **Insert macro block** inserts number signs in the expression. Make sure that the macro expression you create is between the number signs.

g. Click the validate button and click OK.
h. Click OK.

   **Tip:** You can also type the filter expression directly in the **Expression Definition** box. When typing date values, use the YYYY-MM-DD format. For more information about creating expressions, see “Using Relational Calculations” on page 248.

9. In the **Usage** box, specify whether the filter is required, optional, or not to be used.

10. If you create a detail filter, under **Application**, click one of the following options:
   - To apply the filter to individual records in the data source, click **Before auto aggregation**.
     For example, for a specific product type within a product line, you want to filter out individual orders that generated less than one thousand dollars in revenue.
   - To apply the filter to data after the query has grouped and summarized at the lowest level of detail, click **After auto aggregation**.
     For example, you want to filter out the product types that generated less than ten thousand dollars in revenue within a product line.

For example, you have a list that contains the data items Product line, Product type, and Revenue. The revenue values you see are aggregated to the product type level. If you create a detail filter on Revenue and you choose to filter values before auto aggregation, you are filtering non-aggregated revenue values. If you choose to filter values after auto aggregation, you are filtering revenue values aggregated to the product type level.

11. If you create a summary filter, under **Scope**, click the ellipsis (...) button and select the grouping level at which to apply the filter.

   For example, a revenue report is grouped on product line and product type. You can choose to filter total revenue for either the product line or product type level.

   If the query in which you are creating the summary filter is linked to more than one data container, the data item that you select as the grouping level must be grouped in all data containers linked to the query. Otherwise, the report will not run successfully.

   **Tip:** To filter at the overall level, do not click a level. For example, if a report has an overall footer that shows the total revenue for all products, by not choosing a level you will apply the filter to the footer. In addition, if you are using a dimensional data source, excluding a parent level excludes its children, and excluding all children excludes the parent.
Edit or remove a filter

After you created filters, you can edit or remove them.

Procedure
1. Click a data container object that contains a filter.

2. To remove all filters from the object, from the Data menu, click Filters, and click Remove all Filters.

3. To edit a filter or remove a single filter, from the Data menu, click Filters, and click Edit Filters.

Tip: You can also edit filters from Query Explorer. Pause the pointer over the query explorer button and click the query that contains your filter.

Browse or Search the Values of a Data Item

When building expressions in the expression editor, you can browse the data of a data item. This is useful when you do not know how a particular value is stored in the database. For example, you want to filter a report so that it shows data for only New South Wales. The actual value in the database for New South Wales is NSW, and this is what you must insert in the filter expression to obtain the correct results.

Procedure
1. In the source, data items, or query tab, right-click the data item you want to browse.

2. If you want to insert a single value, click Select Value.

3. If you want to insert multiple values, click Select Multiple Values.

4. In the Values box, click the value you want to insert in the expression.

Tip: Use the Keywords box to search for specific values. If you select Use Tab as keyword delimiter, you can add a tab in the Keywords box only by copying and pasting a tab (or the string you want to search on with a tab in it), from another application, such as Notepad. If the data source is case sensitive and you want to perform a case insensitive search, click the Search arrow and then click Case Insensitive.

5. If you clicked multiple values, click the right arrow button to add them to the Selected values box.

6. Click Insert.

Tip: To control where values appear in the Expression Definition box, you can click the copy button if you selected a single value or the copy all button if you selected multiple values instead. The values are copied to the clipboard, and you can then paste them where you want.
In the expression editor, when searching for values for a data item, the results you obtain might contain unexpected results if the data item is not a string data type. Because you can edit the expression for a data item, IBM Cognos Business Intelligence cannot determine with certainty what the data type is.

**Ignoring the Time Component in Date Columns**

Database systems use Date, Time, and Timestamp to represent date and time values. While a timestamp type holds a date and time component, an application may allow the RDBMS to default the time component. When rows are inserted, updated, or queried, the application may specify only a date value and leave the RDBMS to extend the value to include a default time (usually 00:00:00.000).

The challenge with a timestamp is when the application has no immediate interest in the time component. For example, the business question How many orders were taken today? implies all orders taken irrespective of what time the order was booked. If the application defaulted the time component as it stored rows, the query that was used to answer the question returns the count of orders taken today. If the application stored the actual time component, the query likely returns no data, because the number of orders entered at midnight is probably zero.

Relying on the time defaults can be dangerous if the application changes and starts to capture actual times. To avoid this problem, you can:

- [trunc](#) the time by creating a derived column
- convert the timestamp to a date
- create a hi-low filter

**Truncating the Time by Creating a Derived Column**

In the IBM Cognos Framework Manager model, create a derived column using a calculated expression that truncates the time from a timestamp, and returns a timestamp containing the original date and a default time. For example, if you use an Oracle data source, the following expression creates the derived column `DATEONLY` from `COL1`, where `COL1` contains the values as stored by the application while `DATEONLY` contains the dates with the default time of 12:00:00 AM:

```
Select COL1, trunc(COL1) as DATEONLY from [SCOTT_TIGER].DATES
```

**Tip:** You can change the data format of the column to show only the date value by setting the [Date Style](#) property to Short.

You can then apply filters to the `DATEONLY` column that would return the correct results. If you create a parameter in IBM Cognos Report Studio that filters on this column, the default prompt will present a date and time control because the data type is still a timestamp.

**Converting the Timestamp to a Date**

In the IBM Cognos Framework Manager model, define a calculation that uses the cast function to convert the timestamp to a date. For example, the following expression converts the data type of the column `COL1` to date:

```
cast ([SCOTT_TIGER].[DATES].[COL1],DATE)
```
If you create a parameter in IBM Cognos Report Studio that filters on this calculation, the default prompt presents a date control.

**Ignoring the Time by Creating a Hi-Low Filter**

You can create a hi-low filter to ignore the time. In the IBM Cognos Framework Manager model, create a filter in the form date-column between date-lowtime and date-hightime. For example, the following expression returns all values between 00:00:00:000 and 23:59:59:000 for a given day:

```
[SCOTT_TIGER].[DATES].[COL1] between?p1? and cast(substring(?p1?,1,10),'23:59.59.000',timestamp)
```

---

**Sorting Relational Data**

You can sort items to view them in your preferred order. By default, IBM Cognos Report Studio retrieves items in the order defined in the data source. OLAP data sources always have a defined order. Relational and dimensionally-modeled relational data sources may not always have a defined order. The data modeler defines the sorting options in the model.

For more information about data modeling, see the IBM Cognos Framework Manager *User Guide*. Sorting applied in Report Studio applies only to the layout within the report.

You can sort items in a list in ascending or descending order based on a value or a label, such as revenue or employee name. You can also perform advanced sorting to sort columns within groups or to sort a row or column using another data item.

**Tip:** The Sales Growth Year Over Year sample report in the GO Data Warehouse (analysis) package includes sorting. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Procedure**

1. Click the column or row on which to sort.

2. Click the sort button and click Ascending or Descending.

   An arrow appears beside the data item to indicate that a sort order was set. When you specify a sort order for more than one column, the columns are sorted in the order in which they were inserted in the report. For example, you add columns A, B, and C to a report and specify a sort order for each. When you run the report, column A is sorted first, then column B, and then column C. You can change the order in which the columns are sorted in the Edit Layout Sorting options.

   **Tip:** To remove a sort order, click Don't Sort.

**Perform Advanced Layout Sorting**

If you use relational data, in a list, you can sort columns within groups and change the sort order of columns. In a crosstab, you can sort a row or column by another item, such as Order year by Revenue.

**Procedure**

1. Click a column.
In a crosstab, click the sorted row or column on which to perform advanced sorting.

2. Click the sort button and click **Edit Layout Sorting**.

3. To sort a list column within a group, do the following:
   - In the **Groups** pane, under the **Groups** folder, expand the folder of the grouped column.
   - In the **Data Items** pane, drag the data items to sort to the **Sort List** folder.

   **Tip:** You can also drag data items from the **Detail Sort List** folder.

4. To change the sort order of columns, in the **Groups** pane, change the order of columns in the **Sort List** folder of a group or in the **Detail Sort List** folder.

   **Tip:** Add items to the **Detail Sort List** folder to sort items that are not groups.

5. If you are sorting a crosstab, do the following:
   - From the **Data Items** pane, drag the item on which to sort to the **Sort List** pane.
     
     For example, you have a crosstab with Product line as rows, Order year as columns, and Revenue as the measure. To sort Order year by Revenue, drag Revenue to the **Sort List** pane.
   - Click the sort order button to specify ascending or descending order.

---

### Working with Relational Queries

Queries specify what data appears in the report. In IBM Cognos Report Studio, you create and modify queries using **Query Explorer**.

**Query Explorer** provides an alternative way to modify existing reports or to author new reports. You can use **Query Explorer** to perform complex tasks and other tasks that are more difficult to do in the report layout. For example, use **Query Explorer** to

- improve performance by changing the order in which items are queried from the database, or by changing query properties to allow the report server to execute queries concurrently where possible

**Note:** Only queries that are referenced in the report layout can be executed concurrently. If a query is defined in Query Explorer by joining two or more existing queries, IBM Cognos Business Intelligence cannot execute the source queries concurrently. By default, queries run sequentially. Your administrator must enable the concurrent query execution feature. For more information, see the IBM Cognos Business Intelligence **Administration and Security Guide**.

- view or add filters and parameters and modify their properties
- view or add dimensions, levels, and facts
- incorporate SQL statements that come from other reports or reports that you write
- create complex queries using set operations and joins
Queries determine what data items appear in the report. Sometimes you want detailed rows of data, which you obtain by using a simple SELECT statement. Other times you must calculate totals or averages using summary functions and grouped columns or must apply filters to show only the data you want.

**Specify a List of Data Items for an Object**

Specify the list of data items for an object when you must reference a data item that is in a query but not in the layout.

For example, you want to add a layout calculation to a list that uses a data item that is in the query definition. If the data item does not appear in the list, you must reference it in order to make the layout calculation work. You must also specify the list of data items if you apply conditional formatting that uses a data item that is not in the query.

**Procedure**

1. Click a layout object.

   **Tip:** For a list of layout objects for which you can specify a list of properties, see the **Properties** property in Appendix G, “Report Studio Object and Property Reference,” on page 771.

2. In the **Properties** pane, double-click the **Properties** property and select data items.

**Relate a Query to a Layout**

Queries and layouts work together. After you decide the type of data that you need, you must create a layout in which to show the results. Each column of data must be both selected for the query and shown in a layout unless there are some columns that you do not want to show. The query and layout portions of a report must be linked to produce a valid report.

IBM Cognos Report Studio automatically links query and layout. For example, when you use Report Studio and the list report layout, query and layout are automatically linked.

**Procedure**

1. Select a data container.

2. In the **Properties** pane, set the **Query** property to a query.

3. From the **Data Items** tab, drag data items from the query to the data container.

**Connecting Queries Between Different Data Sources**

This section lists the data sources between which you can and cannot connect queries.

IBM Cognos Business Intelligence supports the following:

- RDBMS to RDBMS joins
- set operations of any two queries
- master detail relationships between any two queries
• drill from any query to any other query

You cannot create the following types of joins:
• cube-to-cube (homogeneous)
• cube-to-cube (heterogeneous)
• cube-to-RDBMS
• cube-to-SAP BW
• SAP-BW-to-RDBMS

Add a Query to a Relational Report
You can create multiple queries in Query Explorer to suit your particular needs. For example, you can create a separate query for each data container in a report to show different data.

Tip: The Briefing Book sample report in the GO Sales (analysis) package and the Top 10 Retailers for 2011 sample report in the GO Data Warehouse (analysis) package include multiple queries. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Procedure
1. Pause the pointer over the query explorer button and click Queries.
2. From the Toolbox tab, drag one of the following objects to the work area.

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>Adds a query.</td>
</tr>
<tr>
<td>Join</td>
<td>Adds a join relationship</td>
</tr>
<tr>
<td>Union</td>
<td>Adds a union operator</td>
</tr>
<tr>
<td>Intersect</td>
<td>Adds an intersect operator</td>
</tr>
<tr>
<td>Except</td>
<td>Adds an except (minus) operator</td>
</tr>
<tr>
<td>SQL</td>
<td>Adds SQL commands.</td>
</tr>
</tbody>
</table>

Note: When adding queries to the report
• right-click the work area and click Show Package Sources to see the queries that use data items from a package
• right-click the work area and click Expand References to see the relationships that exist between queries in the report, which is useful when you are creating complex queries
3. In the Properties pane, set the object properties.
   For example, if you added a join, set the Join Relationships property to define the join.
4. Double-click a query.
5. From the Source tab, drag data items to the Data Items pane.

Tip: You can add data items to the query that you do not want to appear in the layout. For example, to filter on Product line code and show Product line in the layout, you must add both data items to the query.
6. To create a new data item, from the Toolbox tab, drag Data Item to the Data Item pane.
7. To add a filter, from the Toolbox tab, drag Filter to the Detail Filters or Summary Filters pane and define the filter expression.

Tip: You can also create a filter by dragging a data item from the Source tab to one of the filters panes and completing the filter expression.

Create a Union Query

Create a union query to combine two or more queries into one result set.

You can combine queries that use different data sources. For example, you can combine a query that returns data from a dimensional data source with a query that returns data from a relational data source.

Before you begin

To combine two queries, the following conditions must be met:

• The two queries must have the same number of data items.
• The data items must have compatible types and appear in the same order.
  For numeric data types, integer, float, double, and decimal are compatible.
  For string data types, char, varchar, and longvarchar are compatible.
  For binary data types, binary and varbinary are compatible.
  Date data types must match exactly.

Procedure

1. Pause the pointer over the query explorer button and click Queries.

2. From the Toolbox tab, do the following:
   • Drag Query to the work area.
   • Drag Union, Intersect, or Except to the right of the query.
     Two drop zones appear to the right of the operator.
   • Drag a Query object to each drop zone.
     Two queries are created in the work area, and a shortcut to each query appears in the drop zones.

3. Double-click each query that makes up the union query and add data items to the query.

4. Return to the Queries work area.

5. Click the set operator that you added in step 2.

6. In the Properties pane, set the Duplicates property to remove or preserve duplicate rows.

7. Double-click the Projection List property.
   The projection list shows the list of projected data items for the set operation.

8. To automatically produce the list of projected data items, click Automatically generated.
   IBM Cognos Report Studio generates the projection list using only one of the two queries in the union.
9. To add, delete, move, or rename data items in the projection list, click Manual and make the changes.

10. Double-click the union query.

11. From the Source tab, drag data items to the Data Items pane.

Results
The union query is complete. You can now link the union query to a data container in the layout.

Example - Create a Two-column List Report for Three Data Items
You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a list report that shows revenue for all product lines and order methods. However, you want the product lines and order methods to appear in a single column. To create this report, you use a union query to join the Product line and Order method data items.

Procedure
1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click Blank.
4. Pause the pointer over the query explorer button and click Queries.
5. From the Toolbox tab, do the following:
   • Drag Query to the work area.
   • Drag Union to the right of the query.
     Two drop zones appear to the right of the operator.
   • Drag a Query object to each drop zone.
     Query2 and Query3 are created in the work area, and a shortcut to each query appears in the drop zones.
7. From the Source tab, open Sales and Marketing (query), then open Sales (query). Then drag the following data items to the Data Items pane:
   • from the Products folder, drag Product line
   • from the Sales fact folder, drag Revenue
8. From the Toolbox tab, drag Data Item to the Data Items pane.
   The data item will be used to sort product lines and order methods in the report.
9. In the Expression Definition box, type 'A' and click OK.
10. In the Properties pane, set the Name property to Sort key.
11. On the toolbar, press the back button to return to the Queries work area.
13. From the **Source** tab, drag the following data items to the **Data Items** pane:
   - from the **Order method** folder, drag **Order method type**
   - from the **Sales fact** folder, drag **Revenue**

14. Repeat steps 8 to 10 to create a Sort key data item in Query3 with 'B' as its definition.

15. On the toolbar, press the back button to return to the Queries work area.
16. Click the **Union** operator.
17. Double-click the **Projection List** property.
   The **Product line** item in the projection list contains both product lines and order methods.
18. Click **Manual**.
19. Click **Product line**, and then click the edit button.
20. In the Edit box, type the following after **Product line**:
      & **Order method**
21. Click **OK** twice.
22. Double-click **Query1**.

23. From the **Source** tab, drag the following data items to the **Data Items** pane:
   - **Product line & Order method**
   - **Revenue**
   - **Sort key**

24. Select the **Sort key** data item and, in the **Properties** pane, set the Pre-Sort property to **Sort ascending**.
   When you run the report, all product lines will appear first followed by all order methods.

25. Pause the pointer over the page explorer button and click **Page1**.
26. From the **Toolbox** tab, drag **List** to the work area.
27. Click the list.
28. Click the select ancestor button in the title bar of the **Properties** pane and click **List**.
   **Tip:** You can also click the container selector (three orange dots) of the list to select it.
29. In the **Properties** pane, set the **Query** property to **Query1**.
   The list is linked to the union query.
30. Double-click the **Properties** property.
31. Select the **Sort key** check box and click **OK**.
   Since the Sort key data item does not appear in the list, you must make it a property of the list before it can sort product lines and order methods.

32. From the **Data Items** tab, drag the following items from Query1 to the list:
• Product line & Order method
• Revenue

33. Run the report.

Results

A list report with two columns is produced. All product lines and order methods appear in the first column.

<table>
<thead>
<tr>
<th>Product line &amp; Order method</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Equipment</td>
<td>1,589,036,664.03</td>
</tr>
<tr>
<td>Golf Equipment</td>
<td>726,411,357.89</td>
</tr>
<tr>
<td>Mountaineering Equipment</td>
<td>409,660,132.9</td>
</tr>
<tr>
<td>Outdoor Protection</td>
<td>75,994,296.25</td>
</tr>
<tr>
<td>Personal Accessories</td>
<td>1,885,673,307.76</td>
</tr>
<tr>
<td>E-mail</td>
<td>179,043,044.16</td>
</tr>
<tr>
<td>Fax</td>
<td>70,073,542.01</td>
</tr>
<tr>
<td>Mail</td>
<td>40,091,338.97</td>
</tr>
<tr>
<td>Sales visit</td>
<td>310,104,834</td>
</tr>
<tr>
<td>Special</td>
<td>27,361,320.25</td>
</tr>
<tr>
<td>Telephone</td>
<td>340,985,781.06</td>
</tr>
<tr>
<td>Web</td>
<td>3,712,235,908.4</td>
</tr>
</tbody>
</table>

Create a Join Relationship

You can create a join relationship to join two queries.

In general, join relationships should be created in the IBM Cognos Framework Manager model. Create a join relationship in IBM Cognos Report Studio if what you are trying to do cannot be modeled in Framework Manager.

Procedure

1. Pause the pointer over the query explorer button and click Queries.

2. From the Toolbox tab, do the following:
   • Drag Query to the work area.
   • Drag Join to the right of the query.
     Two drop zones appear to the right of the operator.
   • Drag a Query object to each drop zone.
     Two queries are created in the work area, and a shortcut to each query appears in the drop zones.

3. Double-click each query that makes up the join query and add data items to the query.

4. On the toolbar, press the back button to return to the Queries work area.

5. Click Join.
6. In the Properties pane, double-click the Join Relationships property.
7. Click New Link.
8. To create the link, click a data item in the left query, and then click a data item in the right query.
9. For each query, click Cardinality and click a cardinality option.
10. Click Operator and click an operator.
11. To convert the join relationship to an expression, click Convert to expression.
   Convert the join relationship to an expression to make changes to the join definition.

   **Note:** After you convert the relationship to an expression, you cannot change it back to a relationship.
12. Double-click the join query.
13. From the Source tab, drag data items to the Data Items pane.

**Results**

The join query is complete. You can now link the join query to a data container in the layout.

**Reference a Package Item in a Child Query**

When you create a child query in IBM Cognos Report Studio, you can only reference items from its parent or from other queries. For example, if you add a filter to a child query, the only items that you can insert into the expression are items that exist in other queries defined in the report. To add an item from the package, you must unlink the child query from its parent.

**Procedure**

1. Pause the pointer over the query explorer button and click Queries.
2. Unlink the child query from its parent by selecting the parent query shortcut to the right of the child query and clicking the delete button.
3. Double-click the child query.
   Package items are now available on the Source tab.
4. Add package items.
   For example, to create a detail filter that references a package item, drag the filter object from the Toolbox tab to the Detail Filters pane, and then add the items in the Expression Definition box.
5. Click the up arrow button in the toolbar to return to the queries work area.
6. Recreate the link between the child query and the parent query by dragging the parent query to the right of the child query.
7. If necessary, double-click the child query to complete it.
   For example, if you created a filter, you may need to complete the filter expression.
Create a Master Detail Relationship

Create a master detail relationship to deliver information that would otherwise require two or more reports. For example, you can combine a list with a chart. The list can contain product lines and the chart can show details for each product line.

Master detail relationships must appear in nested frames to produce the correct results. You can create a master detail relationship in two ways:

- Use a parent frame for the master query and a nested frame for the detail query.
- Associate a report page with the master query and use a data container, such as a list or crosstab, for the detail query.

You cannot display parent data in the child frame or child data in the parent frame. Also, you cannot perform calculations across master detail queries.

You can use a master detail relationship to show data from separate data sources in a single report. However, the data sources must be contained in the same package.

If you are working with an SAP BW data source, you cannot use a data item from the master query that contains non-ASCII values.

Before you begin

To create a master detail relationship using queries that reference two different dimensional data sources, you must first link the members by aliasing the levels that contain the members.

Procedure

1. To use a parent frame for the master query and a nested frame for the detail query:
   - From the Toolbox tab, drag a List, Repeater Table, or Repeater to the report.
   - Add a second data container to the object you inserted.
     You can insert a list, crosstab, chart, repeater table, or repeater into a list.
     You can add a list to a repeater table or repeater.
   - Add data items to both data containers.

2. To associate a report page with the master query and use a data container for the detail query:
   - Click anywhere in the report page.
   - In the Properties pane, click the select ancestor button and click Page.
   - Set the Query property.
   - From the Toolbox tab, drag a data container to the report.

3. To link a data item in the master query to a parameter in the detail query instead of to another data item, create the parameter in the detail query.
   Use parameters to filter values at a lower level in the detail query.
   - Pause the pointer over the query explorer button and click the detail query.
   - From the Toolbox tab, drag the Filter object to the Detail Filters box.
   - In the Expression Definition box, create the parameter.
4. Pause the pointer over the page explorer button and click the report page.
5. Click anywhere in the report page.
6. In the Properties pane, click the select ancestor button and click Page.
7. Click the data container containing the details.
8. From the Data menu, click Master Detail Relationships.
9. Click New Link.
10. In the Master box, click the data item that will provide the primary information.
11. To link the master query to a data item, in the Detail box, click the data item that will provide the detailed information.
12. To link the master query to a parameter, in the Parameters box, click the parameter that will provide the detailed information.

**Tip:** To delete a link, select the link and press the Delete key.

13. If your detail query object is a chart, you can disconnect a chart title from the master query.
   • Click the chart title, and then, from the Data menu, click Master Detail Relationships.
   • Clear the Use the master detail relationships from the chart check box.
14. If your master detail relationship includes a chart as the detail and you are using automatic axis ranges, you can set the same axis range for all the detail charts:
   • Select the axis.
   • In the Properties pane, under General, double-click the Axis Range property.
   • Under Maximum and Minimum, click Automatic.
      The master detail report option is only available with automatic axis ranges.
   • Under Master detail reports, select the Same range for all instances of chart check box.

For more information about axis ranges, see “Change the Axis Scale of a Current Default Chart” on page 120.

**Tip:** To avoid seeing the same data item twice in the report, click the data item in the data container driven by the detail query and click the cut button. This removes the item from the report display but keeps it in the query.

**Master Detail or Burst Reports with Charts or Crosstabs May Result in Denial of Service**

When running a master detail or burst report that includes a chart or crosstab, disk space exhaustion may cause the report or other requests to fail. A large set of burst keys or master rows may produce one or more charts per detail, resulting in many master detail executions. This may cause the temp folder to accumulate many gigabytes of temporary files containing data required for successful chart rendering.

To avoid this issue, we recommend that you test large master detail or burst reports that include charts or crosstabs to determine the potential peak disk requirements for the report.
Working with Queries in SQL

For each query in a report, you can work with the SQL that is executed when you run a report.

You can

- view the SQL for an entire report or a query
- build a report using your own SQL
- convert a report to use SQL directly
- edit the SQL

When working with a relational data source, the SQL that is produced by IBM Cognos Report Studio depends on the report output selected.

Note: MDX is not available with relational or DMR data sources.

View the SQL for an Entire Report or a Query

View the SQL to see what is passed to the database when you run a report.

Procedure

1. To view the SQL for the entire report, from the Tools menu, click Show Generated SQL/MDX.
   This option shows the SQL that will be executed in the data source. The SQL is organized by query and by query result. If a query is used in more than one data container, a query result is generated for each data container.
2. To view the SQL for a specific query, do the following:
   - Pause the pointer over the query explorer button and click the query.
   - In the Properties pane, double-click the Generated SQL/MDX property.
   The Generated SQL/MDX property shows the SQL that is executed when you view tabular data (from the Run menu, click View Tabular Data). Tabular data shows the data that is produced by the query in the form of a list. You can use this property to help you build advanced queries.

Results

The SQL for the query appears in the Generated SQL/MDX dialog box. You can choose to view native SQL, which is the SQL that is passed to the database when you execute the query, or Cognos SQL, which is a generic form of SQL that IBM Cognos Report Studio uses. Cognos SQL is converted to native SQL before the query is executed.

Related tasks:

"Showing prompt values in generated SQL or MDX" on page 326
You can show prompt values when viewing the generated SQL or MDX of a query or report.

Build a Report Using Your Own SQL

You can build a report by adding SQL from an external source, such as another report.

If you edit the SQL of a query, you must change the Processing property for that query to Limited Local.
Procedure

1. From the **File** menu, click **New**.
2. Click **Blank**.
3. Pause the pointer over the query explorer button and click **Queries**.
4. From the **Toolbox** tab, drag **Query** to the work area.
5. From the **Toolbox** tab, to build an SQL query, drag **SQL** to the right of the query.

   **Tip**: You can drag **SQL** anywhere in the work area, and IBM Cognos Report Studio will automatically create a query.
6. In the **Properties** pane, double-click the **Data Source** property and click a data source.
7. If required, set the **Catalog** property to the name of the catalog.
8. Double-click the **SQL** property and type the SQL.
9. Click **Validate** to check for errors.
10. In the **Properties** pane, set the **Processing** property to **Limited Local**.
11. Double-click the query.
    
    If the SQL is valid, the data items defined in the SQL appear in the **Data Items** pane.
12. Pause the pointer over the page explorer button and click a report page.
13. From the **Toolbox** tab, drag an object to the work area.
    
    For example, drag a list, crosstab, chart, or repeater.
14. Click the data container.
15. In the **Properties** pane, click the select ancestor button and click the container you just created.
    
    For example, if you created a list, click **List**.
    
    **Tip**: You can also click the container selector (three orange dots) of the container to select it.
16. Set the **Query** property to the query for the report type.
17. From the **Data Items** tab, drag the items from the query you chose in the previous step to the data container.

**Convert a Query to SQL**

You can convert a query to SQL to edit it. You may want to do this to improve performance or to use SQL features that are not supported directly by IBM Cognos Report Studio.

Converting a query to SQL is an irreversible process.

**Procedure**

1. Pause the pointer over the query explorer button and click the query.
2. In the **Properties** pane, double-click the **Generated SQL** property.
3. Click **Convert**.
4. Click **Validate** to check for errors.
5. In the Properties pane, set the Processing property to Limited Local.

**Edit the SQL**
You can edit the SQL for a query that has been created as an SQL query or that has been converted to SQL.

**Procedure**

1. Pause the pointer over the query explorer button and click the SQL item under the query.
2. Double-click the SQL item.
3. Make changes in the text box.
4. Click Validate to check for errors.

**First-Rows Optimization**
The SQL produced by IBM Cognos Report Studio depends on the report format you choose. For example, if you specify HTML format, first-rows optimization is requested. All-rows is requested if you specify PDF.

It is important for database administrators and programmers to remember that Report Studio does not always use first-rows optimization. If you assume first-rows optimization is always requested, this can cause the RDBMS optimizer to process the query differently than you expect.

**Using Relational Calculations**
Insert a calculation to make your report more meaningful by deriving additional information from the data source. For example, you create an invoice report, and you want to see the total sale amount for each product ordered. Create a calculated column that multiplies the product price by the quantity ordered.

You build calculations in the expression editor using functions.

If a calculation is used in multiple reports, or by different report authors, ask your modeler to create the calculation as a standalone object in the model and include it in the relevant package.

You can add calculations to lists, crosstabs, and all other data containers. You can also add calculations directly to a page header, body, or footer. However, you must first associate a query to the page. For more information, see "Add a Page to a Report" on page 437.

**Units of Measure**
When creating calculations in IBM Cognos Report Studio and IBM Cognos Query Studio, you may encounter problems with the units of measure. For example, the calculation Cost*Cost returns the unit of measure * instead of a currency unit of measure. To avoid this problem, change the format of the corresponding column to obtain the desired unit of measure.

**Calculation Solve Order**
When calculations in the rows and columns of a report intersect, calculations are performed in the following order: addition or subtraction, multiplication or division, aggregation (rollup), and then the remaining arithmetic functions.
The remaining functions are as follows:
• absolute, round, round down, average, minimum, maximum, medium, count
• percentage, % difference (growth) or % of total
• rank, quartile, quantile, or percentile

If both calculations have the same precedence, for example, if they are both business functions, then the row calculation takes precedence.

Limitations of Calculations

You should use only the expressions and functions available in Report Studio, and follow their syntax.

Minimal checking exists for calculations. If your calculation uses an invalid expression, your report results may contain unexpected values.

In addition, you should define member summaries as follows:

\[ \text{summary\_function (currentMeasure within set set\_reference)} \]

where \( set\_reference \) is a level or set inserted from the Source tab.

Unless otherwise required, \( \text{summary\_function} \) should be the \( \text{aggregate} \) function. If you use an explicit summary function, you may encounter problems with measures and scenario or account dimension members (such as profit margin, distinct count, and so on) that have complex rollup rules, or members that do not roll up.

Know your data, and confirm with the owner of the cube where overriding the automatic aggregation is safe.

Because of the above limitations, summaries of calculations may not provide reliable values. For convenience, you may have to build reports where row summaries and calculated member columns intersect. In such reports, these intersections may contain unexpected values. In contrast, row calculations intersecting with column aggregates using the \( \text{aggregate} \) function are safe because the calculation is performed on the reliably summarized values.

Create a Simple Calculation

You can select items in your report and create simple calculations.

In addition to simple arithmetic calculations, you can perform the following calculations:

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rollup</td>
<td>Summarizes, or rolls up, all the values in a hierarchy.</td>
</tr>
<tr>
<td>%</td>
<td>Calculates the value of a selected item as a percentage of another item.</td>
</tr>
</tbody>
</table>
### Calculation Description

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Growth</td>
<td>Calculates the difference between two items as a percentage.</td>
</tr>
<tr>
<td>% Of Base</td>
<td>This calculation is only available if you select two members from different hierarchies, each from a different edge. This calculation takes the first selected member from edge A and the second selected member from edge B. The calculation result of a percent of base should be such that all the values for member A will compare itself to the intersect value of member A and B.</td>
</tr>
<tr>
<td>Custom</td>
<td>Allows you to specify your own value when performing a simple arithmetic calculation. Also allows you to change the order of operands or type a custom name for the new calculated row or column.</td>
</tr>
</tbody>
</table>

### Procedure

1. Select the items in your report to calculate.
2. Click the insert calculation button and select the calculation to perform.
   - **Tip:** Calculations that are not applicable to the items you selected are greyed out.
3. To change the order of the operands, the name of the calculated item added to the report, or to create a custom calculation, click **Custom**.

### Results

The calculation appears as a new row or column in your report.

### Insert a Query Calculation

Insert a query calculation into your report to add a new row or column with values that are based on a calculation.

For example, you create a query calculation named Euros that converts dollars to euros by multiplying an existing dollar measure by a conversion rate. Euros can then be displayed to end users in a separate row or column.

**Note:** When creating an expression that will be used in a double-byte environment, such as Japanese, the only special characters that will work are ASCII-7 and ~ -- ||-$¢£¬.

Not all data sources support functions the same way. The data modeler can set a quality of service indicator on functions to give a visual clue about the behavior of the functions. Report authors can use the quality of service indicators to determine which functions to use in a report. The quality of service indicators are:

- not available (X)
The function is not available for any data source in the package.
- limited availability (!!)
  The function is not available for some data sources in the package.
- limited support (!)
  The function is available for all data sources in the package but is not naturally supported for that data source. IBM Cognos Business Intelligence uses a local approximation for that function. Because an approximation is used, performance can be poor and the results may not be what you expect.
- unconstrained (check mark)
  The function is available for all data sources.

When inserting literal strings in an expression, you must enclose the string in single quotation marks. If the string contains a quotation mark, it must be escaped. For example, if you want to insert the string ab’c, you must type ‘ab”c’.

**Procedure**

1. From the Toolbox tab, drag Query Calculation to the report.
2. Type a meaningful name for your expression in the Name box.
   For example, if you are calculating the difference between 2012 revenue and 2011 revenue, you could name your expression **2012 - 2011 Revenue**.
3. In the Available Components pane, define the calculation:
   - To add data items that are not shown in the report, on the source tab, double-click the data items.
   - To add data items that are in the report but not necessarily in the model, such as calculations, on the data items tab, double-click the data items.
   - To add data items from a specific query, on the queries tab, double-click data items.
   - To add functions, summaries, and operators, on the functions tab, double-click elements.
   - To add a value that is derived from a parameter, on the parameters tab, double-click a parameter.
     Parameters define prompts, drill-through reports, and master detail relationships.
   - To add a macro, on the macros tab, double-click the parameter maps, session parameters, or macro functions you want to add to the macro expression.
   
     **Tip:** Use Insert macro block to get you started with creating the macro expression. Insert macro block inserts number signs in the expression. Make sure that the macro expression you create is between the number signs.

**Tips:**
- You can also type the calculation directly in the Expression Definition box.
- When typing date values, ensure that the date format is correct for your database type.
• To copy and paste expression components in the Expression Definition pane, you can use the copy button and the paste button.

For information about specific functions, see Appendix F, “Using the expression editor,” on page 555.

4. Click the validate button.


Related concepts:

Chapter 17, “Using Query Macros,” on page 403

A macro is a fragment of code that you can insert in the Select statement of a query or in an expression. For example, add a macro to insert a new data item containing the user’s name.

Create a Layout Calculation

Create a layout calculation to add run-time information to your report, such as current date, current time, and user name.

When creating layout calculations, only report functions are available to you in the expression editor. These functions are executed in the report layout, as opposed to the other functions which are executed in the query.

Procedure

1. From the Toolbox tab, drag Layout Calculation to the report.
2. In the Available Components box, define the calculation:

   • To add data items that are not shown in the report, on the source tab, double-click data items.

   • To add data items from a specific query, on the queries tab, double-click data items.

   • To add functions, summaries, and operators, on the functions tab, double-click elements.

   • To add a value that is derived from a parameter, on the parameters tab, double-click a parameter.

     Parameters define prompts, drill-through reports, and master detail relationships.

     Tip: You can also type the expression directly in the Expression Definition box.

For more information about creating expressions, see Appendix F, “Using the expression editor,” on page 555.

3. Click the validate button.
Chapter 10. Dimensional Reporting Style

The dimensional reporting style is recommended for dimensionally-modeled relational (DMR) and Online Analytical Processing (OLAP) data sources. Dimensional data is best represented by crosstabs, maps, and charts. This data is shown in IBM Cognos Report Studio in dimensions, hierarchies, levels, and members.

In dimensional reporting, you summarize data by using member summaries and within set aggregation. You focus data in dimensional reporting by adding only the relevant members to the edge of a crosstab or to the context filter. You can also enable drilling up and drilling down in dimensional reports.

IBM Cognos Analysis Studio and IBM Cognos Workspace Advanced are also available to work with dimensional data.

Add Dimensional Data to a Report

For dimensional and mixed model data sources, you can view the full data tree by clicking the view package tree button in the Source tab. You can switch to the dimensional-only data tree by clicking the view members tree button.

Note: We recommend using dimensional data in the dimensional reporting style. However, if you are using relational data, see “Add Relational Data to a Report” on page 213.

If you are working with a dimensional data source, data items are organized hierarchically. Dimensional data sources include OLAP and dimensionally-modeled relational (DMR) data sources. The Source tab shows a metadata-based view of the data.
Note: The names of levels and members in a dimension come from the model. It is the responsibility of the modeler to provide meaningful names.

1. Package
   Packages are subsets of a model containing items that you can insert in a report.

2. Dimension
   Dimensions are broad groupings of descriptive data about a major aspect of a business, such as products, dates, or markets.

3. Level hierarchy
   Level hierarchies are more specific groupings within a dimension. For example, for the Years dimension, data can be organized into smaller groups, such as Years, Current Month, and Last Month.

4. Members folder
   Members folders contain the available members for a hierarchy or level. For example, the Members folder for the Years level hierarchy contains everything found in the Year, Quarter, and Month levels.

5. Level
   Levels are positions within the dimensional hierarchy that contain information at the same order of detail and have attributes in common. Multiple levels can exist within a level hierarchy, beginning with a root level. For example, the Years level hierarchy has the following related levels.

<table>
<thead>
<tr>
<th>Level</th>
<th>Level name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>Years</td>
<td>The root level.</td>
</tr>
<tr>
<td>First</td>
<td>Year</td>
<td>Years in the Years root level. For example, 2013, 2012, and 2011.</td>
</tr>
<tr>
<td>Second</td>
<td>Quarter</td>
<td>Quarters for each year in the Year level. For example, 2013 Q1, 2013 Q2, and 2013 Q3.</td>
</tr>
<tr>
<td>Level</td>
<td>Level name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Third</td>
<td>Month</td>
<td>Months for each quarter in the Quarter level. For example, Jan., Feb., and Mar.</td>
</tr>
</tbody>
</table>

**Tip**: The Measures dimension contains the measures available in the data source.

6. Member property

Member properties are attributes that each member possesses. For example, gender could be a property for all employee members. For more information about member properties, see “Insert a Member Property” on page 258.

**Related concepts**:

“Limited Support for Relational Functions When Used with OLAP Data Sources” on page 533

When working with an OLAP data source, we recommend that you not use relational functions, such as substring and concatenation functions, in a report that also contains a measure with the Aggregate Function property set to Calculated or Automatic in the model. If you do so, you may encounter unexpected results. For example, some summaries are calculated using the Minimum function instead of the aggregate function derived from the individual query items.

**Customize the Source Tree**

When using dimensional data, you can customize how you see data in the source tree. You can select a full package tree view that shows dimensions, hierarchies, levels, an optional members folder, and member properties.

Or you can select a members-only tree view, such as the one in IBM Cognos Analysis Studio. This view allows you to add only members in your report.

**Procedure**

1. Click the Source tab.

2. To view the members-only tree view, click the view members tree button.

3. To view the full package tree, click the view package tree button.

4. To change what is included in the tree, right-click and click Package Tree Settings.

   Use the Preview pane to choose the appropriate settings.

**Insert a Member**

By default, when you insert members from the source tree into your report with IBM Cognos Report Studio, single members are inserted without any details or children. You can change how members are inserted. For example, you may want to insert a member along with all its children or insert only the member’s children. Or you may want to insert members as a set.

When you double-click a member that you have already inserted in your report, by default, its children are inserted after the parent member. You can control whether the children are inserted before or after, nested, or not inserted at all: from the Tools menu, click Options, and then choose a setting on the Edit tab.
When inserting members into a crosstab, ensure that you insert members from the same hierarchy on only one of the crosstab edges. If you insert members from the same hierarchy on both edges of the crosstab, you may encounter unexpected results. For example, a report that uses members from years in the rows and Quarters in the columns is very difficult to read because the useful numbers are distributed over a large area of mostly blank cells.

Calculated members from the data source also appear in the source tree. However, calculated members in Microsoft SQL Server Analysis Services (SSAS) cubes do not appear in the list of descendants in the query. If you want these calculated members to appear in a report or a prompt, you must insert them explicitly.

**Procedure**

1. From the **Source** tab, click the insert single member button and select how to insert members.
2. If you want to insert members within a set instead of inserting individual members, click the create sets for members button.
3. Do one of the following:
   - To insert a member above or below another member, drag the new member above or below a cell. A flashing black bar appears where you can drop the new member.
   - To replace a member, drag the new member to the cell with the member to replace. A flashing black box appears where you can drop the new member.

**Search for a Member**

You can perform a member search to quickly find data.

You can control the number of members that are returned in a search by specifying a value for the **Member display count limit (in source tree)** option (Tools, Options, Advanced tab).

**Procedure**

1. From the **Source** tab, select and then right-click a hierarchy or level and click **Search**.
2. In the **Words** box, type the words or characters for which to search.
3. Click the search parameter to use.
4. To search all descendants instead of just the immediate children, select the **Search all descendants** check box. For example, when performing a search in a hierarchy, selecting this check box returns members found in all the levels of the hierarchy.
5. Click **Search**.

**Results**

The search results appear in a hierarchical structure on the **Search** tab. You can browse the hierarchy to explore members at lower levels.
Tip: You can insert members directly from the Search tab into a report. This can save you time, because you do not have to define a filter. For example, instead of inserting Product line from the Source tab and adding a filter for Camping Equipment, you can insert Camping Equipment from the Search tab.

Nest Members
When you insert members in your report, you may want to nest them in another row or column to make your report easier to use. You can nest members from different dimensions. You can also nest sets.

For example, in the following report, the quarters (Q1 to Q4) were selected from the Time dimension and nested only for the GO Americas member, which is from a different dimension.

![Figure 40. Crosstab showing the year to date amount for all four quarters nested beside GO Americas](image)

When you filter out children from nested sets, the parent set still appears in your report. To avoid this, filter only the top level set and nest only the complete set of descendants at the desired levels.

Procedure

From the Source tab, drag the members to the work area. A flashing black bar indicates where you can drop an item.

Insert a Hierarchy
You can quickly insert entire hierarchies in a report.

When using an IBM Cognos PowerCube, a SSAS cube, or a dimensionally-modeled relational data source, you can insert multiple hierarchies from the same dimension in a crosstab. For example, you can place one hierarchy from a dimension on an edge of the crosstab and nest another hierarchy from the same dimension on the same edge, on another edge, or in the Context filter area.

You can also perform arithmetic, percentage, and ranking calculations using multiple hierarchies.

Summary or analytic calculations that refer to different hierarchies from the same dimension, such as total, count, average, minimum, and maximum, are not supported. These calculations cause an OP-ERR-0250 error.
Procedure

1. From the **Source** tab, drag the hierarchy to the report.

2. In the **Insert Hierarchy** dialog box, choose which members to insert:
   - To insert only the root members of the hierarchy, click **Root members**.
   - To insert all members of the hierarchy, click **All members**.

   **Tip:** The default **Level Indentation** property for all members is to show the hierarchical structure when the report is run.

Insert a Member Property

You can insert member properties, which are attributes of a member, in your report. Member properties provide additional information about the member. For example, an employee level may have a property named gender.

In the data tree of the **Source** tab, member properties are identified by the icon.

You can only insert properties. You cannot group them.

Procedure

From the **Source** tab, drag the member property to the report.

Create a Set of Members

Use sets to group members that are logically related for various actions, such as formatting, nesting, and sorting. Creating sets is also useful when members may be dynamic over time. For example, the child accounts of a total assets account may change from year to year. By creating a set, you do not have to modify the report each time accounts are added or removed.

After you create a set of members, you can add or remove members within the set.

You can perform operations on the set such as excluding specific members, moving members within the set, showing only the top or bottom members, filtering the set, and expanding or collapsing the members within a set. You can view the set definition to see and edit operations that are performed on the set.

Procedure

1. From the **Source** tab, click the create sets for members button to toggle between adding individual members and creating sets for members.

2. From the **Source** tab, select the items to include in the set and drag them to the work area.

3. To add or remove members from the set, right-click the set and click **Edit Members**.

   **Tip:** You can also select the set and, in the **Properties** pane, double-click the **Members** property.
Sharing sets between reports

When you are working with dimensional data, you can share a set to make it available for inclusion in multiple reports.

You must first create a set in a source crosstab report, create a shared set definition, then refer to it in a target report.

A shared set definition is copied from one report to another. As a result, the set definition must not depend on any context from the source report. For example, it must meet the following criteria:

- It cannot reference other shared sets.
- It cannot reference other data items.
- It cannot contain calculated members.

You cannot change the definition of a shared set from within the target report. However, you can carry out other set operations on it, such as excluding a member or creating a top 10 list.

Procedure

1. In the source crosstab report, create a set that you want to share.
2. Select the set. In the Properties pane, in the Sharing property, enter a name and description for the set.
3. Save the report.
4. In the Source tab, right-click the model. Click Add Shared Set Report.
5. In the Open box, select the source report that contains the shared set definition. If it does not exist, a new folder that is called Shared Sets is created. It is displayed at the bottom of the source tree.
6. Optional: To add or delete other reports in the Shared Sets folder, right-click it and click Add Report or Delete Report.
7. Create or open a crosstab target report that you want to refer to the shared set definition.
8. In the Source tab, open the Shared Sets folder. Open the source report that contains the shared set definition that you want to use.
9. Add the shared set definition into your target report.
10. Right-click the shared set in the target report and click Edit Set.
11. In the Set Definition box, select the shared set.
13. Set the Inclusion property.
   a. Select Run time to retrieve the latest shared set definition every time that the report runs.
   b. Select Design time to store the shared set definition in the target report. This definition is used until you refresh it.
14. Click OK, then save the report.

Managing shared set references

After you share a set to make it available for inclusion in multiple reports, you can manage references to it. You can also copy it locally.
You can manage set references in the following ways:

- Create a copy of a shared set to be used in a report.

  Converting a shared set reference into a copy replaces the reference with a set definition that is stored in the report. The report does not reference the shared set from the source report. Any changes that are made to the shared set in the source report are not reflected in the target report. Instead of converting the reference to a copy, you can change the Inclusion property of the reference to Design time. Design time references use a static stored version of the shared set definition. However, you can later change the Inclusion property to be Run time to use the latest shared set definition.

- Modify the properties of a set reference in the target report, including its Inclusion property. You can also modify a set reference in the target report by changing the set definition it refers to.

- Refresh an existing set reference after a set changed in a source report and you want to use the updated set. To use the updated set if the Inclusion property is set to Design time, you must refresh its reference.

- Review problems with an existing set reference.

**Procedure**

1. Open the target report that contains the reference to the shared set that you want to work with.
2. Click Tools > Manage Shared Set References.
3. To create a copy of a shared set to be used in a report, do the following steps:
   a. Select the shared set reference that you want to copy.
   b. Click Convert Reference to Copy. The Confirm Convert window is displayed.
   c. If you want to convert the set reference into a copy that is stored in this report, click Yes. If you do not want to create a copy, click No.

   **Important:** If you convert the set reference into a copy, it is not updated when the set in the original report changes.

4. To modify an existing set reference in a target report, do the following steps:
   a. Select the shared set reference that you want to modify.
   b. Click Edit Reference.
   c. To modify the set reference properties, select the reference, then click Edit. The Set Reference Properties window opens. You can now modify the set reference properties.
   d. To modify the definition of the set, click New. The Set Definition window opens. You can now modify the set definition.

5. To refresh an existing set reference after a set changed in a source report, do the following steps:
   a. Select the set reference that you want to refresh.
   b. Click Refresh Reference.

6. To review problems with an existing set reference in a target report, do the following steps:
   a. Select the set reference that is causing the problem.
   b. Review and act upon the message that is displayed.

7. Save the report.
Edit a Set

After you create a set, you can view its definition to see, change, add, or delete operations, such as the following, that are performed on the set:
- excluding a member
- moving members to the top or bottom of a set
- applying a top or bottom filter
- filtering the set
- expanding or collapsing members in a set

The definition of a set shows you, in the graphical tree, the history of all the operations that are performed on the set.

You can see and change the order in which the operations are performed. For example, you apply a top three filter and then exclude one member. Your set now contains only two members. You can choose to exclude the member before the top three filter so that your set still contains three members.

Procedure
1. Right-click a set and click Edit Set.
   A graphical tree shows all the operations that are performed on the set of members.
2. To see the details of an operation, hover your mouse over an operation node.
3. To change the order of an operation, click the operation node, and click the right or left arrow.
4. To edit an operation, click the operation node and click the edit button.
5. To add a new operation, click the new button.

Exclude Members from a Set
You can choose to remove individual items that are not needed in your analysis.

Procedure
1. Right-click the set and click Edit Set.
2. Click the new button and click Exclude.
3. In the Exclude box, choose the members to exclude and use the right arrow to move them to the Members pane.

Move Members Within a Set
You can move one or more members to the top or bottom of a set.

Procedure
1. Right-click the set and click Edit Set.
2. Click the new button and click Move to Top or Move to Bottom.
3. In the Available members list, select the member to move to the top or bottom and click the right arrow.
   The members appear in the order shown in the Members list.
**Limit Data to Top or Bottom Values**

You want to focus your report on the items of greatest significance to your business question. For example, you want to identify your top 100 customers and what that group of customers is worth.

You can limit the data to the top or bottom values of a set. This keeps the amount of data shown in the work area small, even when using large data sources.

You can define a top or bottom rule by specifying
- a number, such as the 50 top or bottom performing sales people
- a percentage, such as customers who contribute to the top 10% of overall revenue
- a cumulative sum, such as customers who contribute to the first ten million dollars of overall revenue

If the selected set contains a user-defined filter, the top or bottom rule applies only to the included values. For example, if you apply a filter to show only retailers with revenue greater than one million dollars, the bottom rule applies to the lowest values within those results.

You can filter a set of members to show only those at the top or bottom and base the filter on the measure you are using and the set of members on the opposite edge. You can also specify a custom filter.

**Procedure**

1. Right-click the set and click **Edit Set**.

2. Click the new button and click **Top** or **Bottom**.

3. Under **Type**, select whether to filter the **Top** or **Bottom** values.

4. To choose how and how much you want to filter, under **Number of items**, click **Count**, **Percent**, or **Sum** and type the number of data items to show, the percent of data items to show, or the value to sum.

5. Choose the intersection on which to base your filter by clicking the ellipsis (...) button and selecting the members and measure for the intersection from either the **Source** tab or the **Calculated Members and Measures** tab.

   For more information, see “Create an Intersection (Tuple)” on page 306.

**Filter the Members Within a Set**

You can remove members from within a set by specifying filter conditions so that only the members that you require remain. You can filter a set based on
- member captions
  - For example, your set includes employee names and you want to keep only the employees with names that begin with a specific letter.
- properties that are numeric or strings
  - For example, your set includes employee names and you want to filter employees using the gender property.
- intersection of values (tuple)
  - For example, you want to keep only the employees who used less than ten sick days for the year 2012.
If you want to create a more complex filter, you can combine multiple conditions using AND, OR, and NOT operators. By default, multiple conditions are combined with an AND operator, which means that all conditions must be met for the filter to take effect.

Filtering the members in a set is not the same as relation detail or summary filters.

**Procedure**
1. Right-click the set and click **Edit Set**.
2. Click the new button and click **Set Filter**.
3. Select how you want to filter.
   - If you want to filter the set using a caption, click **Caption**.
   - If you want to filter the set using a member property, click **Property** and select from the drop-down list.
   - If you want to filter the set using an intersection of members, or tuple, click **Intersection (tuple)** and click the ellipsis (...) button. Then, from **Available members and measures**, select the desired items and click the right arrow to move them to the **Intersection members and measures** list.
4. Specify the **Operator** and **Value** to complete your condition and click **OK**.
5. To add additional conditions, click the new button.
6. If you include multiple conditions in your filter, use the AND, OR, and NOT operators to combine them.
   - To add an operator, select the conditions that you want to combine, and click an operator.
   - To change an operator, double-click the operator to toggle among available ones or select the operator and then select a different one from the list.
   - To remove an operator, select it in the condition string, and then click the delete button.
7. To change a condition, click the edit button.

**Expand and Collapse a Member Within a Set**
You can expand a member to add its child members below it as new rows, and you can collapse expanded members to remove the expanded member's child members.

**Procedure**
1. Right-click the set and click **Edit Set**.
2. Click the new button and click **Expand** or **Collapse**.
3. In the **Available members** list, select the member to expand or collapse and click the right arrow to move it to the **Members** list.

**Extended Data Items**
Extended data items differ from the traditional, expression-based text strings by enabling you to add children, view details, select sets, sort sets, and apply contextual calculations.
**Note:** Extended data items do not apply to relational packages or non-dimensionally modeled packages.

You can do the following with extended data items that are not available with traditional data items:

- **Add an extended data item's child members**
  You can double-click an extended data item to insert its child members beside it in the report object.

- **View an extended data item's details**
  You can select an extended data item and see where the data item appears in the package in the **Properties** pane under **Data Item**.

- **Select member sets**
  You can select a member in a set to highlight it as the primary selection and the other members as secondary selections.

- **Sort sets**
  You can select an extended data item, click the sort button, and click **Advanced Set Sorting** to sort a set in ascending or descending order, to sort hierarchically, and to sort by caption, by a property, or by an intersection (tuple).

- **Apply contextual calculations**
  You can select an extended data item and then, from the **Data** menu, click **Calculate** to see a contextual list of calculations that are available for this extended data item.

Extended data items are optional. If you use extended data items in a report, we recommend that you use them for the entire report instead of mixing them with expression-based data items. For information about turning on extended data items, see "Specify Report Properties" on page 350.

**Note:** In IBM Cognos Workspace Advanced, extended data items are always used. Therefore, if you are authoring a report in IBM Cognos Report Studio for report consumers who are working in Cognos Workspace Advanced, you should use extended data items.

**Tips for Working with Ragged or Unbalanced Hierarchies**

In ragged or unbalanced hierarchies, some members that are not at the lowest level of the hierarchy may have no descendants at one or more lower levels. Support for these hierarchy gaps in relational data sources is limited. More complete support is provided for OLAP data sources, but some reports may still result in unexpected behavior.

For example, the following may occur:

- Groups corresponding to missing members may appear or disappear when grouped list reports are pivoted to a crosstab. This happens with set expressions using the `filter` function, and detail filters on members.
- Ragged and unbalanced sections of the hierarchy are suppressed when set expressions in that hierarchy are used on an edge.
- When a crosstab is sectioned or is split into a master detail report, sections corresponding to missing members become empty.
- Cells that were suppressed may still appear in the report output for reports with ragged or unbalanced hierarchies.
Some of these behaviors may be corrected in a future release, while others may be codified as supported behavior. To avoid these behaviors, do not use levels from ragged or unbalanced hierarchies. Instead of using levels, use the descendants, children, or ancestors.

We consider the following scenarios to be safe:
- One or more nested level references on an edge with no modifying expression.
- A hierarchy reference on only one level of one edge.
- One or more explicit members or sets of explicit members as siblings on only one level of one edge.
- Summaries of the previous three scenarios.

In all cases, you should test reports based on ragged and unbalanced hierarchies to confirm that hierarchy gaps are handled correctly.

For more information about ragged or unbalanced hierarchies, see the IBM Cognos Framework Manager User Guide.

Related concepts:
- “Creating Sections on Reports That Access SAP BW Data Sources” on page 480

SAP BW data sources may have problems with sections in reports under different circumstances:

Creating dimensional custom groups

Create custom groups to classify existing data items into groups that are meaningful to you.

You can reduce the number of values into smaller, more meaningful groups. For example, you can change a list of employees into my team and others.

When you are working with dimensional data and crosstabs or charts, you can create custom groups only on sets of members. After you create a custom group in a set, the members in the set are replaced by the custom groups.

When you create custom groups in a list, a new column is added to the report, with the name `data_item (Custom)`. You can use this new column to group or sort the report. If you want to show only the new groups, you can delete the original column.

When you are creating custom groups on sets of members, consider the following constraints:
- Members included in a custom group must be from the same level in a hierarchy.
- If you insert a hierarchy that is used in a custom group elsewhere in the data container, it must be inserted as a sibling of the custom group.
- Automatic aggregation on a custom group might not work as expected. For example, you might encounter error cells (- -).

Procedure
1. To add custom groups in a crosstab or chart that includes sets of members, do the following steps:
   a. Right-click a set and click Edit Set.
b. Click New and select Custom Grouping.

c. In the Custom Grouping window, click New Custom Group.

d. Type a New group name.

e. Select members from the Available members list and move them to the Members list.

2. To add custom groups to a list, do the following steps:
   a. Click the column that you want to group on and from the Data menu, select Calculate > Define Custom Groups.

   Tip: You can click either the column heading or one of the column cells.

   b. Click New Group Entry.

   c. To group by values that you select, click New Select Values Group, type a New Group Name, select the required values from the Values box, and move the values to the Selected Values box.

   d. To group by a range of values, click New Range Group, type a New Group Name, and enter the From and To values.

3. If you do not want a group name to be displayed for remaining values, select Do not show remaining values. Clicking this option produces empty cells for remaining values.

4. If you want the group name for remaining values to match each value, select Use each remaining value as a group name.

5. If you want to specify your own group name for all remaining values, select Group remaining values into a single group and type the name that you want.

6. If you are adding custom groups in a list, under New data item name, select either the default data item name or type a new data item name.

---

**Summarizing Data Dimensionally**

Summarize data in your reports to obtain totals, averages, and so on.

The summary options that you can use depend on the type of data source that you are using. If you are querying an OLAP data source, all measure values that appear in reports are pre-summarized because the data source contains rolled up values. The type of summary that is used is specified in the data source itself. As a result, we recommend that you use the Automatic summary when creating dimensional style reports. This ensures that the report always uses the type of summary that the data modeler specified in the data source.

For example, the modeler may have specified that the rollup for revenue is total and the rollup for stock prices is average.

If you use other types of summaries with dimensional style reports, you may encounter unexpected results.

You can also add summaries, which are supported for any data source. Summaries specify how data items are totaled in the headers and footers of a list and in the total rows and columns of a crosstab. For list reports, these summaries only summarize the data that is visible on that page of the report.
You can specify summary aggregation in different ways by using any of the following:
- aggregation properties that are specified in the model
- the Auto Group & Summarize property
- the summarize button in the toolbar
- aggregation properties for a data item
- the solve order of calculations

**Limitation**

If a summary is applied to a report that contains binary large object (BLOB) data, such as images or multimedia objects, you cannot also perform grouping or sorting.

**The Type of Data**

How data is summarized also depends on the type of data that you are summarizing. Summary rules are applied differently to facts, identifiers, and attributes. For example, if you summarize a data item that represents part numbers, the only summary rules that apply are count, count distinct, count non-zero, maximum, and minimum. For information about how to determine the type of data that a data item represents, see “Add Relational Data to a Report” on page 213 and “Add Dimensional Data to a Report” on page 253.

For information about how null or missing values are handled in summaries, see “Null (Missing) Values in Calculations and Summaries” on page 542.

**Limitations on Measure Rollups**

For all OLAP data sources except IBM Cognos PowerCube and Microsoft SQL Server 2005 Analysis Services (SSAS), aggregation and re-aggregation are supported only for calculations and measures that use the following rollups: Sum (Total), Maximum, Minimum, First, Last, and Count.

All other types of rollup either fail or return error cells, which typically appear as two dash characters (--).

This problem occurs in, but is not limited to, the following:
- footers
- aggregate function
- context filters that select more than one member of a hierarchy that is used elsewhere on the report.
Summarizing Sets

When working with Microsoft SQL Server Analysis Services (SSAS) data sources, we recommend that you not summarize values for sets which contain members that are descendants of other members in the same set. If you do so, SSAS double-counts values for automatic summaries and all data sources double-counts values for explicit summaries.

Related concepts:
“Limitations When Summarizing Measures in DMR Data Sources” on page 534

There are limitations when summarizing dimensionally-modeled relational (DMR) measures and semi-additive measures in crosstabs using the aggregation function count distinct, median, standard-deviation, or variance. The following limitations can produce empty or error cells when the report is run:

Add a Simple Summary

You can add simple summaries in a report by using the summarize button. This button provides a subset of the summary functions available in IBM Cognos Report Studio.

The summarize button sets the rollup aggregate property for the data item to the selected summary, and places the data item into an appropriate footer. A footer is created for each set, hierarchy, or level.

In crosstabs and charts, the summary appears as a node.

In crosstabs, you can add multiple summaries at the same level. For example, you have a crosstab with Product line as rows, Order year as columns, and Revenue as the measure. For Product line, you can add the Total summary as a header, which will total all revenue for each order year. You can then add the Average summary as a footer, which will give the average revenue of all product lines for each order year.

For information about adding a rolling or moving average, see “Rolling and Moving Averages” on page 272.

Procedure
1. Click the item to which to add a summary.

2. Click the summarize button and click a summary type.

3. To change the summary label, do the following:
   - Click the label.
   - In the Properties pane, under Text Source, set the Source Type property to the source type to define the label.
     For example, set it as Data Item Value to produce a dynamic label for the summary based on data item values.
   - Set the property below Source Type to specify the label.
     This property depends on the source type you chose. For example, if you chose Data Item Value as the source type, set the Data Item Value property to the data item to use to define the label.

4. To change a summary, select it and, in the Properties pane, under Data Item, click Aggregation Method and choose a different function.
Summarizing Values in Crosstabs

In crosstabs, summaries are calculated using two summary expressions.

The expressions used to summarize data are:

- `aggregate ([measure within set [data item]])` summarizes the member values from the data source within the current content.

- `aggregate ([measure within detail [data item]])` summarizes the lowest level of details in the report.

The expression that is used to calculate a summary is controlled by the summary’s Use Set Aggregation property. For example, in the following crosstab, if you specified Total as the summary, the following aggregate expressions are produced when the Use Set Aggregation is set to Yes and No respectively:

- **Total ([Revenue] within set [Quarter])**
  
  This expression totals the quarter values from the data source at the intersecting product line. At the bottom right corner, it totals the aggregate over all product lines for each quarter.

- **Total ([Revenue] within detail [Quarter])**
  
  This expression totals the month values visible in the report at the intersecting product line. At the bottom right corner, it totals all of the intersecting month-product line values visible in the report.

In simple cases, the members and values visible in the report and the aggregate rules in the report are the same as those in the data source and all of these expressions produce the same results.

For example, for the quarter and month values, if you are totaling the values for all months in all quarters, it makes no difference whether the visible values, the values in the cube, or the month and quarter values are used. The result is the same.
Tip: The Total Revenue by Country or Region sample report in the GO Data Warehouse (query) package includes a total summary function. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Different results appear when you start filtering, changing aggregation types, or using set expressions or unions.

For example, the following crosstab shows the quantity of products sold across all product types for each product line. The bottom of the crosstab has two summary values that show the average quantity of products sold by product line.

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Month</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005 Q 3</td>
<td>2005/Sep</td>
<td>$5,300,599.08</td>
</tr>
<tr>
<td>2005 Q 4</td>
<td>2005/Oct</td>
<td>$11,778,348.92</td>
</tr>
<tr>
<td></td>
<td>2005/Nov</td>
<td>$3,355,399.46</td>
</tr>
<tr>
<td></td>
<td>2005/Dec</td>
<td>$4,645,069.28</td>
</tr>
<tr>
<td>2006 Q 1</td>
<td>2006/Jan</td>
<td>$3,119,227.22</td>
</tr>
<tr>
<td></td>
<td>2006/Feb</td>
<td>$3,045,478.38</td>
</tr>
<tr>
<td></td>
<td>2006/Mar</td>
<td>$6,867,919.10</td>
</tr>
<tr>
<td>2006 Q 2</td>
<td>2006/Apr</td>
<td>$8,306,586.26</td>
</tr>
<tr>
<td></td>
<td>2006/May</td>
<td>$5,129,112.24</td>
</tr>
<tr>
<td></td>
<td>2006/Jun</td>
<td>$6,927,447.24</td>
</tr>
<tr>
<td>2006 Q 3</td>
<td>2006/Jul</td>
<td>$5,078,789.96</td>
</tr>
<tr>
<td></td>
<td>2006/Aug</td>
<td>$5,180,166.38</td>
</tr>
<tr>
<td></td>
<td>2006/Sep</td>
<td>$4,692,742.38</td>
</tr>
<tr>
<td>2006 Q 4</td>
<td>2006/Oct</td>
<td>$15,887,237.28</td>
</tr>
<tr>
<td></td>
<td>2006/Nov</td>
<td>$3,609,599.42</td>
</tr>
<tr>
<td></td>
<td>2006/Dec</td>
<td>$6,240,652.38</td>
</tr>
<tr>
<td><strong>Total (Quarter)</strong></td>
<td></td>
<td><strong>$171,576,307.80</strong></td>
</tr>
</tbody>
</table>
Each summary value uses a different expression that is set by the value specified for the Use Set Aggregation property.

- **Average (Product line) - within detail**
  This summary is the average of the detail values in the crosstab.

- **Average (Product line) - within set**
  This summary is the average of the product type values rolled up into sets at the Product line level. The values are obtained from the data source. If filters or slicers exist, the values are recomputed using the aggregation rules defined in the data source.

In most cases, you should use the within detail expression because the results are easiest to understand and the same as the results for footers in a grouped list report. The within set expression should be reserved for reports with a purely dimensional focus, such as when there are no detail or summary filters defined in the report.

**Summarizing Member Sets**

When you work with dimensional data sources and you summarize member sets, if an explicit summary function such as **Total** is used and the set contains
duplicates, the result is double-counted. If you are using an OLAP data source, the result produced for the automatic summary function depends on the data source.

For example, the product line rows below were defined using the expression

union ([Product line], [Camping Equipment], ALL)

where [Product line] is the level that contains Camping Equipment.

<table>
<thead>
<tr>
<th>Product Line</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Equipment</td>
<td>$1,569,036,664.03</td>
</tr>
<tr>
<td>Mountaineering Equipment</td>
<td>$409,650,132.9</td>
</tr>
<tr>
<td>Personal Accessories</td>
<td>$1,885,673,307.78</td>
</tr>
<tr>
<td>Outdoor Protection</td>
<td>$75,994,296.25</td>
</tr>
<tr>
<td>Grill Equipment</td>
<td>$726,411,367.89</td>
</tr>
<tr>
<td>Camping Equipment</td>
<td>$1,569,036,664.03</td>
</tr>
<tr>
<td>Summary</td>
<td>$4,685,775,765.85</td>
</tr>
<tr>
<td>Total</td>
<td>$6,275,812,432.85</td>
</tr>
</tbody>
</table>

Figure 42. Crosstab showing revenue by product line with an aggregate row and a total row

For PowerCubes, Aggregate (Product line) is the sum of the product lines excluding duplicates. For more information about how the aggregate function is processed, see “Summary Functions” on page 273.

Rolling and Moving Averages

Use rolling and moving averages to analyze data for specific time series and to spot trends in that data. When viewing these averages on a line chart, use a longer period of time to reveal long-term trends.

In IBM Cognos Report Studio, to add a rolling or moving average, you must create summary and custom calculations using layout expressions.[Insert a Query Calculation] on page 303.

The Rolling and Moving Averages interactive sample report includes rolling and moving calculations. For more information about The Sample Outdoors Company, see Appendix C, “Sample Reports and Packages,” on page 507.

Rolling Average

A rolling average continuously updates the average of a data set to include all the data in the set until that point. For example, the rolling average of return quantities at March 2012 would be calculated by adding the return quantities in January, February, and March, and then dividing that sum by three.

Tip: The Historical Revenue sample report in the Sales and Marketing (Cube) package also includes a rolling average. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.
Moving Average

A moving average calculates the average of a data set for a specified period. For example, the moving average of return quantities at March 2012 with a specified period of two would be calculated by adding the return quantities in February and March and then dividing that sum by two. In IBM Cognos Report Studio, you can use a prompt to specify the period.

Show Data for Specific Time Periods

Show data for a specific time period to associate a report with a business time period rather than the execution time. For example, you have a monthly report that you run at the beginning of each month, and you want the last business day of the previous month to appear rather than the day on which you run the report.

If the report functions AsOfDate and AsOfTime are added to the report, they will return a value based on the results of the As of Time Expression object. If the As of Time Expression object is not added to the report, these two functions return the date and time at which the report is run.

If the As of Time Expression object is added more than once in the report, the first occurrence of the object in the layout that returns a valid value is used.

Procedure

1. From the Toolbox tab, drag As of Time Expression to the report.
2. Double-click As of Time Expression.
3. In the Expression Definition box, type the expression.
   The expression must return a date-time value.
   Tip: You can drag a function that returns a constant from the Constants folder in the Functions tab. You can then change the constant to the value to use.
   If you do not specify a time, the default time 12:00:00.000 AM is used.

Summary Functions

This section describes the summary functions that are available in IBM Cognos Report Studio. Some functions, such as Custom, are available only when you click the summarize button on the toolbar. Additional summary functions are available in the expression editor.

Automatic Summary

Depending on the type of data item, applies the function None, Summarize, or Calculated based on the context in which the data item appears.

Calculated is applied if the underlying data source is OLAP. It is also applied if the data item expression
• contains a summary function
• is an if then else or case expression that contains a reference to at least a modeled measure in its condition
• contains a reference to a model calculation or to a measure that has the Regular Aggregate property set to a value other than Unsupported
contains a reference to at least one data item that has the Rollup Aggregate Function property set to a value other than None

If the underlying data source is relational and if the data item expression contains no summary functions and a reference to at least one fact that has the Regular Aggregate property set to a value other than Unsupported, Summarize is applied.

If the underlying data source is OLAP and if the data item expression contains no summary functions and a reference to at least one fact that has the Regular Aggregate property set to a value other than Unsupported, Calculated is applied.

If the underlying data source is SAP BW, reports containing aggregations and summaries run more efficiently if the aggregation applied to a query item matches the aggregation rule for the underlying key figure on the SAP BW server. In Report Studio, the easiest way to accomplish this is to change the value of the Aggregate Function property to Automatic.

In all other contexts, None is applied.

For relational and dimensionally modeled relational (DMR) data sources, if this function is applied to a data item expression that has the average function, weighted averages are computed based on the lowest level detail values in the data source.

Note: This is the default function.

**Average**

Adds all existing values and then divides by the count of existing values.

**Calculated**

Specifies that all the terms within the expression for a data item are aggregated according to their own rollup rules, and then the results of those aggregations are computed within the overall expression.

For example, a list contains the data item Quantity with the Aggregation property set to Total. You add a query calculation named Quantity Calculated. You define its expression as Quantity + 100 and you set its Aggregation property to Calculated. When you run the report, the values for Quantity Calculated are computed by first adding all values for Quantity and then adding 100 to each value.

Note: In crosstabs, this function overrides any solve orders that are specified.

**Count**

Counts all existing values.

If the underlying data source is OLAP, Count behaves as follows if it is specified in the Rollup Aggregate Function property for a data item.

<table>
<thead>
<tr>
<th>Object</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Count distinct is used. A warning appears when you validate the report.</td>
</tr>
</tbody>
</table>
If the underlying data source is dimensionally modeled relational (DMR), Count behaves as follows if it is specified in the **Aggregate Function** or **Rollup Aggregate Function** property for a data item.

<table>
<thead>
<tr>
<th>Object</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member set</td>
<td>Count distinct is used. A warning appears when you validate the report.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Not supported. An error is returned when you run the report.</td>
</tr>
<tr>
<td>Measure</td>
<td>Supported.</td>
</tr>
</tbody>
</table>

If you add the count summary to a non-fact column in a list and then group the column, the column will not be grouped when you run the report. To resolve this issue, group the column first before adding the count summary.

**Count Distinct**

Returns the total number of unique non-null records.

If the underlying data source is OLAP, count distinct behaves as follows if it is specified in the **Rollup Aggregate Function** property for a data item.

<table>
<thead>
<tr>
<th>Object</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Count distinct is used. A warning appears when you validate the report.</td>
</tr>
<tr>
<td>Member set</td>
<td>Count distinct is used. A warning appears when you validate the report.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Supported.</td>
</tr>
<tr>
<td>Measure</td>
<td>Supported.</td>
</tr>
</tbody>
</table>

If the underlying data source is dimensionally modeled relational (DMR), count distinct is supported for levels, member sets, attributes, and measures when it is specified in the **Aggregate Function** or **Rollup Aggregate Function** property for a data item.

**Custom**

Summarizes data based on an expression that you define.

**Maximum**

Selects the largest existing value.
Median
Returns the median value of the selected data item.

Minimum
Selects the smallest existing value.

None
Does not aggregate values.

If the underlying data source is relational, the data item is grouped when the query property **Auto Group & Summarize** is set to **Yes**.

Not Applicable
This function is available only for the **Aggregate Function** property. It specifies that the aggregate attribute is ignored. For example, the calculation will be applied after the data is aggregated.

This function differs from the **Calculated** function, which applies aggregation to the rows and then performs the calculation.

For example, for the average ( [Quantity] ) + 5 expression, when the **Aggregate Function** property is set to **Not Applicable**, five is added to the average of the single Quantity value. When the function is set to **Calculated**, five is added to the average of a set of numbers.

This setting should not be applied to a simple model reference.

This setting is relevant only for reports that are upgraded from IBM Cognos ReportNet 1.x.

Standard Deviation
Returns the standard deviation of the selected data item.

**Note:** From a mathematical perspective, this function is not useful for small numbers of items and is not supported if the query property **Auto Group & Summarize** is set to **Yes**.

Summarize
Aggregates data based on model or data type information. This function can be thought of as a calculate and then aggregate rule. Data is aggregated using the following rules:

If the underlying data source type is relational and the data item or calculation is a reference to a single fact query item in the model that has the **Regular Aggregate** property set to **Sum, Maximum, Minimum, Average, or Count**, aggregate data using this function. Otherwise, data is aggregated according to the data type of the data item as follows:

- **Total** is applied for numeric and interval values.
- **Maximum** is applied for date, time, and date-time values.
• **Count** is applied for everything else.

The underlying data source type can be dimensional and the data item or calculation a reference to a single item in the model. In this case, if the Regular **Aggregate** property of the model item is not **unknown**, the summary function that corresponds to the Regular **Aggregate** property is applied. If the Regular **Aggregate** property is **unknown**, set the function to **Calculated**.

For example, a list contains the data item Quantity with the **Aggregation** property set to **Average**. You add a query calculation named Quantity Summarize. You define its expression as Quantity + 100 and you set its **Aggregation** property to **Summarize**. When you run the report, the values for Quantity Summarize are computed by first adding 100 to each value for Quantity and then calculating the total. The total is calculated because Quantity contains numeric values and the expression for Quantity Summarize is not a simple data item reference. If Quantity Summarize is defined as Quantity, the function **Average** is applied to each value.

**Total**

Add all existing values.

**Variance**

Returns the variance of the selected data item.

**Note:** From a mathematical perspective, this function is not useful for small numbers of items and is not supported if the query property **Auto Group & Summarize** is set to Yes.

**Mapping Summary Functions From Data Sources to Framework Manager and Report Studio**

The summary functions available in IBM Cognos Framework Manager and IBM Cognos Report Studio reflect summary functions supported by relational and dimensional data sources. The following table shows how summary functions in data sources are mapped to Framework Manager and Report Studio.

<table>
<thead>
<tr>
<th>Data source</th>
<th>Framework Manager</th>
<th>Report Studio</th>
</tr>
</thead>
<tbody>
<tr>
<td>None specified, or none</td>
<td>Unsupported</td>
<td>None</td>
</tr>
<tr>
<td>average (avg)</td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td>count</td>
<td>Count</td>
<td>Count</td>
</tr>
<tr>
<td>count distinct</td>
<td>Count Distinct</td>
<td>Count distinct</td>
</tr>
<tr>
<td>maximum (max)</td>
<td>Maximum</td>
<td>Maximum</td>
</tr>
<tr>
<td>minimum (min)</td>
<td>Minimum</td>
<td>Minimum</td>
</tr>
<tr>
<td>median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td>standard deviation (stddev)</td>
<td>Standard Deviation</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>sum</td>
<td>Sum</td>
<td>Total</td>
</tr>
<tr>
<td>variance (var)</td>
<td>Variance</td>
<td>Variance</td>
</tr>
<tr>
<td>count non zero</td>
<td>Count Non-Zero</td>
<td>Automatic</td>
</tr>
<tr>
<td>average non zero</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
<tr>
<td>external</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
<tr>
<td>any</td>
<td>unknown</td>
<td>Automatic</td>
</tr>
</tbody>
</table>
The following summary functions exist only in either Framework Manager or Report Studio. There is no corresponding summary function in data sources.

<table>
<thead>
<tr>
<th>Framework Manager</th>
<th>Report Studio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Automatic</td>
</tr>
<tr>
<td>Calculated</td>
<td>Calculated</td>
</tr>
<tr>
<td>No corresponding summary function</td>
<td>Summarize</td>
</tr>
<tr>
<td>No corresponding summary function</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

**Focusing Dimensional Data**

In IBM Cognos Report Studio, there are three approaches to focusing dimensional data in a crosstab:

- drag data items to the edges of your crosstab
- drag data items to the context filter area
- create set expressions on the edges of your crosstab that return members or sets of members, such as the `filter` function

**Note:** Summary and detail filters are recommended only for focusing data in relational reporting.

The approach you choose should depend on the capabilities of the data source, the performance requirements, the persistence of the dataset, and how complex you want the report to be.

If you filter values that are floating-point numbers, you may encounter imprecise filtering results due to rounding or data truncation. Floating-point numbers may appear differently in the report than how they are stored in the data source. To ensure accurate results, your filter criteria should account for any rounding issues.

Consider the limitations when filtering dimensional data and when filtering data from SAP BW data sources.

You can also use prompts, or prompt controls, to allow users to provide input for the filter.
Related concepts:
“Limitations When Filtering Dimensional Data Sources” on page 534

Avoid combining context filters (slicers) with dimensional constructs that involve members from hierarchies that are used elsewhere in the report. This combination gives results that are often not what you might expect and that may change in a future release.

Create a Context Filter

When working with dimensional data, you can use context filters, or slicer filters, to quickly focus your report on a particular view of the data.

For example, the following crosstab contains product lines in the rows, years in the columns, and revenue as the measure. We want to filter the values to show us the revenue for only Web orders from Asia Pacific. To change the context, you drag Asia Pacific and Web from the source tree to the Context filter section of the overview area. The crosstab then shows the revenue for only Asia Pacific and Web. Changing context changes only the values that appear. It does not limit or change the items in the rows or columns.

<table>
<thead>
<tr>
<th>Revenue</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equipment</td>
<td>36,415,479.91</td>
<td>77,222,359.67</td>
<td>121,312,260.99</td>
<td>83,623,152.78</td>
</tr>
<tr>
<td>Mountaineering Equipment</td>
<td>18,115,590.33</td>
<td>37,258,799.5</td>
<td>51,810,456.11</td>
<td></td>
</tr>
<tr>
<td>Personal Accessories</td>
<td>67,582,845.5</td>
<td>94,568,084.5</td>
<td>136,872,087.5</td>
<td>106,879,947.27</td>
</tr>
<tr>
<td>Outdoor Protection</td>
<td>4,335,492.66</td>
<td>4,388,590.73</td>
<td>2,241,022.7</td>
<td>1,167,650.48</td>
</tr>
<tr>
<td>Golf Equipment</td>
<td>8,586,746.5</td>
<td>27,061,602.93</td>
<td>49,822,888.09</td>
<td>39,248,971.12</td>
</tr>
</tbody>
</table>

Figure 43. Crosstab showing revenue by product line by year filtered to show only Asia Pacific and web orders

The members that are used as the context filter appear in the report header when you run the report.

Any summary values in the report are recomputed to reflect the results returned by the context filter.

You can create multiple context filters to filter across two or more different hierarchies.

Tip: The Budget vs. Actual sample report in the Sales and Marketing (cube) package includes a context filter. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Note: There are limitations when using context filters:
- Calculated measures do not get suppressed by the filter. To avoid this, drag the desired members from the Source tab directly onto the row or column edge instead of using the members in the context filter.
- Suppression can be slow on large cubes, because all of the data is retrieved and then filtered.
If the desired data in the context filter is from a hierarchy that appears on the

crosstab edge, drag the desired members from the Source tab directly onto the
crosstab edge.

If the desired data is from a hierarchy that does not appear on the crosstab edge,
use a context filter in conjunction with other means to limit the data, such as a
top or bottom filter.

To use a calculation or expression to define a context filter, create a slicer member
set using the query explorer. A slicer member set is an expression that returns a set
of members from the same dimension. For example, you can define a slicer
member set that filters for the top five products with the greatest revenue.

Context filters are different from other filters. When you filter data, members that
do not meet the filter criteria are removed from the report. A context filter does not
remove members from a report. Instead, their values are filtered or and you see
blank cells.

You can also use context filters as prompts when the report is run.

When creating context filters,
• use only members from hierarchies that are not already projected on an edge of
  the crosstab
• use only one member per hierarchy

Create a simple context filter
When working with dimensional data, you can use context filters, or slicer filters,
to quickly focus your report on a particular view of the data.

Procedure
1. If you do not see the Context filter area, show it using the View menu (Panes,
   Context Area).

2. From the Source tab, select or search for one or more items on which to
   filter.
3. Drag the item on which to filter into the Context filter section of the overview
   area.
4. To change context, select a new item from the Context filter box.

Create an advanced slicer filter
To use a calculation or expression to define a context filter, create a slicer member
set using the query explorer. A slicer member set is an expression that returns a set
of members from the same dimension. For example, you can define a slicer
member set that filters for the top five products with the greatest revenue.

Procedure
1. Pause the pointer over the query explorer button and click the query to
   which to add a slicer.
2. On the Toolbox tab, drag Slicer Member Set to the Slicer pane.
3. Drag the members to the Expression Definition box.
   You must join the members in the expression by using the set function. For
   example, the following expression contains the Fax and Telephone members
   from the Order Method dimension:
4. To add members from another dimension, repeat step 2 to create a separate slicer member set.

**Define a Prompt Using a Context Filter**

If context filters (or slicers) are defined, you can use these filters to create prompts in the report.

Prompts provide questions that help users to customize the information in a report to suit their own needs. For example, you create a prompt so that users can select a region. Only data for the specified region is retrieved and shown in the report.

**Before you begin**

Before you define a prompt, you must create context filters.

**Procedure**

1. In the **Context filter** section of the overview area, click the down arrow next to the filter to use as a prompt.

2. Click **Prompt** and do one of the following:
   - To remove the prompt, click **No Prompt**.
   - To allow users who run the report to select any member from the hierarchy, click **Prompt on Hierarchy**.
   - To allow users who run the report to select from a list of members in the current level, click **Prompt on Level**.

**Sorting Dimensional Data**

You can sort items to view them in your preferred order. By default, IBM Cognos Report Studio retrieves items in the order defined in the data source. OLAP data sources always have a defined order. The data modeler defines the sorting options in the model.

For more information about data modeling, see the IBM Cognos Framework Manager **User Guide**.

**Tip:** The Sales Growth Year Over Year sample report in the GO Data Warehouse (analysis) package includes sorting. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Limitations When Sorting SAP BW Data**

If you use a SAP BW data source, each level in a hierarchy has an item with the same name as the level and a role of _businessKey. Such items are known as level identifiers. You should use only the level identifier of all the time-related characteristics, such as 0CALDAY and 0CALMONTH, for sorting. All other (attribute) items in these hierarchies are formatted string representations of the characteristic values with which they are associated. These formatted values sort alphanumerically and not chronologically.

**Tip:** By level identifier, we are referring to a query item with the same name as the level that has the role of _businessKey. Each level in a SAP BW hierarchy has a level identifier.
Sort Sets by Label

You can sort sets of members based on their label (or caption). By default, when you select Sort by Label, Ascending or Descending, the members within the set are sorted only by their labels. You can also perform advanced sorting to sort sets using an intersection (tuple) or a member property.

For example, a set contains regions and you sort the regions within the set in descending alphabetical order. Only extended data items that represent sets support set sorting.

Before you can perform set sorting, you must create sets for the members in your report.

You cannot sort data items from different dimensions that are intermixed. For example, if you have years and products on the same edge and at the same level in your crosstab, you can sort the years, and you can sort the product lines, but you cannot sort them as combined set because years and products come from different dimensions.

Procedure

1. In a crosstab, select a set to sort.

2. Click the sort button and under Sort by Label, click Ascending or Descending.

   Tip: To remove a sort order, click Don't Sort.

Sort Sets in the Opposite Axis by Value

You can sort members in the sets on the opposite axis based on the value of a member or measure that you select.

For example, a crosstab contains years inserted as individual members in the columns, product lines inserted as a member set in the rows, and revenue as the measure. You select the 2012 column and then select Sort by Value, Descending. The values in the 2012 column are sorted.

<table>
<thead>
<tr>
<th>Revenue</th>
<th>2012</th>
<th>2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal Accessories</td>
<td>994,009,408.42</td>
<td>943,619,493.85</td>
</tr>
<tr>
<td>Camping Equipment</td>
<td>500,382,422.83</td>
<td>352,910,329.97</td>
</tr>
<tr>
<td>Golf Equipment</td>
<td>230,110,270.55</td>
<td>174,740,819.29</td>
</tr>
<tr>
<td>Mountaineering Equipment</td>
<td>161,039,823.26</td>
<td>141,520,649.7</td>
</tr>
<tr>
<td>Outdoor Protection</td>
<td>10,249,175.84</td>
<td>4,471,025.26</td>
</tr>
</tbody>
</table>

Figure 44. Crosstab showing revenue by year by product line with the 2012 column sorted in descending order

Before you can perform this sorting, you must create sets for the members in your report.

In nested crosstabs, items are sorted based on the values of the innermost nested row or column of the opposite axis. For example, a crosstab contains years in the columns and retailers nested within product line in the rows. Select 2004, sort in descending order, and you see retailers listed from the largest value to the smallest value for each product line.
Procedure
1. In a crosstab, select a member or measure to sort.
   For example, to sort a set in the row axis, select a member or measure in the column axis.

2. Click the sort button and under Sort by Value, click Ascending or Descending.

   Tip: To remove a sort order, click Don't Sort.

Perform Advanced Set Sorting
If you use dimensional data, you can sort using the property of a member.

For example, your report includes employee names and you want to sort the employees using the gender property. You can also sort using an intersection of members, also known as a tuple. For example, you want to sort the employees using the value of sick days taken for the year 2012.

Procedure
1. Select a set.

2. Click the sort button and under Other Sort Options, click Edit Set Sorting.

3. Specify the sorting options.

4. If you are sorting members from different levels and want to preserve the hierarchy, select the Hierarchized sorting check box.

5. To sort items using a member property, under Sort by, double-click Property, navigate the tree, and select the member property to use.

6. To sort items using an intersection of members, or a tuple, under Sort by, click Intersection (tuple) and click the ellipsis (...) button. Then, from the Available members and measures box, select the items to use and click the right arrow to move them to the Intersection members and measures box. You can also select items from the Calculated Members and Measures tab.

Working with Dimensional Queries
Queries specify what data appears in the report. In IBM Cognos Report Studio, you create and modify queries using Query Explorer.

Query Explorer provides an alternative way to modify existing reports or to author new reports. You can use Query Explorer to perform complex tasks and other tasks that are more difficult to do in the report layout. For example, use Query Explorer to

• improve performance by changing the order in which items are queried from the database, or by changing query properties to allow the report server to execute queries concurrently where possible

Note: Only queries that are referenced in the report layout can be executed concurrently. If a query is defined in Query Explorer by joining two or more existing queries, IBM Cognos Business Intelligence cannot execute the source queries concurrently. By default, queries run sequentially. Your administrator must enable the concurrent query execution feature. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.
• view or add filters and parameters and modify their properties
• view or add dimensions, levels, and facts
• incorporate SQL statements that come from other reports or reports that you write
• create complex queries using set operations and joins

**Related concepts:**

Queries determine what data items appear in the report. Sometimes you want detailed rows of data, which you obtain by using a simple SELECT statement. Other times you must calculate totals or averages using summary functions and grouped columns or must apply filters to show only the data you want.

**Specify a List of Data Items for an Object**

Specify the list of data items for an object when you must reference a data item that is in a query but is not in the layout.

For example, you want to add a layout calculation to a list that uses a data item that is in the query definition. If the data item does not appear in the list, you must reference it in order to make the layout calculation work. You must also specify the list of data items if you apply conditional formatting that uses a data item that is not in the query.

**Procedure**

1. Click a layout object.

   **Tip:** For a list of layout objects for which you can specify a list of properties, see the Properties property in Appendix G, “Report Studio Object and Property Reference,” on page 771.

2. In the Properties pane, double-click the Properties property and select data items.

**Relate a Query to a Layout**

Queries and layouts work together. After you decide the type of data that you need, you must create a layout in which to show the results. Each column of data must be both selected for the query and shown in a layout unless there are some columns that you do not want to show. The query and layout portions of a report must be linked to produce a valid report.

IBM Cognos Report Studio automatically links query and layout. For example, when you use Report Studio and the list report layout, query and layout are automatically linked.

**Procedure**

1. Select a data container.
2. In the Properties pane, set the Query property to a query.
3. From the Data Items tab, drag data items from the query to the data container.

**Using Dimensional Data Sources with Queries**

When you are working with SAP BW data sources, you can use only a single hierarchy in a query.
Creating queries using a mix of OLAP and relational data is not supported. If you create queries using a database for which you do not know the type, consult your database administrator or modeler.

When performing multi-cube queries using dimensional data sources, the following restrictions apply:
- Only basic operators (+, *, /, -) are available for cross-cube calculations.
- Inner joins are not supported.
- All joins for multi-cube queries are outer joins.
- Viewing MDX using the Generated SQL/MDX query property may not show you the actual MDX that is executed. This is because when a multi-cube query is executed, a number of smaller queries are actually executed.

Related concepts:

“Problems with large SAP BW queries” on page 498

When you are working with an SAP BW data source, if your report includes a data set greater than 1,000,000 cells, you might encounter an XQE error.

Connecting Queries Between Different Data Sources

This section lists the data sources between which you can and cannot connect queries.

IBM Cognos Business Intelligence supports the following:
- RDBMS to RDBMS joins
- set operations of any two queries
- master detail relationships between any two queries
- drill from any query to any other query

You cannot create the following types of joins:
- cube-to-cube (homogeneous)
- cube-to-cube (heterogeneous)
- cube-to-RDBMS
- cube-to-SAP BW
- SAP-BW-to-RDBMS

Multiple-fact Queries

You must understand the result of a query on more than one fact table to achieve the results that you want. The result of a multiple-fact query varies depending on whether you are working with conformed or non-conformed dimensions, on the level of granularity, and on the additive nature of the data.

The GO Data Warehouse (analysis) package has characteristics that affect the results if you use a multiple-fact query with the Inventory and Sales namespaces. Granularity for time differs in that inventory levels are recorded monthly and sales are recorded daily. Also, Sales includes a non-conformed dimension, Order method.

The following examples will help you interpret the results of a multiple-fact query and understand the options for changing a query to obtain the results that you want.
Conformed Dimensions

Individual queries on Inventory and Sales by Quarter and Product yield the following results.

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Product</th>
<th>Opening Inventory</th>
<th>Closing Inventory</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 2011</td>
<td>Husky Rope 50</td>
<td>0</td>
<td>4,859</td>
<td>9,259</td>
</tr>
<tr>
<td></td>
<td>Aloe Relief</td>
<td>10,614</td>
<td>6,513</td>
<td>8,805</td>
</tr>
<tr>
<td></td>
<td>Course Pro Umbrella</td>
<td>34,907</td>
<td>37,327</td>
<td>70,256</td>
</tr>
<tr>
<td>Q1 2011</td>
<td>Average</td>
<td>15,174</td>
<td>16,231</td>
<td></td>
</tr>
<tr>
<td>Q1 2011</td>
<td>Total</td>
<td></td>
<td></td>
<td>88,320</td>
</tr>
<tr>
<td>Q2 2011</td>
<td>Husky Rope 50</td>
<td>4,859</td>
<td>5,728</td>
<td>8,726</td>
</tr>
<tr>
<td></td>
<td>Aloe Relief</td>
<td>6,513</td>
<td>7,379</td>
<td>8,266</td>
</tr>
<tr>
<td></td>
<td>Course Pro Umbrella</td>
<td>37,327</td>
<td>39,870</td>
<td>57,964</td>
</tr>
<tr>
<td>Q2 2011</td>
<td>Average</td>
<td>16,231</td>
<td>17,869</td>
<td></td>
</tr>
<tr>
<td>Q2 2011</td>
<td>Total</td>
<td></td>
<td></td>
<td>74,856</td>
</tr>
</tbody>
</table>

A query on multiple facts and conformed dimensions respects the cardinality between each fact table and its dimensions and returns all the rows from each fact table. The fact tables are matched on their common keys, Product and Time.

Product and Time apply to both Inventory and Sales. However, inventory levels are recorded monthly and sales are recorded daily. In this example, results are automatically aggregated to the lowest common level of granularity. Quantity, which comes from Sales, is rolled up to months.

Nulls are often returned for this type of query because a combination of dimensional elements in one fact table may not exist in the other. For example, if Husky Rope 50 was available in inventory in Q1 2011, but there were no sales of this product in the same time period, Quantity would show a null in the Husky Rope 50 row.

IBM Cognos Business Intelligence does not support conformed dimensions generated by IBM Cognos Framework Manager for SAP BW data sources.

Non-conformed Dimensions

If a non-conformed dimension is added to the query, the nature of the results returned by the query is changed.
Order method type exists only in Sales. Therefore, it is no longer possible to aggregate records to a lowest common level of granularity because one side of the query has dimensionality that is not common to the other side of the query. Opening inventory and Closing inventory results are repeated because it is no longer possible to relate a single value from these columns to one value from Quantity.

Grouping on the Quarter key demonstrates that the result in this example is based on the same data set as the query on conformed dimensions. Summary values are the same. For example, the total quantity for Q1 2011 is 88,320 in both examples.

**Filters on Non-conformed Dimensions**

By default, to improve performance, filters are applied at the database level. The default behavior can result in unexpected nulls when you add a filter to a multiple-fact query. If you create a filter on the Order method dimension to show only the fax order method and apply the filter at the data source, the report includes nulls.
The filter is applied only to one subject area, Sales. Because Order method type does not exist in Inventory, all products still appear in the report. For example, Course Pro Umbrella was in inventory in Q2 2012. Because there were no sales using the fax order method for this product in Q2 2012, Order method type and Quantity are null.

To remove the nulls, change the filter so that it is applied to the result set instead of the data source. By applying the filter to the result set, you get the same results that you would get if you were working with conformed dimensions. Only the products that were ordered using the fax order method appear in the report.
The overall summary for Quantity is 8,043 using either filter method, which shows that the results are based on the same data set.

In IBM Cognos Report Studio, there are two types of filters. A detail filter is applied to the data source. A summary filter is applied to the result set. For more information about how to apply detail and summary filters, see “Create a Detail or Summary Filter” on page 229.

Add a Query to a Dimensional Report

You can create multiple queries in Query Explorer to suit your particular needs. For example, you can create a separate query for each data container in a report to show different data.

**Tip:** The Briefing Book sample report in the GO Sales (analysis) package and the Top 10 Retailers for 2011 sample report in the GO Data Warehouse (analysis) package include multiple queries. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Procedure**

1. Pause the pointer over the query explorer button and click **Queries**.

2. From the **Toolbox** tab, drag one of the following objects to the work area.

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>Adds a query.</td>
</tr>
<tr>
<td>Object</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>SQL</td>
<td>Adds SQL commands.</td>
</tr>
<tr>
<td>MDX</td>
<td>Adds MDX commands.</td>
</tr>
</tbody>
</table>

**Note:** When adding queries to the report

- right-click the work area and click **Show Package Sources** to see the queries that use data items from a package
- right-click the work area and click **Expand References** to see the relationships that exist between queries in the report, which is useful when you are creating complex queries

3. In the **Properties** pane, set the object properties.
4. Double-click a query.
5. From the **Source** tab, drag data items to the **Data Items** pane.

**Tip:** You can add data items to the query that you do not want to appear in the layout. For example, to filter on Product line code and show Product line in the layout, you must add both data items to the query.

6. To create a new data item, from the **Toolbox** tab, drag **Data Item** to the **Data Item** pane.
7. To add a filter, from the **Toolbox** tab, drag **Filter** to the **Detail Filters** or **Summary Filters** pane and define the filter expression.

**Tip:** You can also create a filter by dragging a data item from the **Source** tab to one of the filters panes and completing the filter expression. When working with a dimensional data source, you can quickly filter data by dragging a member to a filter pane instead of dragging the level to which the member belongs. For example, dragging the member 2012 from the Years level will filter data for the year 2012. This method is quicker than dragging the Years level and specifying data only for the year 2012 in the filter expression.

### Add Dimension Information to a Query

Add dimension information to a query when the data source has no dimension information available or when you want to override, extend, or restrict the information.

Add dimension information if any of the following apply:

- There is no dimension information available in the data source.
  
  For example, the data source contains flat data that you want to model dimensionally.
- You want to override the dimension information in the data source.
- You want to extend or restrict dimension information in the data source.

If you do not add dimension information and the data source contains dimensions, then the dimension structure of the data source is used. If the data source contains no dimensions, IBM Cognos Business Intelligence creates a default dimension structure.
Dimension information is not intended to define the presentation of information, but to help query planning. Dimension information can be considered as a form of query hint. In addition, users will not be able to drill down on dimensions that you add to a report.

**Procedure**

1. Pause the pointer over the query explorer button and click a query.
2. In the Properties pane, set the **Override Dimension Info** property to **Yes**.
   The **Dimension Info** tab appears in the work area.
3. Click the **Dimension Info** tab.
4. To create a dimension from an existing data item, from the **Source** tab, drag the data item to the **Dimensions** pane.
   IBM Cognos Report Studio automatically generates the entire dimension.
5. To create a new dimension, from the **Toolbox** tab, drag **Dimension** to the **Dimensions** pane, and then build the new dimension by adding objects to the dimension:
   • To create a level, drag **Level**.
   • To create a level hierarchy, drag **Level Hierarchy**.
   • To create a member hierarchy, drag **Member Hierarchy**.
   For information about these objects, see "Add Dimensional Data to a Report" on page 253.
6. From the **Source** tab, drag data items to the objects you added in the previous step.
   For example, if you created a level, define a key by dragging a data item to the **Keys** folder.
7. To create a fact, from the **Source** tab, drag a data item to the **Facts** pane.

**Example - Create a Crosstab Report that Shows Empty Rows**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a crosstab report that shows the total revenue per year for each product line broken down by order method. By default, if no revenue was produced for a particular product line in a specific year, no row appears in the crosstab for that product line and year. You override the dimension information of the crosstab so that empty rows appear in the report.

**Procedure**

1. Open IBM Cognos Report Studio with the **GO Data Warehouse (query)** package.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Crosstab** and click **OK**.
4. From the **Source** tab, drag the following data items from **Sales and Marketing (query)**, **Sales (query)** to the crosstab:
   • from the **Product** folder, drag **Product line** as rows
   • from the **Time dimension** folder, drag **Year** as nested rows
   • from the **Order method** folder, drag **Order method** as columns
   • from the **Sales fact** folder, drag **Revenue** as the measure
5. Right-click the crosstab and click **Go to Query**.

6. In the **Properties** pane, set the **Override Dimension Info** property to **Yes**. The **Dimension Info** tab appears at the bottom of the work area.

7. Click the **Dimension Info** tab.

8. From the **Source** tab, drag the following items to the **Dimensions** pane:
   - **Product line**
   - **Year**

   **Product line** and **Year** become separate dimensions in the query.

9. Pause the pointer over the page explorer button and click **Page1**.

10. Click **Year**.

11. In the **Properties** pane, double-click the **Sorting** property.

12. In the **Data Items** box, drag **Year** to the **Sort List** box.

13. Run the report.

**Results**

All order years appear for all product lines, even if no revenue was produced.

<table>
<thead>
<tr>
<th>Revenue</th>
<th>Telephone</th>
<th>Fax</th>
<th>Mol</th>
<th>Special</th>
<th>Web</th>
<th>E-mail</th>
<th>Sales visit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Camping Equipment</strong></td>
<td>2010</td>
<td>80,467,598.88</td>
<td>9,034,763.39</td>
<td>10,015,621.15</td>
<td>5,644,211.12</td>
<td>125,828,519.92</td>
<td>38,124,634.73</td>
</tr>
<tr>
<td></td>
<td>2011</td>
<td>47,502,256.31</td>
<td>6,228,274.27</td>
<td>8,340,759.90</td>
<td>5,012,666.47</td>
<td>270,405,415.86</td>
<td>21,291,065.31</td>
</tr>
<tr>
<td></td>
<td>2012</td>
<td>17,715,451.4</td>
<td>5,226,451.57</td>
<td>2,482,258.28</td>
<td>332,065.44</td>
<td>426,353,675.75</td>
<td>10,612,304.02</td>
</tr>
<tr>
<td></td>
<td>2013</td>
<td>8,149,587.54</td>
<td>1,064,909.25</td>
<td>3,906.72</td>
<td>1,190,606.41</td>
<td>311,192,071.84</td>
<td>4,871,150.57</td>
</tr>
<tr>
<td><strong>Outdoor Protection</strong></td>
<td>2010</td>
<td>8,141,169.76</td>
<td>1,435,512.2</td>
<td>1,492,539.23</td>
<td>856,775.41</td>
<td>13,735,716.85</td>
<td>4,135,915.36</td>
</tr>
<tr>
<td></td>
<td>2011</td>
<td>3,230,287.7</td>
<td>365,329.2</td>
<td>594,355.43</td>
<td>267,366.66</td>
<td>16,479,270.86</td>
<td>1,406,331.47</td>
</tr>
<tr>
<td></td>
<td>2012</td>
<td>507,485.83</td>
<td>123,028.45</td>
<td>101,500.05</td>
<td>16,790.96</td>
<td>5,570,078.91</td>
<td>289,343.01</td>
</tr>
<tr>
<td></td>
<td>2013</td>
<td>76,371.43</td>
<td>22,814.84</td>
<td>2,186,745.33</td>
<td>56,498.04</td>
<td>154,605.82</td>
<td></td>
</tr>
<tr>
<td><strong>Personal Accessories</strong></td>
<td>2010</td>
<td>45,940,892.79</td>
<td>11,312,266.47</td>
<td>2,952,291.35</td>
<td>3,467,802.5</td>
<td>284,822,828.47</td>
<td>22,819,705.21</td>
</tr>
<tr>
<td></td>
<td>2011</td>
<td>18,428,595.15</td>
<td>3,612,228.75</td>
<td>1,719,745.82</td>
<td>1,390,361.2</td>
<td>411,577,877.16</td>
<td>10,253,053.91</td>
</tr>
<tr>
<td></td>
<td>2012</td>
<td>5,979,547.46</td>
<td>2,149,819.49</td>
<td>1,346,338.05</td>
<td>1,465,195.56</td>
<td>560,698,077.03</td>
<td>5,668,561.15</td>
</tr>
<tr>
<td></td>
<td>2013</td>
<td>3,173,298.96</td>
<td>586,673.75</td>
<td>400,961.81</td>
<td>152,249.24</td>
<td>427,367,391.96</td>
<td>4,009,763.27</td>
</tr>
<tr>
<td><strong>Golf Equipment</strong></td>
<td>2010</td>
<td>44,244,120.93</td>
<td>6,255,935.08</td>
<td>8,390,407.78</td>
<td>3,765,129.72</td>
<td>49,503,491.41</td>
<td>29,322,537.92</td>
</tr>
<tr>
<td></td>
<td>2011</td>
<td>27,340,352.57</td>
<td>3,530,563.93</td>
<td>2,355,735.76</td>
<td>1,296,633.25</td>
<td>16,139,894.38</td>
<td>8,551,232.61</td>
</tr>
<tr>
<td></td>
<td>2012</td>
<td>6,411,233.84</td>
<td>2,408,222.14</td>
<td>1,041,144.7</td>
<td>203,385,896.81</td>
<td>5,901,733.78</td>
<td>10,562,630.68</td>
</tr>
<tr>
<td></td>
<td>2013</td>
<td>734,405.51</td>
<td>3,037,587.46</td>
<td>157,498,057.23</td>
<td>4,358,426.85</td>
<td>8,012,340.24</td>
<td></td>
</tr>
<tr>
<td><strong>Mountaineering Equipment</strong></td>
<td>2010</td>
<td>10,620,292.36</td>
<td>6,129,791.95</td>
<td>2,497,183.28</td>
<td>2,636,822.56</td>
<td>65,855,489.46</td>
<td>2,517,063.13</td>
</tr>
<tr>
<td></td>
<td>2011</td>
<td>5,566,124.07</td>
<td>3,500,047.35</td>
<td>1,034,475.38</td>
<td>517,010.05</td>
<td>132,756,442.87</td>
<td>1,029,100.01</td>
</tr>
<tr>
<td></td>
<td>2012</td>
<td>5,698,610.37</td>
<td>2,186,530.85</td>
<td>316,175.52</td>
<td>3,130,285.22</td>
<td>13,182,967.61</td>
<td></td>
</tr>
</tbody>
</table>

**Create a Master Detail Relationship**

Create a master detail relationship to deliver information that would otherwise require two or more reports. For example, you can combine a list with a chart. The list can contain product lines and the chart can show details for each product line.

Master detail relationships must appear in nested frames to produce the correct results. You can create a master detail relationship in two ways:
• Use a parent frame for the master query and a nested frame for the detail query.
• Associate a report page with the master query and use a data container, such as
  a list or crosstab, for the detail query.

You cannot display parent data in the child frame or child data in the parent
frame. Also, you cannot perform calculations across master detail queries.

When you are working with dimensional data, IBM Cognos Report Studio does
not support master detail relationships that use a custom group.

You can use a master detail relationship to show data from separate data sources
in a single report. However, the data sources must be contained in the same
package.

If you are working with an SAP BW data source, you cannot use a data item from
the master query that contains non-ASCII values.

**Before you begin**

To create a master detail relationship using queries that reference two different
dimensional data sources, you must first [link the members] by aliasing the levels
that contain the members.

**Procedure**

1. To use a parent frame for the master query and a nested frame for the detail
   query:
   • From the Toolbox tab, drag a List, Repeater Table, or Repeater to the
     report.
   • Add a second data container to the object you inserted.
     You can insert a list, crosstab, chart, repeater table, or repeater into a list.
     You can add a list to a repeater table or repeater.
   • Add data items to both data containers.
2. To associate a report page with the master query and use a data container for
   the detail query:
   • Click anywhere in the report page.
   • In the Properties pane, click the select ancestor button and click Page.
   • Set the Query property.
   • From the Toolbox tab, drag a data container to the report.
3. To link a data item in the master query to a parameter in the detail query
   instead of to another data item, create the parameter in the detail query.
   Use parameters to filter values at a lower level in the detail query.
   • Pause the pointer over the query explorer button and click the detail
     query.
   • From the Toolbox tab, drag the Filter object to the Detail Filters box.
   • In the Expression Definition box, create the parameter.
4. Pause the pointer over the page explorer button and click the report page.
5. Click anywhere in the report page.
6. In the Properties pane, click the select ancestor button and click Page.
7. Click the data container containing the details.
8. From the Data menu, click Master Detail Relationships.
9. Click New Link.
10. In the Master box, click the data item that will provide the primary information.
11. To link the master query to a data item, in the Detail box, click the data item that will provide the detailed information.
12. To link the master query to a parameter, in the Parameters box, click the parameter that will provide the detailed information.

Tip: To delete a link, select the link and press the Delete key.
13. If your detail query object is a chart, you can disconnect a chart title from the master query:
   • Click the chart title, and then, from the Data menu, click Master Detail Relationships.
   • Clear the Use the master detail relationships from the chart check box.
14. If your master detail relationship includes a chart as the detail and you are using automatic axis ranges, you can set the same axis range for all the detail charts:
   • Select the axis.
   • In the Properties pane, under General, double-click the Axis Range property.
   • Under Maximum and Minimum, click Automatic.
   • Under Master detail reports, select the Same range for all instances of chart check box.
   For more information about axis ranges, see “Change the Axis Scale of a Current Default Chart” on page 120.

Tip: To avoid seeing the same data item twice in the report, click the data item in the data container driven by the detail query and click the cut button. This removes the item from the report display but keeps it in the query.

Related concepts:
“Master Detail or Burst Reports with Charts or Crosstabs May Result in Denial of Service” on page 499
When running a master detail or burst report that includes a chart or crosstab, disk space exhaustion may cause the report or other requests to fail. A large set of burst keys or master rows may produce one or more charts per detail, resulting in many master detail executions. This may cause the temp folder to accumulate many gigabytes of temporary files containing data required for successful chart rendering.

Link Members from Two Data Sources
If you create a master detail relationship using queries that reference two different dimensional data sources, you may need to create a relationship between levels with the same member captions but different Member Unique Names (MUNs).

For more information about MUNs, see “Recommendation - Use Member Unique Name (MUN) Aliases” on page 547.
For example, you may want to link the Americas member in the Sales territory level in the Sample Outdoors cube with Americas in the GO Subsidiary level in the Sample Outdoors Finance cube. To create the relationship, you must alias the levels that contain the members to link.

**Procedure**

1. In the parent frame or the report page, double-click the level that contains the member that will provide the primary information.
   The Data Item Expression dialog box appears.
2. In the Expression Definition box, use the expression in the following function:
   
   \[ \text{caption (expression)} \]
   
   For example, caption ( [great_outdoors_company].[Sales Territory].[Sales Territory].[Sales territory] )
3. Repeat steps 1 to 2 for the level in the data container that contains the details.
   A caption alias is created for each level. You can now use the caption alias for each level to create a master detail relationship using member captions, which are the same, instead of MUNs, which are different.

**Results**

You can now create the master detail relationship between the two levels.

**Working with Queries in SQL or MDX**

For each query in a report, you can work with the SQL or MDX that is executed when you run a report.

**Note:** MDX is not available with DMR data sources.

**View the SQL or MDX for an Entire Report or a Query**

View the SQL or MDX to see what is passed to the database when you run a report.

Only SQL is available when using DMR data sources because these data sources were relational.

**Procedure**

1. To view the SQL or MDX for the entire report, from the Tools menu, click Show Generated SQL/MDX.
   This option shows the SQL or MDX that will be executed in the data source. The SQL or MDX is organized by query and by query result. If a query is used in more than one data container, a query result is generated for each data container.

2. To view the SQL or MDX for a specific query, do the following:

   - Pause the pointer over the query explorer button and click the query.
   - In the Properties pane, double-click the Generated SQL/MDX property.

   The Generated SQL/MDX property shows the SQL or MDX that is executed when you view tabular data (from the Run menu, click View Tabular Data). Tabular data shows the data that is produced by the query in the form of a list. You can use this property to help you build advanced queries.
Results

The SQL or MDX for the query appears in the Generated SQL/MDX dialog box. For SQL, you can choose to view native SQL, which is the SQL that is passed to the database when you execute the query, or Cognos SQL, which is a generic form of SQL that IBM Cognos Report Studio uses. Cognos SQL is converted to native SQL before the query is executed.

Related tasks:

“Showing prompt values in generated SQL or MDX” on page 326
You can show prompt values when viewing the generated SQL or MDX of a query or report.

Build a Report Using Your Own SQL or MDX

You can build a report by adding SQL or MDX from an external source, such as another report.

If you are working with MDX, you must be aware of the MDX syntax that IBM Cognos Report Studio supports.

If you edit the SQL of a query, you must change the Processing property for that query to Limited Local.

You cannot build a report by adding your own MDX if the data source is IBM Cognos TM1®. IBM Cognos TM1 does not support custom MDX.

Procedure

1. From the File menu, click New.
2. Click Blank.
3. Pause the pointer over the query explorer button and click Queries.
4. From the Toolbox tab, drag Query to the work area.
5. From the Toolbox tab, do one of the following:
   • To build an SQL query, drag SQL to the right of the query.
   • To build an MDX query, drag MDX to the right of the query.

   Tip: You can drag SQL or MDX anywhere in the work area, and Report Studio will automatically create a query.
6. In the Properties pane, double-click the Data Source property and click a data source.
7. If required, set the Catalog property to the name of the catalog.
8. Double-click the SQL or MDX property and type the SQL or MDX.
9. Click Validate to check for errors.
10. If you are working with SQL, in the Properties pane, set the Processing property to Limited Local.
11. Double-click the query.
   If the SQL or MDX is valid, the data items defined in the SQL or MDX appear in the Data Items pane.
12. Pause the pointer over the page explorer button and click a report page.
13. From the Toolbox tab, drag an object to the work area.
   For example, drag a list, crosstab, chart, or repeater.
14. Click the data container.

15. In the Properties pane, click the select ancestor button and click the container you just created.
   For example, if you created a list, click List.
   
   Tip: You can also click the container selector (three orange dots) of the container to select it.

16. Set the Query property to the query for the report type.

17. From the Data Items tab, drag the items from the query you chose in the previous step to the data container.

Convert a Query to SQL
You can convert a query to SQL to edit it. You may want to do this to improve performance or to use SQL features that are not supported directly by IBM Cognos Report Studio.

Converting a query to SQL is an irreversible process.

Procedure

1. Pause the pointer over the query explorer button and click the query.
2. In the Properties pane, double-click the Generated SQL property.
3. Click Convert.
4. Click Validate to check for errors.
5. In the Properties pane, set the Processing property to Limited Local.

Edit the SQL or MDX
You can edit the SQL or MDX for a query that has been created as an SQL or MDX query or that has been converted to SQL.

Procedure

1. Pause the pointer over the query explorer button and click the SQL or MDX item under the query.
2. Double-click the SQL or MDX item.
3. Make changes in the text box.
   If you are working with MDX, you must be aware of the MDX syntax that IBM Cognos Report Studio supports.
4. Click Validate to check for errors.

First-Rows Optimization
The SQL produced by IBM Cognos Report Studio depends on the report format you choose. For example, if you specify HTML format, first-rows optimization is requested. All-rows is requested if you specify PDF.

It is important for database administrators and programmers to remember that Report Studio does not always use first-rows optimization. If you assume first-rows optimization is always requested, this can cause the RDBMS optimizer to process the query differently than you expect.
Supported MDX Syntax

If you are working with MDX, you must be aware of the syntax that Report Studio supports. IBM Cognos Report Studio supports the MDX grammar as specified in Microsoft Data Access Software Developer Kit version 2.8.

Report Studio supports the following MDX functions. This is a partial list and applies only when you are working with the following data sources:

- cubes
- SAP BW
- IBM Cognos TM1
- DB2® OLAP
- IBM Cognos Consolidation
- IBM Cognos Contributor

For these data sources, IBM Cognos Business Intelligence uses a variation of the Microsoft SQL Server Analysis Services syntax.

- Acos
- Aggregate
- Ancestor
- Ancestors
- Asin
- Atan
- Avg
- BottomCount
- BottomPercent
- BottomSum
- Children
- ClosingPeriod
- CoalesceEmpty
- Convert
- Cos
- Cosh
- Count
- Cousin
- Crossjoin
- CurrentMember
- DefaultMember
- Descendants
- Dimension
- Distinct
- Except
- Exp
- Filter
- FirstChild
- FirstSibling
- Generate
- Head
Hierarchize
Hierarchy
IIf
Intersect
Is
IsAncestor
IsEmpty
Item
Lag
LastChild
LastPeriods
LastSibling
Lead
Level
Levels
LinkMember
Ln
Log10
Max
Median
Members
Min
Mtd
Name
NextMember
OpeningPeriod
Order
Ordinal
ParallelPeriod
Parent
PeriodsToDate
PrevMember
PreviousMember
Properties
Qtd
Rank
Siblings
Sin
Sinh
Stddev
Subset
Sum
Tail
Tan
Tanh
Example - Create a Dynamic Report That Shows Period-to-date Data

When you are working with a dimensional data source, you can use dimensional functions to retrieve data for a specific period-to-date.

In this topic, you learn how to create a dynamic report that retrieves year-to-date revenue for each product line. The report also shows the percentage of the revenue that is generated in the month users select when they run the report.

It should take 15-20 minutes to complete this topic, and your report will look like this.

![Figure 45. Crosstab report showing year-to-date revenue for product lines](image)

**Procedure**

1. Create a crosstab report that uses the Go Data Warehouse (analysis) package.
2. Add the following data items to the report:
   - From **Product**, drag **Product line** to the **Rows** drop zone.
   - From **Sales fact**, drag **Revenue** to the **Measures** drop zone.

   **Tip:** Use the **Source** tab.

3. Create the following query calculation named **Selected Month** in the **Columns** drop zone:
   
   \[
   [\text{Sales}].[\text{Time}].[\text{Time}].[\text{Month}] \rightarrow \text{?Month}?
   \]

4. Create this query calculation named **Year to Date Set** next to the **Selected Month** column:
   
   \[
   \text{periodsToDate}([\text{Sales}].[\text{Time}].[\text{Time}].[\text{Year}],[\text{Selected Month}])
   \]
Tip: Information about the periodsToDate function is displayed in the Information pane when you click the function in the Functions tab. The periodsToDate function is in the M-Q folder in the Dimensional Functions folder.

5. Click the Year to Date Set column and click the cut icon.
6. Create the following query calculation named YTD Revenue next to the Selected Month column:
   \[
   \text{total (currentMeasure within set [Year to Date Set])}
   \]
7. Create the following query calculation named Percent of YTD Revenue next to the YTD Revenue column:
   \[
   [\text{Selected Month}]/[\text{YTD Revenue}]
   \]
8. Click the Unlock icon to unlock the report.
9. Add this text item to the left of the text in the YTD Revenue column, and add a blank space after the text:
   YTD Revenue:
10. Set the font style of the text to italic.
11. In the YTD Revenue column, click \(<\#\text{YTD Revenue}\#>\) and change the Source Type property to Report Expression.
12. Double-click the Report Expression property and drag the Month parameter to the Expression Definition box.

   Tip: You can find the Month parameter in the Parameters tab.
13. Click the lock icon to lock the report.
14. Right-click the Percent of YTD Revenue column title and click Select Member Fact Cells.
15. In the Properties pane, double-click the Data Format property.
16. Under Format type, click Percent.
17. Set the No. of Decimals property to 2.
18. Run the report to view it.

Results

Users are prompted to select a month. When they click OK, the report shows year-to-date revenue for each product line and the percentage of revenue that is generated for the selected month.

For more information, see the following references:

• Creating Report Templates
• Add Relational Data to a Report or Add Dimensional Data to a Report
• Using Dimensional Calculations
• Format Crosstabs
• Specify the Font
• Format Data
Using Dimensional Calculations

Insert a calculation to make your report more meaningful by deriving additional information from the data source. For example, you create an invoice report, and you want to see the total sale amount for each product ordered. Create a calculated column that multiplies the product price by the quantity ordered.

You build calculations in the expression editor using functions.

If a calculation is used in multiple reports, or by different report authors, ask your modeler to create the calculation as a standalone object in the model and include it in the relevant package.

Related concepts:
- “Considerations when Creating Calculations” on page 537

Consider the following information and limitations when creating dimensional calculations.

Create a Simple Member Calculation

When working with dimensional data sources, you can select members in your report and create simple calculations.

In addition to simple arithmetic calculations, you can perform the following calculations:

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rollup</td>
<td>Summarizes, or rolls up, all the values of the selected items in a hierarchy.</td>
</tr>
<tr>
<td>%</td>
<td>Calculates the value of a selected item as a percentage of another item.</td>
</tr>
<tr>
<td>% Difference</td>
<td>Calculates the difference between two items as a percentage.</td>
</tr>
<tr>
<td>% Of Base</td>
<td>This calculation is only available if you select two members from different hierarchies, each from a different edge. This calculation takes the first selected member from edge A and the second selected member from edge B. The calculation result of a percent of base should be such that all the values for member A will compare itself to the intersect value of member A and B.</td>
</tr>
</tbody>
</table>

For example, you have a crosstab showing quantity sold by year by product line. You select 2012 and Camping Equipment, right-click the selection, click Calculate, and then click % of Base (2012, Camping Equipment). The calculated row shows the percentage value of each product line for 2012 as compared to Camping Equipment for 2012.
Calculation Description

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom</td>
<td>Allows you to specify your own value when performing a simple arithmetic calculation. Also allows you to change the order of operands or type a custom name for the new calculated row or column.</td>
</tr>
</tbody>
</table>

**Procedure**

1. Select the items in your report to calculate.

2. Click the insert calculation button and select the calculation to perform.

   **Tip:** Calculations that are not applicable to the items you selected are greyed out.

3. To change the order of the operands or the name of the calculated item added to the report, click **Custom**.

**Results**

The calculation appears as a new row or column in your report.

**Insert a Query Calculation**

When working with dimensional data, insert a query calculation into your report to add a new row or column with values that are based on a calculation. For example, you create a query calculation named Euros that converts dollars to euros by multiplying an existing dollar measure by a conversion rate. Euros can then be displayed to end users in a separate row or column.

In IBM Cognos Report Studio, you can create the following types of calculations:

- Create calculated members or calculated measures where the expression is a member or a tuple-based (as opposed to property-based) value expression. For example, `[2013] - [2012]` is a calculated member and `[Revenue] - [Planned Revenue]` is a calculated measure.

- Use set expressions to create calculated sets of members. For example, `children([2012])` is a set expression that displays the child members of 2012.

- You must base each set expression on a hierarchy and the set expression must contain only members from this hierarchy.

- Create value expressions when you want to insert a string, number, date, or interval value.

You can also create a calculation that uses an intersection (tuple) that you already defined.

If you are using a DB2 data source, the subtract operator is invalid if you combine the datatypes `timestamp2` and `packed decimal`.

When creating an expression that will be used in a double-byte environment, such as Japanese, the only special characters that will work are ASCII-7 and `~`.
Procedure

1. From the Toolbox tab, drag Query Calculation to the report.
2. In the Name box, type a meaningful name for your calculation.
   For example, if you are calculating the difference between 2012 revenue and
   2011 revenue, you could name your expression 2012 - 2011 Revenue.
3. Click the type of calculation you want to create and select the hierarchy that
   contains the data that is the focus of your calculation.
4. In the Available Components pane, define the calculation:
   • Specify how you want to view the available data items in your data package
     by clicking the view member tree button or the view package tree button.
   • To add data items that are not shown in the report, on the source tab,
     double-click the data items.
   • To add data items that are in the report but not necessarily in the model,
     such as calculations, on the data items tab, double-click the data items.
   • To add data items from a specific query, on the queries tab, double-click
     data items.
   • To add functions, summaries, and operators, on the functions tab,
     double-click elements.
   Tip: To filter the visible functions, summaries, and operators in the
   Available Components pane, click the filter button and filter by the
   function type, what the function returns, or what the function acts on.
   • To add a value that is derived from a parameter, on the parameters tab,
     double-click a parameter.
     Parameters define prompts, drill-through reports, and master detail
     relationships.
   • To add a macro, on the macros tab, drag the parameter maps, session
     parameters, or functions you want to add to the macro.
   Tip: Use Insert macro block to get you started with creating the macro
     expression. Insert macro block inserts number signs in the expression. Make
     sure that the macro expression you create is between the number signs.
   Tips:
   • You can also type the calculation directly in the Expression Definition box.
   • When typing date values, ensure that the date format is correct for your
     database type.
   • To copy and paste expression components in the Expression Definition
     pane, you can use the copy button and the paste button.
     For information about specific functions, see Appendix F, “Using the expression
     editor,” on page 555.
5. Click the validate button. Any validation errors appear on the Errors tab of the Information pane.

Tips:
- To show or hide the Available Components pane, click the available components button.
- To show or hide the Information pane, click the information button.
- To copy and paste expression components in the Expression Definition pane, you can use the copy button and the paste button.

Related concepts:
Chapter 17, “Using Query Macros,” on page 403
A macro is a fragment of code that you can insert in the Select statement of a query or in an expression. For example, add a macro to insert a new data item containing the user's name.

Create a Layout Calculation
Create a layout calculation to add run-time information to your report, such as current date, current time, and user name.

When you create layout calculations, unsupported functions do not appear in the Functions tab of the expression editor. Specifically, there is no Summaries folder, and some operators, constants, and constructs are also unavailable. These functions are not available because only the database can perform them. Report expressions and calculations based on layout expressions are performed in IBM Cognos Report Studio.

To see the complete list of functions available in the expression editor, except for report functions, create a query calculation. All functions are available when you create a query calculation because these calculations are performed in the database and not in Report Studio.

Procedure
1. From the Toolbox tab, drag Layout Calculation to the report.
2. In the Available Components box, define the calculation:
   - To add data items that are not shown in the report, on the source tab, double-click data items.
   - To add data items from a specific query, on the queries tab, double-click data items.
   - To add functions, summaries, and operators, on the functions tab, double-click elements.
   - To add a value that is derived from a parameter, on the parameters tab, double-click a parameter.

Parameters define prompts, drill-through reports, and master detail relationships.

Tips:
You can also type the calculation directly in the **Expression Definition** box.

When typing date values, ensure that the date format is correct for your database type.

To copy and paste expression components in the **Expression Definition** pane, you can use the copy button and the paste button. For more information about creating expressions, see Appendix F, “Using the expression editor,” on page 555.

3. Click the validate button.

**Create an Intersection (Tuple)**

When working with dimensional data, an intersection, also known as a tuple, is useful for obtaining a value from the combination of two or more members that you specify. Each member must be from a different hierarchy. The intersection can include only one measure.

For example, the intersection (Revenue, 2012, Cooking Gear) shows the revenue value for the year 2012 and for the product line Cooking Gear.

Use tuples to avoid report summaries that do not reflect the visible data. For more information about this issue, see “Summaries in a report do not correspond to the visible members” on page 490.

**Procedure**

1. From the Toolbox tab, drag the Intersection (Tuple) object to the report.
2. From the Available members and measures pane, select items and click the right arrow to move them to the Intersection members and measures box. You can also use items from the Calculated Members and Measures tab.
3. To define the hierarchy for this intersection, click a parent object in the Intersection hierarchy box.

**Assigning a Hierarchy or Dimension**

You must assign each calculated member, set expression, and intersection (tuple) to a hierarchy or dimension. We recommend that you select the hierarchy or dimension upon which your calculation focuses. For example, if your calculated member is based on years, select the Time hierarchy. If you create a calculated measure, select the Measures dimension.

Select only a hierarchy that cannot affect the value of the calculation. For example, the hierarchy Camping Equipment has the same value no matter what Products context it appears in the report. Therefore,

- the calculation ([Camping Equipment] - [Mountaineering Equipment]) has a well-defined meaning only in the Products hierarchy, so select that hierarchy.
- the calculation tuple ([Revenue], [Camping Equipment]) can be assigned to either the Products or Measures hierarchy, as it is not affected by either context.
- the calculations ([2013] - [Camping Equipment]) and ([Fiscal Year 2013] - [Calendar Year 2013]) do not have a well-defined meaning in any hierarchy, and therefore cannot be used reliably at all.
Drilling Up and Drilling Down in Dimensional Reports

When working with dimensional or dimensionally-modeled relational (DMR) data sources, you can create reports that allow the reader to view more general or more detailed information on your data within a predefined dimensional hierarchy (such as Years - Year - Quarter - Month) without having to create different reports.

You can link groups of data items from different queries so that when you drill up or drill down in one query, the data item also drills up or drills down in the linked queries. Linking data items is useful if your report includes more than one query with the same data item. For example, if your report shows a list and a crosstab that each include the Product Line data item, you can link the Product Line data item from the crosstab query to the Product Line data item from the list query so that when you drill up in the crosstab Product Line data item, the list Product Line data item also drills up.

Member Sets

Member sets are used to group data items that are logically related for various actions, such as drill actions, zero suppression, and ranking. They can be defined as a flat list or as a tree structure in which member sets in the same parent chain are considered related.

For example, for drill operations, a member set defines the set of items that can potentially change when a given item in the set is drilled on. The values of other items in the query or even those in the same hierarchy are preserved when any item in this set is drilled on. Usually, a member set references items that have logical roles in a drill action, such as a detail, a summary of the detail, or a parent of the detail. A single data item can belong to only one member set.

If you do not define member sets for items, the IBM Cognos Business Intelligence server associates items into default member sets and behaviors using simple dimension rules on item expressions. You can override the behavior for a particular item while other items continue to use the default.

When you define a member set, you must explicitly define behaviors for each item in the set. Items in the set that have no behaviors have their values preserved.

Drill behaviors always act from a root member set. This means that when an item is drilled on, the root member of its member set is found and all items from the root down are processed. Although calculations and aggregates are not directly related by hierarchy, they respond because of their dependence on the items upon which they are based.

Create Member Sets

Create a member set when you want to define a non-default drill behavior. You specify what items respond to a drill action by adding them to the member set.

Procedure

1. Pause the pointer over the query explorer button and click the query in which to create a member set.
2. In the Properties pane, set the Define Member Sets property to Yes.
3. Double-click the query and click the Member Sets tab.
4. From the Source tab, drag items to the work area.
To define a member set as a tree structure, drag the item that will serve as the root item of the set to the work area, and then drag other items over the root item to create the tree structure.

**Note:** You can also nest member sets.

### Create a Drill-up and Drill-down Report

You can link groups of data items from different queries so that when you drill up or drill down in one query, the data item also drills up or drills down in the linked queries.

When you drill down or up, you can specify whether the column title shows the member caption value or the level label value. By default, titles use member captions. Using level labels may be more meaningful. For example, within the Country or Region column, you drill down on Australia. By default, the column title becomes Australia, the member caption. It may be more meaningful to use Country or Region, the level label, as the column title.

**Tip:** The Returned Items sample report in the GO Data Warehouse (analysis) package includes drilling. For more information about The Sample Outdoors Company samples, see [Appendix C, “Sample Reports and Packages,” on page 507.](#)

### Before you begin

Before you begin, ensure that you are using a dimensional data source.

### Procedure

1. From the Data menu, click **Drill Behavior**.
2. On the Basic tab, in the **Report output drill capabilities** box, select the **Allow drill-up and drill-down** check box.
   
   By default, IBM Cognos Report Studio determines which items can be drilled on based on the dimensional structure.

   **Tip:** To enable the report to be used as the source during a package drill-through, select the **Allow this report to be a package-based drill-through source** check box and click OK. For more information about package drill-through, see the [Administration and Security Guide](#).

3. To disable drill-up or drill-down for a data item, select the data item in either the **Disable drill-up for** box or the **Disable drill-down for** box.

4. From the Advanced tab, you can change the drill-up or drill-down behavior for any data item by selecting the data item and then choosing one of the following behaviors.

<table>
<thead>
<tr>
<th>Behavior name</th>
<th>Drill-up behavior</th>
<th>Drill-down behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preserve</td>
<td>The data item’s value will remain unchanged.</td>
<td>The data item’s value will remain unchanged.</td>
</tr>
<tr>
<td>Empty Set</td>
<td>The set of values associated with this data item is set to be the empty set (novalues). For crosstabs, the data item will effectively be removed from the report.</td>
<td>The data item will be removed from the report.</td>
</tr>
<tr>
<td>Behavior name</td>
<td>Drill-up behavior</td>
<td>Drill-down behavior</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Replace Item</td>
<td>The data item’s value will change to become the parent (if a lowest-level summary was drilled up on) or the grandparent (if a lowest-level detail of a dimension was drilled up on) of the item that was drilled on.</td>
<td>The data item’s value will change to become the item that was drilled on.</td>
</tr>
<tr>
<td>Replace Expression</td>
<td>The data item’s value will change to become the children of the parent (if a lowest-level summary was drilled up on) or the children of the grandparent (if a lowest-level detail of a dimension was drilled up on) of the item that was drilled on.</td>
<td>The data item’s value will change to become the children of the item that was drilled on.</td>
</tr>
<tr>
<td>Change Expression</td>
<td>The data item’s value will change to become the value of some other data item in the query.</td>
<td>The data item’s value will change to become the value of some other data item in the query.</td>
</tr>
<tr>
<td>Depth Based Expression</td>
<td>The data item’s value will change to become all data items n generations higher in the dimensional hierarchy than the item that was drilled on. The number of generations or levels is determined by the Depth value.</td>
<td>The data item’s value will change to become all data items n generations higher in the dimensional hierarchy than the item that was drilled on. The number of generations or levels is determined by the Depth value.</td>
</tr>
<tr>
<td>Ancestor</td>
<td>The data item’s value will change to become the data value of the ancestor n generations higher in the dimensional hierarchy than the item that was drilled on. The number of generations or levels is determined by the Depth value.</td>
<td>The data item’s value will change to become the data value of the ancestor n generations higher in the dimensional hierarchy than the item that was drilled on. The number of generations or levels is determined by the Depth value.</td>
</tr>
</tbody>
</table>

5. In the **Column title drill behavior** box, specify whether to use member captions or level labels as the columns titles for each data item.

6. From the **Data Items Linking** tab, you can link groups of data items:

   - Click the new button.
   - Type a name for this group of linked data items and click **OK**.
     To rename a group of linked data items, click the group, click the rename button, type a name, and click **OK**.
   - Use the arrows to move data items from the queries on the left to the groups on the right.

**Note:** You can only add a data item to one group.
Results

The report will generate links for any item that can be drilled up or down on.

When you run a report, you can drill down or drill up by right-clicking the data item and choosing the action from the context menu. The menu items will be disabled if an item cannot be drilled up or down on.

Related tasks:

“Create a Drill-up and Drill-down Chart” on page 153

If you use a dimensionally-modeled data source, you can create a chart that allows you to drill down to lower-level data or drill up to higher-level data.

Recommendation - Drilling Down in Very Large Data Sources

When you drill down in a report that uses a very large data sources query results can be much larger than what can be held in memory. For example, if you insert the lowest level of four hierarchies that each contain 1000 members, the report can contain 1000,000,000,000 cells. At best, this query will take a very long time to run. While this large query executes, all other queries for the same server process will likely be blocked. For most data sources, the query will likely fail due to insufficient memory or timing out. Then the memory is released and normal services resume.

However, when using a Microsoft SQL Server 2005 Analysis Services (SSAS) cube, the memory consumed by the SSAS client is not always released in a way that other queries can effectively re-use. The server continues to run with insufficient memory, causing new queries to either run very slowly or fail completely. You may encounter an error, and to continue, the system administrator must manually stop and restart the IBM Cognos BI service.

To avoid these problems, consider the size of hierarchy levels and sets when creating reports and avoid combining them in ways that create large queries. To determine the size of a set, create and run a very small report that includes only a count of the members within that set. You can also use filtering techniques to focus your data.

Example - Create a Dashboard Report

Create a dashboard report to show a high-level graphical view of company performance indicators.

By using dashboard reports, users can

- drill up and drill down to see higher and lower levels of detail
- navigate to other targets, including IBM Cognos Report Studio reports, IBM Cognos Query Studio reports, and IBM Cognos Analysis Studio analyses
- include multiple charts derived from different data sources in a single report

Dashboard reports are not the same as workspaces or dashboards created in IBM Cognos Workspace. Dashboard reports display several different report objects in one report, but the information in the report is static. Workspaces created in Cognos Workspace are a collection of IBM Cognos content that you assemble into an interactive workspace. Content is displayed in widgets. For information about creating interactive and sophisticated workspaces, see the IBM Cognos Workspace User Guide. You can also create dashboards within IBM Cognos Connection. In these dashboards, content is displayed in portlets within pages. For more information, see the IBM Cognos Connection User Guide.
In this topic, you learn how to create a dashboard report so that users can examine important sales data in a single view.

It should take 15 to 20 minutes to complete this topic, and your dashboard report will look like this.

Procedure
1. Create a blank report with the GO Data Warehouse (analysis) sample package.
2. Add a column chart, a bullet chart, and a crosstab.
3. Drag the following items to the bullet chart:
   - Revenue (in Sales and Marketing (analysis), Sales, Sales fact) to the Bullet Measure drop zone
   - Planned Revenue to the Target Measure drop zone
   - Product line level (in Products) to the Series (matrix rows) drop zone
   - Region level (in Retailers) to the Categories (matrix columns) drop zone
4. Drag the following items to the column chart:
   - Revenue (in Sales fact) to the Default measure (y-axis) drop zone
   - Product line level (in Product) to the Categories (x-axis) drop zone
   - Retailers level (in Retailers) to the Series (primary axis) drop zone
5. Drag the following items to the crosstab:
   - **Gross margin** (in Sales fact) to the *Measures* drop zone
   - **Product line** level (in Product) to the *Rows* drop zone
   - **Region** level (in Retailer site) to the *Columns* drop zone

6. Click the bullet chart.

7. Set the **Size & Overflow** property width to 300 px and height to 300 px.

8. Set the **Title** property to Show.

9. Double-click the chart title and type the following text:
   **Gross Profit for Product Lines by Region**

10. Set the **Font** property for the chart title to Arial, 12pt, Bold.

11. Set the **Padding** property for the left and bottom padding to 20 px.

12. In the **Bullet Indicators** property, set the bar width to 30%.

13. Click the chart, click the palette button, and click **Contemporary** from the available palettes.

14. Set the **Size & Overflow** property width to 300 px.

15. Set the **Title** property to Show.

16. Double-click the chart title and type the following text:
   **Product Lines: Revenue by Retailer**

17. Set the **Font** property for the chart title to Arial, 12pt, Bold.

18. From the **Data** menu, click **Drill Behavior**.

19. Select the **Allow drill-up and drill-down** check box.

20. Run the dashboard report to view what it will look like for your users.

### Results

Users can drill up or drill down on values in the report to view related information.

For more information, see the following references:

- Creating Report Templates
- Add Relational Data to a Report or Add Dimensional Data to a Report
- “Customize a Current Default Gauge Chart” on page 144
- Customize a Legacy Gauge Chart
- Specify the Height and Width
- Specify the Font
- Create a Drill-up and Drill-down Report
Chapter 11. Adding Prompts to Filter Data

You can add prompts to a report to add interactivity for users. Prompts act as questions that help users to customize the information in a report to suit their own needs. For example, you create a prompt so that users can select a product type. Only products belonging to the selected product type are retrieved and shown in the report.

Prompts are composed of three interrelated components: parameters, prompt controls, and parameter values. Parameters are based on parameterized filters and form the questions to ask users. Prompt controls provide the user interface in which the questions are asked. Parameter values provide the answers to the questions.

IBM Cognos Report Studio provides several ways to create prompts. You can
• use the Build Prompt Page tool
• build your own prompt and prompt page
• create a parameter to produce a prompt
• insert prompts directly into the report page
• in dimensional reporting, you can also define prompts using context filters

If you include reports from different packages in an interactive dashboard that uses global filters, ensure that you use the same parameter name for the prompt in all the reports. For more information about creating interactive dashboards in IBM Cognos Connection, see the IBM Cognos Business Intelligence Administration and Security Guide.

You can also create prompts in the package. For more information, see the IBM Cognos Framework Manager User Guide.

You can also use context filters, also known as slicer filters, to focus the data in your report.

Syntax of Prompt Expressions

Prompt expressions use the following syntax, where p represents the parameter name.

You can also use these expressions to create parameterized data items that you can then use in master detail relationships.

The type of expression you use depends on whether you have chosen relational or dimensional reporting style. For more information about reporting styles, see "Relational and dimensional reporting styles" on page 22.

<table>
<thead>
<tr>
<th>Relational: Detail Filter Expressions</th>
<th>Dimensional: Context (Slicer) or Edge Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>[data item] = ?p?</td>
<td>[level or hierarchy]-&gt;?p?</td>
</tr>
<tr>
<td>for single-select prompts and master detail relationships</td>
<td>For single-select member prompts and master detail relationships</td>
</tr>
</tbody>
</table>
Relational: Detail Filter Expressions | Dimensional: Context (Slicer) or Edge Expressions
--- | ---
[data item] in ?p? | set([level or hierarchy]->?p?)
for multi-select prompts | For multi-select member prompts
filter [set], [property expression] = ?p? ) | For single-select value prompts and master detail relationships
A property expression is an expression that is assigned to a property to enable dynamic update of the property when the report runs.
filter [set], [property expression] IN ?p? ) | For multi-select value prompts
A property expression is an expression that is assigned to a property to enable dynamic update of the property when the report runs.

Related concepts:
“Prompt control labels” on page 472
You can associate labels to prompt controls to make the prompts accessible.

Related reference:
Appendix H, “Prompt API for IBM Cognos BI,” on page 975
The JavaScript Prompt API provides report authors with a method of customizing prompt interaction in the reports they author.

Use the Build Prompt Page Tool

Use the Build Prompt Page tool to quickly add filters and prompts to a report. This tool creates filters in the query using the prompt information properties that are set in the IBM Cognos Framework Manager model. For more information about the prompt information properties, see the Framework Manager User Guide.

Note: The Promotion Success sample report in the GO Data Warehouse (analysis) package includes a prompt page. The Historical Revenue sample report in the Sales and Marketing (cube) package and the Eyewear Revenue by Brand and Size sample report in the GO Data Warehouse (analysis) package include multiple prompts. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Before you begin

Note: The Build Prompt Page tool creates a static report specification, so if the prompt information properties are updated in the Framework Manager model, those updates will not be reflected in your report.

Procedure
1. Click the column or row on which users will be prompted.
   - To create multiple prompts or a [cascading prompt] click more than one column or row.
2. From the Tools menu, click Build Prompt Page.
   A prompt page is created that has
   - a page header
   - a prompt control for each selected column
   - a page footer containing Cancel, Back, Next, and Finish buttons

Build Your Own Prompt and Prompt Page

Create your own prompt and prompt page to control how they appear in a report.

Note: The Promotion Success sample report in the GO Data Warehouse (analysis) package includes a prompt page. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Procedure

1. Pause the pointer over the page explorer button and click Prompt Pages.
2. From the Toolbox tab, drag Page to the Prompt Pages box.
3. Double-click the page you just created.
4. From the Toolbox tab, drag one of the following prompt controls to the prompt page.

<table>
<thead>
<tr>
<th>Prompt control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Box Prompt</td>
<td>Retrieves data based on a value that users type. Use this control when users know exactly what value they want to enter, such as a name or account number.</td>
</tr>
<tr>
<td>Value Prompt</td>
<td>Retrieves data based on values that users select from a list. Use this control to show a list of possible values from which users can choose. Note: The maximum number of items that can appear in a list is 5000.</td>
</tr>
<tr>
<td>Select &amp; Search Prompt</td>
<td>Retrieves values based on search criteria that users specify. Data is then retrieved based on values users select from the search results. Use this control instead of a value prompt if the list of values is very long, which can slow down performance. Tip: Users have the option of performing a case sensitive or case insensitive search. A case sensitive search is faster, while a case insensitive search usually returns more values. You cannot use this control if you are working with SAP BW data sources.</td>
</tr>
<tr>
<td>Prompt control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Date Prompt</td>
<td>Retrieves data based on a date that users select. Use this control to filter a date column. Users can retrieve data for a specific day, a set of days, or a range of days.</td>
</tr>
<tr>
<td>Time Prompt</td>
<td>Retrieves data based on a time that users select. Use this control to restrict a report to a particular time or time range. For example, users can see how many orders are received after business hours. Users can then use this information to determine the number of staff needed to work after hours.</td>
</tr>
<tr>
<td>Date &amp; Time Prompt</td>
<td>Retrieves data based on a date and time that users select. Use this control to filter a datetime or timestamp column. This control is useful for specifying ranges. For example, users can retrieve all orders received from Monday at 12:00 a.m. to Friday at 5:00 p.m.</td>
</tr>
<tr>
<td>Interval Prompt</td>
<td>Retrieves data based on a time interval that users specify. Use this control to retrieve data that is related to the passage of time. For example, users can retrieve a list of products that were returned 30 or more days after they were purchased.</td>
</tr>
<tr>
<td>Tree Prompt</td>
<td>Retrieves data based on values that users select from a list. Values are organized hierarchically. This control is useful when you are working with dimensional data sources. Data is shown from the top of a dimension hierarchy to the most detailed member, and users can choose the level of detail they want to view in the report. For more information about tree prompts, see “Control the Data That Appears in a Tree Prompt” on page 326.</td>
</tr>
<tr>
<td>Generated Prompt</td>
<td>Selects a prompt control based on the data type of the data item. This control acts like a placeholder. When users run the report, the control is replaced by the appropriate prompt control. For example, if users are prompted for date values, the control is replaced by a date &amp; time prompt.</td>
</tr>
</tbody>
</table>
The **Prompt Wizard** dialog box appears.

5. If you are creating a text box, date, time, date and time, interval, or generated prompt, do the following:
   - Create a new parameter for the prompt or use an existing parameter.
   - Click **Next**.
   - If you created a new parameter, define the expression by selecting a data item from the package and the operator to use.

   **Tip:** Make the prompt optional by selecting the **Make the filter optional** check box.
   - Go to step 7.

6. If you are creating a value, select & search, or tree prompt, do the following:
   - Create a new parameter for the prompt or use an existing parameter.
   - Click **Next**.
   - If you created a new parameter and you want to use the parameter to filter data, select the **Create a parameterized filter** check box and define the expression by selecting a data item from the package and the operator to use.

   If you are creating a tree prompt, you must choose **in** in the **Operator** box.

   You can also use a parameter to provide a value for a layout calculation, such as showing a user’s name in the report. When the report is run, you can use a parameter to prompt the user to type his name and have it appear in the report.

   **Tip:** Make the prompt optional by selecting the **Make the filter optional** check box.
   - Click **Next**.
   - If you created a parameterized filter and you have more than one query defined in the report, select the check box for the query on which to filter and click **Next**.
   - Select the **Create new query** check box to create the query that will be used to build the list of data values shown when the report is run.

   **Tip:** Do not create a new query if you want to link the prompt to an existing query or if you intend to create the query at a later time.
   - Click the ellipsis (...) button beside **Values to use** and click the data item on which to prompt.
   - To choose a data item that is different than what users will see when they are prompted, click the ellipsis (...) button beside **Values to display** and click the data item.
   - To create a **cascading prompt** in the **Cascading source** box, click the parameter that represents the cascade source.

7. Click **Finish**.

**Results**

The prompt control is added to the prompt page. A prompt page is like a report page. You can insert graphics and text and apply formatting.

You can also modify the properties of the prompt control by clicking it and making changes in the **Properties** pane.
Example - Create a Report Showing Products Shipped for a Specific Time Interval

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a report that shows the time interval between closing and shipping dates.

Procedure

1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click List and click OK.
4. From the Source tab, expand Sales and Marketing (query) and Sales (query).
5. Expand Sales fact and double-click Quantity.
6. Expand Products and double-click Product type.
7. Expand Sales order and double-click Order number.
8. Create a time interval calculation:
   - From the Toolbox tab, drag Query Calculation to the right of Quantity in the list.
   - In the Name box, type the following and click OK:
     Time Interval
   - In the Expression Definition box, type the following and click OK:
     \([\text{Sales (query)}].[\text{Time (close date)}].[\text{Date (close date)}]-[\text{Sales (query)}].[\text{Time (ship date)}].[\text{Date (ship date)}]\)
9. Create a filter:
   - From the Data menu, click Filters.
   - Click the add button.
   - In the Expression Definition box, from the Data Items tab, double-click Time Interval calculation you created.
     - Add > ?p1? after Time Interval to create the following expression:
       \([\text{Time Interval}] > ?p1?\)
   - Click OK twice.
   A parameterized filter is created that will return data when the difference between the return date and the order date is greater than the value specified by the user.
10. Create a prompt page:
    - Pause the pointer over the page explorer button and click Prompt Pages.
    - From the Toolbox tab, drag Page to the Prompt Pages pane, and then double-click it.
    - From the Toolbox tab, drag Interval Prompt to the work area. The Prompt Wizard dialog box appears.
    - Click Use existing parameter, and then click p1.
• Click **Finish**.

11. Run the report.

   An interval prompt appears.

12. In the **Days** box, type a value and click **Finish**.

   **Tip:** You can also type values for the **Hrs** and **Mins** boxes.

**Results**

A list report appears showing all products that were shipped after the time interval you specified. For example, if you typed 5, the list will show products that were shipped more than 5 days after the order date.

![Figure 46. List showing order number, product name, quantity, date, and time interval](image)

---

**Create a Parameter to Produce a Prompt**

IBM Cognos Report Studio can automatically generate prompted reports based on parameters you create. When you run the report, Report Studio can generate a prompt page for each parameter not associated to an existing prompt page depending on whether the prompt run option is selected or not.

**Note:** When Report Studio automatically generates a prompt, it creates filters in the query using the prompt information properties that are set in the IBM Cognos Framework Manager model. For more information about the prompt information properties, see the Framework Manager *User Guide*.

You can also define parameters when you want to create a drill-through report or define master detail relationships.

**Procedure**

1. From the **Data** menu, click **Filters**.

2. On the **Detail Filters** tab, click the add button.

   The **Detail Filter** dialog box appears.
3. In the Available Components box, click the Source tab or the Data Items tab to select the data item to use for the prompt:
   • To filter data based on data items not shown in the report, double-click a data item on the Source tab.
   • To filter data that appears in the report but not necessarily in the model, such as calculations, double-click a data item on the Data Items tab.

   The data item appears in the Expression Definition box.

4. In the Expression Definition box, type an operator after the data item or select an operator from the Functions tab.

   The operator sets some of the default properties of the prompt. For example, if the operator is equals (=), users will be able to select only a single prompt value and the prompt’s Multi-Select property is set to No.

   For more information about creating expressions, see “Using Relational Calculations” on page 248 or “Using Dimensional Calculations” on page 302.

5. Type a name after the operator to define the prompt parameter. A question mark must precede and follow the name.

6. To specify whether the prompt is mandatory, in the Usage box, click Required, Optional, or Disabled.

---

**Create a Prompt Directly in a Report Page**

You can add prompt controls directly in a report page instead of creating a prompt page.

Prompt controls that are added to report pages will not appear in the following:
• saved reports
• PDF reports
• reports that are sent to users by email
• scheduled reports

Prompt controls are interactive. They are used to satisfy parameter values before running a report. As a result, prompt controls added to a report page only appear when you run the report in HTML format. When you run a report in HTML format, users select which values they want to see, and the report is refreshed, producing a new report.

For the non-interactive reports listed above, prompt parameter values must be collected and satisfied before the report is run. You provide the parameter values using the Run options tab in IBM Cognos Connection. If you do not provide all the required values, the report will fail to run. You can access the Run options tab by clicking the set properties button for the report.

**Tip:** The Revenue by Product Brand (2011) sample report in the Sales and Marketing (cube) package and the Rolling and Moving Averages interactive sample report include value prompts. For more information about The Sample Outdoors Company samples, see “Sample Reports and Packages,” on page 507.
Procedure

1. From the Toolbox tab, drag a prompt control to the report. The Prompt Wizard dialog box appears.
2. Provide the information necessary to create the prompt.

   Tip: Make the prompt optional by selecting the Make the filter optional check box. Otherwise, when you run the report for the first time, the prompt appears in a generated prompt page rather than in the report page. Alternatively, you can specify a default selection for the prompt.
3. In the work area, click the prompt.
4. In the Properties pane, set the Auto-Submit property to Yes.
   If you do not set the Auto-Submit property to Yes, the report will not refresh when users select a different prompt value.

   Tip: An alternative to setting the prompt to auto-submit is to add a prompt button from the Toolbox tab and set its Type property to Finish.

Results

The prompt control is added to the report page. You can modify its properties by clicking it and making changes in the Properties pane.

Modifying Prompts

For each prompt you create, you can modify its properties by specifying values in the Properties pane.

For example, you can
- change the prompt control interface
- require user input
- enable multiple selections
- show or hide prompt statuses
- specify default selections
- specify prompt values
- add prompt buttons
- create cascading prompts

Some properties you set for a prompt may be overridden under some conditions. For example, some properties set for the filter associated with a prompt may override the corresponding prompt property.

Change the Prompt Control Interface

By default, when you create a prompt, IBM Cognos Report Studio selects the prompt control interface. You can change the prompt control interface depending on the type of prompt you created. For example, for a value prompt, you can choose a drop-down list, a list box, or a radio button group.

Procedure

1. Click the prompt control.
2. In the Properties pane, set the Select UI property to the interface.
Specify That a Prompt Requires User Input

You can specify that a prompt requires user input before the report can run.

Procedure
1. Click the prompt control.
2. In the Properties pane, set the Required property to Yes.
3. Pause the pointer over the page explorer button and click a report page.
4. From the Data menu, click Filters.
5. Click the filter associated with the prompt.
6. In the Usage box, click Required.

Results
When you run the report, a star appears next to the prompt indicating that the user must select or type a value.

If you have a cascading prompt and the parent prompt control is required, the child prompt control is disabled. This ensures that users choose at least one value in the parent prompt before they can choose a value in the child prompt. Conversely, if the parent control is optional, then the child control is populated. This gives users the ability to choose values in the child prompt without having to choose a value in the parent prompt.

Allow Users to Select Multiple Values in a Prompt

You can allow users to select more than one value in a prompt. For example, you have a prompt for which users must select a product line. You can modify the prompt so that users can select more than one product line.

If you enable multiple selections, the Auto-Submit property is always set to no.

Procedure
1. Click the prompt control.
2. In the Properties pane, choose whether to allow users to specify more than one value or a range of values:
   - To allow users to specify more than one value, set the Multi-Select property to Yes.
   - To allow users to specify a range of values, set the Range property to Yes.
3. Pause the pointer over the page explorer button and click a report page.
4. From the Data menu, click Filters.
   - If you have more than one query defined in the report, you must first click an object linked to a query.
5. Double-click the filter associated with the prompt.
6. Change the operator to one of the following:
   - If you are creating a multi-select prompt, change the operator to in.
     For example, [Product_line] in ?Product line? where [Product_Line] is the name of the data item allows users to select multiple product lines.
   - If you are creating a range prompt, change the operator to in_range.
For example, [Margin] in_range ?Margin? where [Margin] is the name of the data item allows users to specify a margin range.

**Show or Hide Prompt Status**
Each prompt you create in a report provides dynamic validation when the report is run. Validity checks are performed to ensure that the data is correct and that required values are supplied. For example, a star appears next to each required prompt. An arrow appears next to a prompt if you must select or type a value. If you type an incorrect value, a dotted line appears. You can choose whether to show the star and arrow for each prompt.

**Procedure**
1. Click the prompt control.
2. In the Properties pane, set the Hide Adornments property to **Yes** to hide the prompt characters or **No** to show them.

**Specify a Default Selection for a Prompt**
You can specify a default selection for a prompt so that users do not have to select or type a value when they run the report.

**Procedure**
1. Click the prompt control.
2. To define a range of values, in the Properties pane, set the **Range** property to **Yes**.
3. To specify more than one default selection, in the Properties pane, set the **Multi-Select** property to **Yes**.
4. In the Properties pane, double-click the Default Selections property.
5. Click the add button [Add] and do one of the following:
   - If you chose to define a single value, type the value as the default selection.
   - If you chose to define a range of values, type the minimum and maximum values of the range in the **Minimum value** and **Maximum value** boxes, respectively.
6. Repeat step 4 to specify other default selections.

**Customize Prompt Text**
You can customize the instructional text that appears around prompts. For example, a value prompt with multiple selections includes a **Select all** link below the choices that you can customize to text other than **Select all**. You can also specify translated text in many languages for international report users.

**Procedure**
1. Click the prompt control.
2. To change the default prompt text, in the Properties pane, double-click any of the properties under Prompt Text.

   **Note:** When you select a property in the Properties pane, its description appears in the information pane below the Properties pane.
3. Click **Specified text**, and then click the ellipsis (...) button.
4. In the **Localized Text** dialog box, type the text that you want to appear.
5. To add customized text for other languages:
• Click Add.
• Select the countries and regions for which you want to specify localized text and click OK.
• To specify text for a country and region, select the country and region and click Edit, and then type the localized text in the text box.

Specify Prompt Values
You can provide your own values in a prompt instead of the default values that come from the database.

You can specify your own prompt values to
• show something different from what is in the database
• improve performance by not accessing the database
• provide text for optional prompts, such as Select a value
• restrict the number of values available

For example, you have a prompt in which users choose a country or region. For the database value United States, you want USA to appear in the prompt.

Tip: The Rolling and Moving Averages interactive sample report includes a value prompt. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Procedure
1. Click the prompt control.
2. In the Properties pane, double-click the Static Choices property.
3. Click the add button.
4. In the Use box, type the prompt value to add.
5. In the Display box, type the value that will appear in the prompt.
6. Repeat steps 3 to 5 to add other prompt values.
7. To link a prompt value to a condition, do the following:
   • In the Variable box, choose the variable to use or create your own.
     For information about creating variables, see “Add a Variable from the Condition Explorer” on page 370.
   • In the Value box, click one of the possible values for the variable.
   • Click the static value to link to the variable and click the edit button.
   • In the Display box, type the value to appear in the prompt.
   • Repeat this procedure for each additional value.

Add a Prompt Button
Add prompt buttons so that users can submit selected items, cancel reports, or navigate between pages.

When you are building prompts and prompt pages, you may have to add prompt buttons to submit selections. Some prompt controls, such as the value prompt, can be set to submit selections automatically. Other prompt controls, such as the date prompt, require a prompt button.
**Procedure**

1. Pause the pointer over the page explorer button and click the page to which to add a prompt button.

2. From the Toolbox tab, drag Prompt Button to the work area.

3. Click the prompt button and, in the Properties pane, set the Type property to one of the following actions.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancel the report</td>
<td>Cancel</td>
</tr>
<tr>
<td>Go to the previous prompt page</td>
<td>Back</td>
</tr>
<tr>
<td>Go to the next prompt page</td>
<td>Next</td>
</tr>
<tr>
<td>Run the report</td>
<td>Finish</td>
</tr>
<tr>
<td>Reprompt the user</td>
<td>Reprompt</td>
</tr>
</tbody>
</table>

**Create a Cascading Prompt**

Create a cascading prompt to use values from one prompt to filter values in another prompt. For example, a report contains the columns Product line and Product type. You create prompts for these columns, and you specify that the Product type prompt is a cascading prompt that uses Product line as the source. When users select a product line, they see only the product types related to the selected product line.

**Tip:** The Employee Training by Year sample report in the GO Data Warehouse (analysis) package includes cascading prompts. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Procedure**

1. To make the cascading source a required prompt, select it and, in the Properties pane, set the Required property to Yes.

2. Click the prompt control to use as a cascading prompt.

3. In the Properties pane, double-click the Cascade Source property.

4. Click the parameter that represents the cascade source.

5. If the prompt allows users to select multiple values, add a prompt button to the cascade source to provide the cascading prompt with the appropriate values:

   - From the Toolbox tab, drag Prompt Button to the report.
   - Click the prompt button and, in the Properties pane, set the Type property to Reprompt.
   - To change the text in the prompt button, on the Toolbox tab, drag Text Item to the prompt button and type the text.
Control the Data That Appears in a Tree Prompt

You can control what data appears in a tree prompt and how the data is structured to get the results that you want. To do this, you add various functions to the filter expression.

In addition, the operator that you chose in the Prompt Wizard dialog box controls what appears next to each prompt value. If the operator is in or not in, check boxes appear next to each prompt value. If the operator is equals (=), no check boxes appear.

Procedure

1. Pause the pointer over the query explorer button and click the query that is associated with the prompt.
2. In the Data Items pane, double-click the data item on which you are prompting.
3. In the Expression Definition box, type one of the following functions.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show the hierarchical structure of all members in the hierarchy. If this function is not used, all members are shown in a flat list.</td>
<td>rootmembers (data_item)</td>
</tr>
<tr>
<td>Show the descendants of the data item in a hierarchical structure where (x) represents the level. For example, if you are prompting on the Year hierarchy and (x=1), you will see 2010, 2011, and 2012 in the tree. If (x=2), you will see 2010 Q1, 2010 Q2, and so on.</td>
<td>descendants (rootmembers (data_item), (x))</td>
</tr>
<tr>
<td>Show the children of a member. For example, 2012 Q1, 2012 Q2, 2012 Q3, and 2012 Q4 appear for the member 2012.</td>
<td>children (member)</td>
</tr>
</tbody>
</table>

Using in_range Filters with Character Data

If you use an in_range filter with character data, and the From value is greater than the To value, the filter returns no results. For example, if the From value is "Zone" and the To value is "Aloe Relief", the report returns no data.

To allow results within a range regardless of whether the From value is greater than the To value, your IBM Cognos administrator can enable a prompting setting. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

Showing prompt values in generated SQL or MDX

You can show prompt values when viewing the generated SQL or MDX of a query or report.

Procedure

1. Pause the pointer over the query explorer button and click the query for which you want to view the generated SQL or MDX.
2. In the Properties pane, set the Use SQL Parameters property to Literal.

Related tasks:
- “View the SQL for an Entire Report or a Query” on page 246
  View the SQL to see what is passed to the database when you run a report.
- “View the SQL or MDX for an Entire Report or a Query” on page 295
  View the SQL or MDX to see what is passed to the database when you run a report.
Chapter 12. Laying Out a Report

When creating a report, a good layout is essential to ensure that the information in the report is presented in a clear and effective manner.

Report Layout Recommendations

Consider the following recommendations when creating the layout of a report.

- Define the page structure.
  Determine what goes into the page header, body, and footer. The page header contains information that appears at the top of each page. The page body contains information that starts on the first instance of the page. If there is too much data to fit on a single page, it continues across all instances of the page. The page footer contains information that appears at the bottom of each page.

- Identify horizontal bands of information.
  Look for natural bands of information running across the page. Each of these bands typically translates into a block.

- Identify vertical bands of information.
  In each horizontal band of information, look for bands that run up and down the page. Each of these bands typically translates into table cells.

- Decide which data frames to use to lay out the data.
  Choose a list, crosstab, chart, repeater, or text frame.

- Set properties at the highest level item possible.
  By setting properties at the highest level, you set them once instead of setting them for each child object. For example, if you want all text items in a report to appear in a specific font, set the font for the page.

Tip: When setting properties for an object, click the select ancestor button in the Properties pane title bar to see the different levels available.

- Use padding and margins to create white space.
  Do not use fixed object sizing unless it is absolutely necessary. When you specify that an object has a fixed size, your layout becomes less flexible.

The Page Structure View

When you add objects to a report, you usually work in the layout. An alternative view of the report is available.

From the View menu, click Page Structure to view the report in a different way.
Use the page structure view
- to view the entire contents of a report page in a tree structure
  Using a tree structure is useful for locating the objects in a page and troubleshooting problems with nested objects.
- to quickly move objects from one area of a page to another
  If you have a complex layout, it may be difficult to select, cut, and paste objects in the layout view. Objects are easier to locate in the page structure view.
- to modify object properties
  You can modify object properties in the layout or in the page structure view.

Tip: To switch back to the report layout, from the View menu, click Page Design.

Related concepts:
“Work in design or structure view” on page 28

IBM Cognos Report Studio has two views in which you can author reports: Page Design view and Page Structure view. You can choose a report authoring view on the View menu.

Copy Object Formatting

You can quickly copy the formatting of items in your report, such as fonts, colors, borders, and number formats, and apply that formatting to other items.

Procedure
1. Click an item that has the formatting to copy.
2. Do one of the following:
   - To copy all the formatting applied to the item, click the pick up style button  , click the item to format, and then click the apply style button  .
   - To copy only one of the formatting styles, click the down arrow to the right of the pick up style button and click the style to copy. Then click the item to format and click the apply style button.
3. If you want to make changes to a style that you copied, click the down arrow to the right of the pick up style button and click Edit Dropper Style.
4. In the Style dialog box, specify basic and advanced style characteristics.
Add a Header or Footer to a Report or List

Add a header or footer to make a report easier to read. Headers and footers are containers in which you can add objects like text, images, and report expressions such as the current date and page numbers. You can add headers and footers to pages and lists.

Tip: The Revenue by Product Brand (2011) sample report in the Sales and Marketing (cube) package includes a customized header and footer. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Add a page header or footer when you want information to appear on every page in the report, such as a title or page numbers. You can use combinations of text, data items, and report expressions in titles.

You can add the following headers and footers to lists to organize data into logical sections or to identify every change in value of a column.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List page header</td>
<td>Adds a header that appears at the top of the list on every page in which list data appears.</td>
</tr>
<tr>
<td>Overall header</td>
<td>Adds a header that appears once at the top of the list.</td>
</tr>
<tr>
<td>Group or section header</td>
<td>Adds a header that appears for each group of a grouped column or each section.</td>
</tr>
<tr>
<td>Group or section footer</td>
<td>Adds a footer that appears for each group of a grouped column or each section.</td>
</tr>
<tr>
<td>Overall footer</td>
<td>Adds a footer that appears once at the bottom of the list.</td>
</tr>
<tr>
<td>List page footer</td>
<td>Adds a footer that appears at the bottom of the list on every page in which list data appears. Note that summary calculations in list page footers only summarize the data that is visible on that page of the list report.</td>
</tr>
</tbody>
</table>

You can also add section footers by adding a summary.

Before you begin

To see the headers and footers, ensure that the visual aids are turned on.

Procedure

1. If you want to add a page header or footer, from the Structure menu, click Headers & Footers, Page Header & Footer, select the appropriate check boxes, and click OK.

   Tip: You can also click the headers & footers button on the toolbar.

2. If you want to add a list header or footer, from the Structure menu, click Headers & Footers, List Headers & Footers, select the appropriate check boxes, and click OK.
Tip: You can also click the headers & footers button on the toolbar and then click the list headers & footers button.

3. If you want to add objects to a header or footer, drag the objects from the Toolbox tab to the appropriate location.

To add objects to a list header or footer, you must first unlock the report. From the Structure menu, click Lock Page Objects.

Tip: To add data items to the page header or footer, you must associate a query to the page.

4. To split a header or footer into multiple cells that span the list, from the Structure menu, click Headers & Footers and click Split List Row Cell.

5. To merge multiple cells in a header or footer, from the Structure menu, click Headers & Footers and click Merge List Row Cells.

6. To add rows to a header or footer, from the Structure menu, click Headers & Footers and click Insert List Row Cells Above or Insert List Row Cells Below.

### Add Borders to an Object

You can add borders to objects in a report such as a column, a header, or a footer or to the whole report.

Tip: You can also add borders using the IBM Cognos Report Studio toolbar.

**Procedure**

1. Click the object to which to add a border.

   Tip: To quickly select the parent of an object, click the object, and then click the select ancestor button in the title bar of the Properties pane.

2. In the Properties pane, double-click the Border property and select values in the Color, Style, and Width boxes.

   Note: Specifying a width using % as the unit of measure is not supported when producing reports in PDF.

3. Apply the selected values to the top, bottom, left side, right side, or all sides of the object by clicking the appropriate button in the Preview section.

   Tip: To clear the border applied, click the clear borders button.

### Add a Text Item to a Report

You can add text to a report. You can insert text in other objects, such as a block or table cell, or directly in the report page.

You can also add multilingual text to a report.

You can now format the text by changing the font, color, size, and so on. Select the text and make the appropriate changes in the Properties pane.
Procedure

1. From the Toolbox tab, drag the Text Item object to the report. The Text dialog box appears.
2. Type the text and click OK.

Tip: You can also paste text from another part of the report.

Specify the Font for a Report

You can specify the font for text in a report.

Procedure

1. Click the object.

Tip: To specify the default font for the report, click the page.

2. Do one of the following:
   • From the toolbar, specify the font properties.
   • In the Properties pane, double-click the Font property and specify the font properties.
   To specify the color of the text, click Foreground Color. Alternatively, you can specify the color of the text by clicking the Foreground Color property for the object.

Tip: Type a list of fonts in the Family box if you are not sure whether a specific font is installed on a user’s computer. For example, if you type Times New Roman, Arial, monospace, IBM Cognos Report Studio checks to see if Times New Roman is installed. If it is not, Report Studio checks for Arial. If Arial is not installed, the monospace font used by the computer is used.

Results

If you clicked (Default) for any of the font properties, the default value for the property is used. Default values are stored in a style sheet that is used across all IBM Cognos Business Intelligence tools. You can modify default values by modifying classes.

Insert an Image in a Report

You can insert an image in a report. You can insert images in other objects, such as blocks or table cells, directly in the report page or as the background image of another object.

Before you begin

The images that you insert must first be uploaded to the IBM Cognos Business Intelligence server or another Web server and must be .gif or .jpg format.

Procedure

1. From the Toolbox tab, drag the Image object to the report and then double-click it.
2. In the Image URL dialog box, type the URL of the image to insert or click Browse to go to the location containing the image.
To browse images on a Web server, you must enable Web-based Distributed Authoring and Versioning (WebDAV) on your Web server. For more information about configuring Web servers, see the IBM Cognos Business Intelligence Installation and Configuration Guide.

Insert a Background Image in an Object

You can insert a background image for objects in a report. For example, use a background image to add a watermark to a page.

You can also create your own background visual effects, such as drop shadows and gradient fills. For more information, see "Add Background Effects."

Before you begin

The images that you insert must first be uploaded to the IBM Cognos Business Intelligence server or another Web server and must be .gif or .jpg format.

Procedure

1. Select the object.
2. In the Properties pane, under Color & Background, double-click the Background Image property.
3. To use the background from the default style for the object (Cascading Style Sheet (CSS) class), click Default.
4. To explicitly not insert a background image, click None.
   This option overrides any default background images including background gradients on objects that appear in the 10.x report styles.
5. To insert a specific image, click Specified and in the Image URL box, type the URL of the image to insert or click Browse to go to the location containing the image.
   To browse images on a Web server, you must enable Web-based Distributed Authoring and Versioning (WebDAV) on your Web server. For more information about configuring Web servers, see the IBM Cognos Business Intelligence Installation and Configuration Guide.
6. In the Position box, choose how to align the image in the object.
7. In the Tiling box, click a tiling option.

Related tasks:
   "Change a Chart Background in a Legacy Chart” on page 115
You can use a solid color, a pattern, or a gradient fill effect to customize the chart background.

Add Background Effects

You can add background effects to enhance the appearance of your report. You can enhance objects, such as data containers (lists, crosstabs, and charts), headers, footers, page bodies, and so on with borders, gradient fill effects, drop shadows, and background images. You can also apply background effects as a class style.

The background effect is rendered within the rectangle area that bounds the object. To use a generated background in a chart, the chart must have a fixed size.

If the background image is complex and large, the size of the report output may be affected proportionately.
Background effects are rendered only if the data container has a fixed height and width; if a percentage size is used, the effects are ignored.

Resizing or overflow behavior is ignored for generated images in HTML reports.

**Add background effects to a chart**

Background effects include borders, fills, drop shadows, and images.

**Procedure**

1. Click the chart object.
2. To choose a preset background effect, click the background effects presets button on the style toolbar, and then click a background.
3. In the **Properties** pane, under **Color & Background**, double-click the **Background Effects** property.
4. Select one or more of the following:
   - To apply a border, click **Border** and specify settings for border style, width, color, corner radius for rounded rectangles, and transparency.
     - If the element also includes a fill with a transparency setting, select the **Allow transparent bleed** check box to apply the same transparency to the border.
   - To apply a fill effect, click **Fill** and specify the settings. The fill effect can either be a solid color, a gradient, or a pattern. You can define a gradient fill effect as a linear, radial line, or radial rectangle gradient.
   - To apply a drop shadow effect, click **Drop Shadow** and specify the shadow color, transparency value, color, and offset settings. The default horizontal and vertical offset is 5 pixels.
   - To specify one or more images as a background, click **Images**. You can specify the transparency value and the position for each defined image. You can also specify a custom position for each image.

**Tip:** To remove the effect, clear its check box.

**Add a background gradient to a page**

A background gradient is a shading that goes from one color to another either horizontally or vertically across your page.

**Procedure**

1. Click anywhere in the report page.
2. In the **Properties** pane title bar, click the select ancestor button and click **Page**.
3. Under **Color & Background**, double-click the **Gradient** property.
4. Select the **Gradient** check box and then select the colors and direction for the gradient.

Background gradients and drop shadows do not appear in Microsoft Excel spreadsheet software output. They are also supported only for report outputs run in the Microsoft Internet Explorer Web browser.

**Tip:** To remove the effect, clear the **Gradient** check box.
Add a drop shadow to a container

A drop shadow is a border on the bottom and right of an object. You can specify the color and transparency of this shadow.

Procedure

1. Click the crosstab, list, repeater table, or table in the report.
2. In the Properties pane title bar, click the select ancestor button and click Crosstab, List, Repeater Table, or Table.
   Tip: You can also click the container selector (three orange dots) of the container to select it.
3. Under Color & Background, double-click the Drop Shadow property.
4. Select the Drop shadow check box and then select the color, offset, and transparency values.

   Background gradients and drop shadows do not appear in Excel output. They are also supported only for report outputs run in the Internet Explorer Web browsers.
   If you add a drop shadow, ensure that you also specify a background color for the object. Otherwise, if you keep the default transparent background, data values also have a drop shadow and are difficult to read.
   Tip: To remove the effect, clear the Drop shadow check box.

Insert a Formatting Object in a Report

In addition to text and images, the Toolbox tab contains other objects that you can add to the report layout.

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
</table>
| Block           | Adds an empty block, which is a container in which you can insert other objects. This is useful for controlling where objects appear.  
   Tip: You can use blocks to add space between objects. However, empty blocks are not rendered. You must insert an object or specify the height and width.  
   Tip: The Revenue by GO Subsidiary 2011 sample report in the GO Data Warehouse (analysis) package includes an empty block. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507. |
<p>| Table           | Adds a table which is a container in which you can insert other objects. This is useful for controlling where objects appear. |
| Field Set       | Adds an empty block that has a caption. This is similar to the Block object, but with a caption. |
| Intersection (Tuple) | Adds an intersection (tuple). For more information, see “Create an Intersection (Tuple)” on page 306 |
| Query Calculation | Adds a calculated column. For more information, see “Using Relational Calculations” on page 248 or “Using Dimensional Calculations” on page 302 |</p>
<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Crosstab Space**             | Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge. Blank cells appear for the edge when the report is run.  
Insert this object when a crosstab edge does not produce useful data and you want blanks to appear in the cells instead. |
| **Crosstab Space (with fact cells)** | Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge. The contents of the fact cells for the edge are rendered when a measure is added or the default measure is specified.  
- If the crosstab space is nested, the scope of the fact cells is the scope of the item that is at the level before the space.  
- If the crosstab space is not nested and there are no items nested below it, the scope of the fact cells is the default measure. |
| **Singleton**                  | Inserts a single data item.                                                                                                                                                                      |
| **Conditional Blocks**         | Adds an empty block that you can use for conditional formatting.                                                                                                                                   |
| **HTML Item**                  | Adds a container in which you can insert HTML code. HTML items can be anything that your browser will execute, including links, images, multimedia, tooltips, or JavaScript.  
HTML items appear only when you run the report in HTML format.  
**Note:** You cannot include `<form>` tags in HTML items.  
**Tip:** The Table of Contents sample report in the GO Sales (analysis) package includes an HTML item. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.  
**Note:** When you upgrade to the next version of IBM Cognos Business Intelligence, the report upgrade processes do not account for the use of undocumented and unsupported mechanisms or features, such as JavaScript, that refers to IBM Cognos HTML objects. |
| **Rich Text Item**             | Inserts an object that is used to render HTML in the layout. This object is similar to the **HTML Item**, except that rich text items also render in PDF output. Using rich text items is useful when you want to add annotations defined in a data source to a report.  
**Note:** Rich text items only support only a restricted set of well-formed XHTML. |
| **Hyperlink**                  | Adds a hyperlink so that users can jump to another place, such as a Web site.                                                                                                                      |
| **Hyperlink Button**           | Adds a hyperlink in the form of a button.                                                                                                                                                           |
| **As of Time Expression**      | Adds an As of Time expression so that you can show data for a specific time period.                                                                                                                  |
| **Page Number**                | Inserts page numbers that you can customize.                                                                                                                                                       |
### Object Description

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Number</td>
<td>Numbers each row of data returned when the report is run.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can add row numbers only to lists and repeaters.</td>
</tr>
<tr>
<td>Layout Component Reference</td>
<td>Adds a reference to another object. Useful when you want to <strong>reuse an object</strong>.</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>Creates a table of contents that is rendered in the output of a report.</td>
</tr>
<tr>
<td>Table of Contents Entry</td>
<td>Adds a table of contents marker.</td>
</tr>
<tr>
<td>Bookmark</td>
<td>Inserts a bookmark so that users can move from one part of a report to another.</td>
</tr>
</tbody>
</table>

For information about each prompt control and prompt buttons, see Chapter 11, “Adding Prompts to Filter Data,” on page 313. The Toolbox tab contains other objects when you work in other areas such as Query Explorer and Condition Explorer.

### Before you begin

Before you can add a hyperlink, HTML item, or hyperlink button, you must have the **HTML Items in Report** capability. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

### Procedure

To add an object, drag or double-click it from the Toolbox tab.

### Elements Supported in Rich Text Items

Rich text items support all XHTML character entities, such as &nbsp; (non-breaking space), and the following elements: div, span, ul, ol, and li.

Each element only supports the style attribute, which must contain a valid CSS style. In addition, ul and ol elements support list-style attributes. Specifically, the ol element supports decimal, and the ul element supports circle, disc, and square, as well as list-style-image.

For example, the following code produces an unordered list entitled List: with three items. Each list item is in a different color, and the list-style attribute used is circle.

```html
<div style="font-size:14pt; text-decoration:underline">List:</div>
<ul style="list-style-type:circle">
<li style="color:green">Item A</li>
<li style="color:red">Item B</li>
<li style="color:blue">Item C</li>
</ul>
```

### Example - Add a Multimedia File to a Report

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You want to insert a Microsoft Windows Media Audio/Video file named GO.wmv in a template that serves as a cover page for all reports.
You must have Windows Media Player installed on your computer.

**Procedure**

1. Open IBM Cognos Report Studio with the **GO Data Warehouse (query)** package.
2. In the **Welcome** dialog box, click **Create a new report or template** and in the **New** dialog box, click **Blank**.
3. From the **Toolbox tab**, drag the **HTML Item** object to the report.
4. Select the **HTML Item**.
5. In the **Properties pane**, double-click the **HTML** property.
6. In the **HTML dialog box**, type the following:
   ```html
   <OBJECT classid="CLSID:6BF52A52-394A-11D3-B153-00C04F79FAA6">
   <PARAM NAME="URL" VALUE="/c10/webcontent/samples/images/GO.wmv"/>
   </OBJECT>
   ```

**Results**

When you run the report in HTML format, the multimedia file plays in Windows Media Player.

---

### Align an Object in a Report

You can specify a horizontal and vertical alignment for an object in a report to determine where they appear.

**Tables** can also be used to determine where objects appear in a report.

**Note:** The **Justify** horizontal alignment works with HTML output but does not apply to PDF output.

**Procedure**

1. Select the object to align.
2. From the toolbar, click one of the available horizontal or vertical alignment buttons.

---

### Using a table to control where objects appear

You can use tables in your report to control where objects appear. Tables can be inserted anywhere in a report, such as a header, a footer, or the page body. After you create a table, insert the objects you want in the cells.

You can also apply a predefined **table style** to tables. Other table properties are described in **Appendix G, “Report Studio Object and Property Reference,”** on page 771.

The **alignment buttons** can also be used to determine where objects appear in a report.

**Tip:** The Singletons on Page Body sample report in the GO Sales (analysis) package includes a table to control the report layout. For more information about The Sample Outdoors Company samples, see **Appendix C, “Sample Reports and Packages,”** on page 507.
Procedure

1. From the Toolbox tab, drag Table to the report.
   The Insert Table dialog box appears.
2. In the Number of columns and Number of rows boxes, type the number of columns and rows for the table.
3. If you want to have the table span the width of the report page, select the Maximize width check box.
4. If you want to add borders to the table, select the Show borders check box.
5. If you want to merge cells in the table, select the cells and click the merge cells button.
6. Select the table object.
7. In the Properties pane, under Positioning, double-click the Table Properties property.
8. To display both the inside and outside borders, do the following:
   - Clear the Collapse borders check box.
   - Type a number in the Cell spacing text box to define how much space to add between the table cells.
   - Select the Show empty cell borders check box to display the borders even for empty cells.
9. To fix the size of the columns in the table, check the Fixed size check box.
   When this check box is cleared, the table columns expand to fit the text.

Apply a Table Style

Apply a table style to quickly format tables. You can also apply a table style to lists, crosstabs, and repeater tables.

About this task

By default, when a new object is inserted in a list or crosstab, the object inherits the style from an object of the same type in the data container. For example, if you insert a measure in a list, the measure inherits the style of a measure that is already in the list, if there is one. If you do not want objects to inherit styles, clear the Table Style inheritance option in the Tools menu (Tools, Options, Report tab).

The following rules explain how style inheritance is applied to lists and crosstabs.

- Styles are inherited in the following order: custom, client default, and server default.
- A custom style is a style that you manually apply. A client default style is one of the styles available in the Apply Table Style dialog box. The server default style is the style applied when Default is selected in the Apply Table Style window.
- When a new column or row is inserted, it inherits the style from the sibling of the same type that is on its right or below.
- If there is no sibling of the same type, then the nearest sibling that is on its left or above is applied.
- If there is no sibling of the same type in the container, then the client or server default table style is applied.
- If a custom style is applied to a column or row and the object is then moved to another location, the object retains the custom style.
Custom styles applied to part of an object, such as the header, body, or footer, may be lost. For example, if a container has a footer, the footer is recreated when a column is moved. The recreated footer is rendered using the style that was applied to the footer before you customized its style.

- Deleting a column or row has no impact on the styles of the other objects in the container.
- Styles are preserved if a column is grouped or ungrouped. The same applies when creating or removing sections.
- When drilling up or down, the style of the parent item is applied.

**Procedure**

1. Click the table, list, crosstab, or repeater table.
2. From the Table menu, click Apply Table Style.
3. In the Table styles box, click a table style.

   **Tip:** Some styles are unique to tables, lists, or crosstabs.
4. If you are applying a table style to a table or list, in the Apply special styles to section, select or clear the various column and row check boxes based on how you want to treat the first and last columns and rows.
   Some check boxes may not be available for particular table styles, or to particular columns or rows.
5. If you are applying a table style to a list or crosstab and you want the style to be applied to all lists or crosstabs in the report, select the Set this style as the default for this object type check box.
   For a list, you may need to clear the First column and Last column check boxes in the Apply special styles to section before you can select this check box. In addition, some table styles cannot be set as the default.

---

**Apply Padding to an Object**

Apply padding to an object to add white space between the object and its margin or, if there is a border, between the object and its border.

**Tip:** You can quickly apply left or right padding by either pressing Tab and Shift+Tab or by clicking the increase indent and the decrease indent buttons in the toolbar. When using the toolbar buttons, you can indent an object by up to nine times the indentation length. You can specify the indentation length to use by clicking the arrow beside either button and clicking Set Default Indent Length.

**Tip:** The GO Balance Sheet as at Dec 31 2012 sample report in the GO Data Warehouse (analysis) package includes padding. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Procedure**

1. Select the object to which to apply padding.
2. In the Properties pane, double-click the Padding property.
3. Specify top, bottom, left, and right padding by typing values in the corresponding boxes and choosing the unit of measure.

   **Note:** Specifying padding using % as the unit of measure is not supported when producing reports in PDF.
Set Margins for an Object

Set the margins for objects in a report to add white space around them.

For Date, Time, Row Number, and Page Number objects, you can only set the left and right margins. If you want to set the top or bottom margins for these objects, place them in a table or a block. Then set the margin or padding properties on the table or block object.

Procedure
1. Select the object.
2. In the Properties pane, double-click the Margin property.
3. Specify the top, bottom, left, and right margins by typing values in the corresponding boxes and choosing the unit of measure.

Note: Specifying margins using % as the unit of measure is not supported when producing reports in PDF.

Add Multiple Items to a Single Column

You can add multiple items to a single column to condense a report. For example, you have a list report that contains many columns. You can reduce the number of columns in the list by putting related information in a single column.

Procedure
1. From the Structure menu, click Lock Page Objects to unlock the report.
2. From the content pane, drag the items to the column.

   For example, you can add data items from the Source tab or text items from the Toolbox tab.

Example - Create a Report with Multiple Items in One Column

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a list report showing the name, position, email address, phone number, extension, and fax number for each sales representative in each city organized by country or region. To reduce the number of columns in the list, you show email addresses, phone numbers, extensions, and fax numbers in a single column.

Procedure
1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click List and click OK.
4. From the Source tab, expand HR (query), Employee summary (query), and Employee by region. Add the following data items to the list by double-clicking them:
   - Country or Region
   - City
   - Employee name
• Position name
• Email
• Work phone
• Extension
• Fax

5. Click the Country or Region column and, from the Structure menu, click Section.

6. From the Structure menu, ensure that the report is unlocked.

7. Click Country or Region and, in the Properties pane, double-click the Font property.

8. Change the font to Arial Black, 11 pt, and Bold, and then click OK.

9. From the Toolbox tab, drag Table to the right of the Work phone text item in the first row of Work phone column and create a table that has one column and three rows.

10. Drag the following data items to the table:
• Email to the first row
• Work phone to the second row
• Fax to the third row

11. Drag Extension to the right of Work phone in the table.

12. From the Toolbox tab, drag Text Item to the left of each item in the table and type the following for each item, putting a blank space before and after the text:
• Email:
• Work phone:
• ext:
• Fax:

13. Ctrl+click the Extension, Email, and Fax column titles and click the delete button.

14. Click the Work phone column title.

15. In the Properties pane, click the Source Type property and click Text.

16. Double-click the Text property.

17. Type the following and click OK:
Contact Information

18. Select the Work phone data item in the first row of the list above the table and click the delete button.

19. Double-click the text item in the page header, type the following, and click OK:
Sales Representatives Contact List

20. From the Structure menu, click Lock Page Objects.

The report is locked.

21. Run the report.

Results

Contact information for each sales representative appears in a single column.
Reuse a Layout Object

You can save time by reusing layout objects that you add to a report instead of re-creating them. For example, you have a multiple-page report and you want to show the company logo in the page header of each page. Insert the logo once and reuse it on all other pages.

Procedure
1. Click the object to reuse.

   Tip: To quickly select the parent of an object, click the object, and then click the select ancestor button [ ] in the title bar of the Properties pane.

2. In the Properties pane, set the Name property to a value beginning with a letter to uniquely identify the object and press the Enter key.
   IBM Cognos Report Studio may have already specified a name for the object.

3. To reuse the object in another report, open that report.

4. From the Toolbox tab [ ], drag the Layout Component Reference object to the location in which it will be reused.
5. In the **Component Location** box, do one of the following:
   - To reference an object in the current report, click **This report**.
   - To reference an object in another report, click **Another report**, click the ellipsis (...) button, and open the report.

   **Note:** In active reports, you cannot reference objects from another active report.

6. In the **Available components to reference** box, click the object and click **OK**.

7. If the referenced object is in another report, click the **Layout Component Reference** object and in the **Properties** pane, set the **Embed** property to specify how to store the referenced object in the report:
   - Click **Copy** to store a copy of the object.
     The copy is not automatically updated if the source object is modified.

   **Note:** In active reports, you can store referenced objects only by copying them to the report.
   - Click **Reference** to store a reference, or pointer, of the object.
     The reference of the object is automatically updated if the source object is modified. For example, if you open or run the report, you see the modified object.

**Results**

A copy or reference of the object appears where you placed the **Layout Component Reference** object. If a source object is changed, you can update reused objects.

**Tip:** You can also create a new report or template and add all the objects to share. All your shared objects then reside in a single location, like a library.

**Change a Reused Object**

If you reuse an object that contains other objects, you can change the child objects to something different. For example, you have a block object containing a text item in the page header and you decide to reuse the block in the page footer. However, you want the text item in the page footer block to show different text from that in the page header.

**About this task**

If you reuse a visualization, you cannot change the data container of the visualization.

**Procedure**

1. In the parent object to reuse, click the child object to change.
2. In the **Properties** pane, set the **Name** property to a value beginning with a letter to uniquely identify the object.
   IBM Cognos Report Studio may have already specified a name for the object.
3. Select the copy of the parent object you created with the **Layout Component Reference** object.
4. In the **Properties** pane, double-click the **Overrides** property.
5. In the **Overrides** dialog box, select the child object to change and click **OK**.
The child object in the copy of the parent object is replaced by the following text:

Drop item to override component child.

6. Drag an object to replace the child object.

You can replace the child object with any other object, not just an object of the same type. For example, if the child object is a text item, you can replace it with an image.

**Update Reused Objects**

If a report contains objects referenced in another report, you can quickly update the referenced objects if the source objects have changed. Shared objects are stored in the layout component cache.

**Procedure**

1. Open the report to update.
2. From the **Tools** menu, click **Layout Component Cache**.
   
   The **Layout Component Cache** dialog box appears, showing all reports that contain referenced objects and the source reports where the objects exist.
3. To view which components are reused, click a source report.
   
   The components that are reused appear in the **Components used** pane.
4. Click **Reload Components** to refresh all referenced objects.
   
   Although referenced objects are automatically refreshed when you open or run a report, clicking this button updates components that were changed while the report is open.
5. Click **Update All Component Copies** to refresh all copied objects.
6. Click **Close**.

**Specify text and container direction**

You can specify text and container direction by choosing any of these options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Direction</strong></td>
<td>Sets the reading order of an object, such as right-to-left. If <strong>Inherit</strong> is selected, the direction is inherited from the parent object. The default is left-to-right. For text objects, an additional choice named <strong>Contextual</strong> exists. This choice sets the text direction based on the first letter in the text. If the letter belongs to a right-to-left script, the text direction is right-to-left. Otherwise, the text direction is left-to-right. Numbers and special characters do not influence the text direction. For example, if the text starts with a number followed by an Arabic letter, the direction is right-to-left. If the text starts with a number followed by a Latin letter, the direction is left-to-right. <strong>Tip:</strong> You can also set the direction of text objects by clicking the Text direction icon in the toolbar. This icon is visible only when the IBM Cognos Connection user preference <strong>Enable bidirectional support</strong> is selected.</td>
</tr>
<tr>
<td><strong>Writing mode</strong></td>
<td>Sets the direction and flow of content in an object.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Bidirectional</td>
<td>Sets the level of embedding in an object. For text objects, if the <strong>Direction</strong> option is set to a value other than <strong>(Default)</strong> and no value is selected for this option, this option is set to <strong>Embed</strong>. Setting this option to <strong>Embed</strong> ensures that the base text direction specified for the text is applied.</td>
</tr>
<tr>
<td>Justification</td>
<td>Sets the type of alignment used to justify text in an object.</td>
</tr>
<tr>
<td>Kashida space</td>
<td>Sets the ratio of kashida expansion to white space expansion when justifying lines of text in the object. This property is used in languages that use the Arabic writing script.</td>
</tr>
</tbody>
</table>

**About this task**

Specifying text direction and container direction depends on the object selected in the report. The following list describes the types of objects in Report Studio that you can specify text or container direction.

- **Compound objects**
  Compound objects are container objects that contain text. Examples include charts, maps, active reports, and prompts. You cannot set text direction for specific text objects in a compound object.

  To specify container direction for compound objects, click the **Direction & Justification** property for the object. To specify text direction for text in compound objects, click the **Contained Text Direction** property for the object.

- **Container objects**
  You can specify only container direction for container objects, such as a report page. Click the **Direction & Justification** property for the object to specify container direction.

  By default, the text direction of text in a container object is inherited from the container.

- **Text objects**
  For all text objects, the **Direction & Justification** property specifies the text direction.

In charts, container direction impacts all aspects of a chart. The following list describes the parts of a chart that are affected by container direction.

- The chart itself.
- The location of the Y-axis and the horizontal run direction of the X-axis.
- The labeling, including the orientation of angled labels on axes.
- The location of the legend as well as the legend content.

Container direction has no impact on rotary axes. For example, the slices in a pie always progress in the same direction around the pie. However, container direction does affect labelling as well as position and direction of the legend.

**Note:** You cannot specify base text direction and container direction for legacy charts.
**Procedure**

1. Click an object.

   **Tip:** To specify the base text direction of a text object that is inside another object, such as a list column, you must first unlock the report.

2. In the Properties pane, click one of the following properties:
   - Direction & Justification
   - Contained Text Direction
     This property applies only to compound objects.

3. Specify the text direction and container direction options that you want.

**Related concepts:**

"Support for bidirectional languages" on page 56

You can author reports that support bidirectional languages. You can specify base text direction, digit shaping, and container direction.

---

### Specify Line Spacing and Breaking

You can specify text properties by choosing any of these options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Height</td>
<td>Sets the distance between lines of text in an object.</td>
</tr>
<tr>
<td>Letter Spacing</td>
<td>Sets the amount of additional space between letters in an object.</td>
</tr>
<tr>
<td>Text Indent</td>
<td>Sets the indentation of the first line of text in an object.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This property works with HTML output but does not apply to PDF output.</td>
</tr>
<tr>
<td>Word Break</td>
<td>Sets line-breaking behavior within words.</td>
</tr>
<tr>
<td>Break words when necessary</td>
<td>Sets whether to break words when the content exceeds the boundaries of an object.</td>
</tr>
</tbody>
</table>

---

### Specify the Height and Width of an Object

You can specify the height and width of objects using various units of measurement. In addition, if the object is a field set, text box prompt, prompt button, hyperlink button, block, or a conditional block, you can specify how to handle content overflow. Specify the height and width by choosing any of these options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>Sets the height of the object.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Width</td>
<td>Sets the width of the object.</td>
</tr>
<tr>
<td>Content is not clipped</td>
<td>If the contents of the block exceed the height or width of the block, the block automatically resizes to fit the contents.</td>
</tr>
<tr>
<td>Content is clipped</td>
<td>If the contents of the block exceed the height or width of the block, the content is clipped. <strong>Note:</strong> The clipped content still exists. It is just not visible in the block.</td>
</tr>
<tr>
<td>Use scrollbars only when necessary</td>
<td>If the contents of the block exceed the height or width of the block, scrollbars are added to the block.</td>
</tr>
<tr>
<td>Always use scrollbars</td>
<td>Scrollbars are added to the block.</td>
</tr>
</tbody>
</table>

Do not use percentages to resize charts and maps that contain interactive elements that are activated when you pause the pointer over them, such as tooltips or drill-through links, because the browser is unable to realign the hard-coded hot spots after an image is resized.

When you use a percentage to specify the size of an object, the percentage is relative to the object's parent. In some cases, setting the size of an object using percentages will not give you the results that you want unless you also specify the size of the parent container.

**Procedure**
1. Click an object.
2. In the **Properties** pane, double-click the **Size & Overflow** property and specify the height and width.

### Control How Other Objects Flow Around an Object

You can control how objects flow around other objects by choosing any of the following options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Float</td>
<td>Sets how other objects flow around the object.</td>
</tr>
<tr>
<td>Allow floating objects on both sides</td>
<td>Allows other objects to flow on both sides. The <strong>Float</strong> property must be set.</td>
</tr>
<tr>
<td>Move below any floating object on the left side</td>
<td>If there are other objects to the left of the object, the object moves below those objects. The <strong>Float</strong> property must be set.</td>
</tr>
<tr>
<td>Move below any floating object on the right side</td>
<td>If there are other objects to the right of the object, the object moves under those objects. The <strong>Float</strong> property must be set.</td>
</tr>
<tr>
<td>Move below any floating object</td>
<td>Moves the object under any other object in which the <strong>Float</strong> property was set.</td>
</tr>
</tbody>
</table>

**Procedure**
1. Click an object.
2. In the **Properties** pane, double-click the **Floating** property.
3. Specify how other objects flow around the object by specifying the floating options.

Specify Report Properties

You can change the default report property settings.

Procedure
1. From the File menu, click Report Properties.
2. To specify the classes that IBM Cognos Report Studio uses to format objects, click Report styles and choose one of the available options.
   For more information, see “Create and Modify Report and Object Styles.”
3. For reports with multiple data containers, to render the default number of rows of each data container on each HTML page, you must set the Page break by data container for interactive HTML option to Yes. The default value is No.
   For more information, see “Controlling the Rows Per Page for Multiple Containers in HTML and PDF” on page 48.
4. To automatically create extended data items every time a data item is inserted, select the Always create extended data items check box.
   IBM Cognos Report Studio creates extended data items by default. For more information about extended data items, see “Extended Data Items” on page 263.
5. To create a single, scrollable HTML page when you run and save this report as HTML from IBM Cognos Connection, clear the Paginate saved HTML output check box.
   The default is to paginate HTML reports with the same page breaks as PDF reports. The single, scrollable HTML page will be available from IBM Cognos Connection when you click the view report output versions action.
6. When producing reports in Excel 2007 format, if you do not want to group repeating cells in lists and crosstabs, clear the Group repeating cells when exporting to Excel check box.
   When this check box is selected, grouped cells in crosstab rows and columns and grouped columns in lists are merged into a cell area in the Excel output. When the check box is cleared, merged cells are split. For more information, see “Option to group repeating cells in reports produced in Excel 2007 format” on page 50.
7. If you are working with reports created in IBM Cognos ReportNet and you want to create CSV report output, click Use 1.x CSV export.
   This option ensures that all the query columns are exported. In IBM Cognos ReportNet, if a data item was referenced using the Properties property of a list, it was included in the CSV output. In IBM Cognos Business Intelligence, the default is to export only the columns in the list.

Create and Modify Report and Object Styles

Create your own classes or modify existing classes in a report to format objects across a report according to your particular needs. In IBM Cognos Report Studio, objects in reports are assigned a Cascading Style Sheet (CSS) class that provides a default style for the object. For example, when you create a new report, the report title has the class property Report title text assigned to it. In addition, objects inherit the classes set on their parent objects.
You can use classes to highlight data using conditional styles.

Classes you create or modify can be applied only to the current report. To create or modify classes for all reports, you must modify a layout style sheet. In addition, some classes can be used to format Query Studio reports.

If you use the Microsoft Internet Explorer Web browser, version 6, some color gradients used in the 10.x default report styles are not supported.

Modify the report style
You can define whether your report uses the default styles for this version or the styles from a previous version.

Procedure
1. From the File menu, click Report Properties.
2. Click Report styles and select one of the following options:
   - To work with classes in the default style sheet, click 10.x styles.
   - To work with classes that were used in IBM Cognos 8, click 8.x styles.
     Use 8.x report styles when you are working with reports created in IBM Cognos 8 and you want to preserve their original appearance.
   - To work with classes that were used in IBM Cognos ReportNet, click 1.x styles.
     Use 1.x report styles when you are working with reports created in ReportNet and you want to preserve their original appearance.
   - To work with classes that have minimal styling defined, click Simplified styles.
     This option is useful when creating financial reports.

Modify object styles
You can change global classes to modify the style of all objects in a report or you can modify local classes and apply them to specific objects.

Procedure
1. Pause the pointer over the page explorer button and click Classes.
2. To create a new class, from the Toolbox tab, drag Class to the Local Classes pane.
3. To modify an existing class, in the Local Classes or Global Class Extensions pane, click the class.
   Modify a global class to apply a change to all objects that use that class. For example, if you modified the style List column title cell, all column titles in lists will reflect your modifications.

   Tip: Ctrl+click classes to make the same change to more than one class.
4. In the Properties pane, modify the properties to specify your desired formatting.

   Tip: Look at the Preview pane to preview your changes for different report objects, such as blocks, table cells, and text items.
   If you modify a global class, a pencil symbol appears beside the global class icon to indicate that the class was modified.
5. Apply the class to objects:
   - Pause the pointer over the page explorer button and click a report page.
   - Click an object to which to apply a class.
   - In the Properties pane, double-click the Class property.
   - Click the classes to apply from the Local classes and Global classes panes
     and click the right arrow button ➔.
   - If you applied more than one class, in the Selected classes pane, specify the
     order in which the classes are applied using the up and down arrow buttons.
     Classes in the Selected classes pane are applied from top to bottom. The
     style properties from all classes are merged together when they are applied.
     However, if the classes have style properties in common, the properties from
     the last class applied override those from previous classes.

(Don't Print) Class

The (Don't Print) class allows HTML items to display in the Web browser but not

to print.

The (Don't Print) class behaves as follows in the various report output formats:
  - HTML
    The Web browser defines the behavior. The HTML standard is that the item
    appears on the screen in the Web browser but is excluded by the print operation
    of the Web browser.
  - PDF
    The item is excluded from the output.
  - Microsoft Excel 2002 spreadsheet software
    The class is specified in the HTML that IBM Cognos uses to render Microsoft
    Excel 2002 output. However, Microsoft Excel 2002 does not appear to honor it
    and displays the item as missing some or all other style definitions.
  - Microsoft Excel 2007 spreadsheet software
    The item is excluded from the output.
  - CSV
    The item is included in the output.
  - XML
    The item is included in the output.

An alternative way to consistently exclude a report object from HTML, PDF and
Microsoft Excel output is to set the Box Type property for the object to None. CSV
and XML report outputs will still contain the object.

Modify Classes to Format Query Studio Reports

Some global classes are specific to IBM Cognos Query Studio or can be applied to
Query Studio reports. You can modify the following classes to format Query Studio
reports.

To format a Query Studio report using the modified classes, the template must be
applied to the report. For more information about applying a template to a Query
Studio report, see the Query Studio User Guide.
## Add Color to an Object

You can add background and foreground color to objects in the report. The foreground color applies to the text within objects.

**Tip:** The TOC Report sample report in the GO Data Warehouse (query) package includes objects with color. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

### Procedure

1. Select the object.
   - **Tip:** To quickly select the parent of an object, click the object, and then click the select ancestor button in the title bar of the Properties pane.
2. Do one of the following:
   - Click the background color button or the foreground color button.
   - In the Properties pane, double-click the Background Color or Foreground Color property.
3. To apply an existing color, click the Named Colors tab or Web Safe Colors tab and choose one of the available colors.
   - Use a named color to select a color from a small set of colors. Use Web safe colors to select from 216 available colors.
4. To apply a custom color, click the Custom Color tab and type values in the Red, Green, and Blue boxes.
   - The values must be hexadecimal.

### Related tasks:

“Add Background Effects to a Chart Object in a Legacy Chart” on page 116

You can change the look of certain charts and chart objects by applying visual effects such as drop shadows, borders, fills, texture effects, and bevel effects.
Chapter 13. Working With Your External Data

You can supplement your enterprise data with your own external or personal data file.

This allows you to create a report using an external file that contains data which is not part of your IBM Cognos Business Intelligence enterprise data. For example, you are an analyst and you receive a spreadsheet that contains what-if data about opening new branches of your retail store. You are asked to analyze the impact of these possible new branches on existing sales volumes. You link this scenario data with your enterprise data and create a professional report using IBM Cognos BI.

You import your own data file and start reporting on it right away. After importing, your external data file is protected by the same IBM Cognos security as your enterprise data, thus allowing you to report on your data in a secure and private environment.

You can use the following types of files:
- Microsoft Excel (.xls) spreadsheet software files
  IBM Cognos BI supports external data sources from Microsoft Excel up to version Microsoft Excel 2007.
  To review an up-to-date list of the environments that are supported by IBM Cognos products, including information about operating systems, patches, browsers, web servers, directory servers, database servers, and application servers, see [Supported Software Environments](http://www.ibm.com/support/docview.wss?uid=swg27037784).
- tab-delimited text (.txt) files
- comma-separated (.csv) files
- XML (*.xml) files
  Your XML files must adhere to the IBM Cognos schema specified at c10_location/bin/xmldata.xsd. For more information, contact your IBM Cognos administrator.
  This schema consists of a dataset element, which contains a metadata element and a data element. The metadata element contains the data item information in item elements. The data element contains all the row and value elements.
  For example, the following simple XML code produces a table with two columns (Product Number and Color) and two rows of data.
  ```xml
    <metadata>
      <item name="Product Number" type="xs:string" length="6" scale="0" precision="2" />
      <item name="Color" type="xs:string" length="18" scale="0" precision="8" />
    </metadata>
    <data>
      <row>
        <value>1</value> <value>Red</value>
      </row>
      <row>
        <value>2</value> <value>Blue</value>
      </row>
    </data>
  </dataset>
  
  To work with your external data, you start with an existing IBM Cognos package. You import data from your external file into the package and create links between data items in your file and data items in your enterprise data source. You then publish a new package that allows you to create reports that use your data and
your enterprise data, or reports that use only your data. You can link your external data with both dimensional and relational data sources.

Before you can import your own external data file, your IBM Cognos administrator must grant you permission for the Allow External Data capability found within the Report Studio capability, and you must have permission to use IBM Cognos Report Studio or IBM Cognos Workspace Advanced. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

External Data Packages

When you import external data into a package, you do not overwrite the original package. You create a new package that includes the original package, the new external data, and any links or relationships that you defined between the two. By default, the new package is saved in the My Folders area of IBM Cognos Connection, with External Data appended to the original package name. You can change where your package is published.

Note: The current content language is used to create the new package, and users cannot select the language for the package. For example, the content language in Cognos Connection is set to English. External data is imported into the GO Sales (query) package, and a new package called GO Sales (query) External Data is created. If the content language is changed to a different language, the package name is not translated, even though GO Sales (query) is a multilingual package.

IBM Cognos Samples

Sample external data sources in Microsoft Excel (.xls) format are provided with IBM Cognos BI. You can import these sample files into the Sample Outdoors Company sample reports found within the Cognos Workspace Advanced folder of the GO Data Warehouse (analysis) or GO Data Warehouse (query) package.

You can find the following external data source files on the server where IBM Cognos BI is installed in the c10_location/webcontent/samples/datasources/other directory.

- accounts.xls
- employee.xls
- organization.xls
- product_brand.xls
- product_color.xls
- product_line.xls
- product_name.xls
- product_size.xls
- product_type.xls
- products.xls
- promo_sets.xls
- promotions.xls
- region.xls
- retailers_site.xls
- time.xls

To obtain these files, contact your IBM Cognos administrator.
Preparing to Work with your External Data

To work with your external data, follow the process in this section.

There are four steps to preparing to work with your external data. The following diagram details these steps.

1. Prepare your external data file for import.
   Ensure that your external data file matches your enterprise data for your reporting needs. For example, if your external data file contains sales values by month, ensure that the formatting of months in your file matches the formatting used in your enterprise data source. Ensure that you can uniquely link at least one column from your external data file, such as product codes or years, with your enterprise data source.
   The maximum file size that you can import is 2.5 MB, with a maximum of 20000 rows. You can import a maximum of one external data file per package. Your IBM Cognos modeler can override these governors in IBM Cognos Framework Manager.

2. Import your external data.
   You import your external data file from your own computer or from a location on your network into an existing IBM Cognos package.
   A step-by-step wizard guides you through importing your data. If you want to create reports that contain data from both your external data file and your enterprise data source, you must link data items from the two data sources.
   You can import all or a subset of data columns from your external file.
   By adding external data, you extend an existing IBM Cognos package definition with the new data items from your file and you create a new package.

3. Create reports with your external data file.
   After you import and link your external data, it appears as a new namespace in the data tree of the Source tab and is integrated with the IBM Cognos content.
   You can then create reports with your data and perform any operation, such as filtering, sorting, grouping, or adding calculations. When you run the report, it uses data items from your external data file.
   You can save reports that contain your external data within the My Folders area of the IBM Cognos portal.

4. Determine whether you want to share your reports that use external data with other people in your organization. If you decide to share, take into account these considerations.
Working with Date Data

If the data that you import contains dates, ensure that the dates use the format yyyy-mm-dd.

Working with Dimensional Data Sources

If your enterprise data source is dimensional, such as OLAP or dimensionally-modeled relational, and you want to link your external data with your enterprise data, you must first create a tabular or list report. Create and save a list report with your enterprise data that contains the data items that you want to use to link with your external data. Ensure that you remove the aggregate rows that are automatically added in the footer of the list. This list is a projection of your dimensional data source.

When you import your external data, use the list report that you created to link your external data with the query subject from your enterprise data.

Data in your external file is relational by nature because it consists of tables and rows. If your enterprise data source is dimensional, you can still import and work with your external data. However, you cannot mix relational data from your external data files, and dimensional data from your enterprise data source within the same query. For example, a data container, such as a list, crosstab, or chart, uses one query and you cannot mix relational and dimensional data within the same list, crosstab, or chart. Doing so will cause an error.

If you want to use data from both the external data file and the original package within the same query, you must link the external data to a query subject within the current package instead of another report.

Working with External Data in an Unsecured IBM Cognos Application

If your IBM Cognos application is not secured, and users can log on anonymously, you may encounter issues if multiple people import external data in the same package.

For example, Robert imports his external data into package A and saves the package and reports he created in My Folders. Then, Valerie also imports her external data into the same package A and saves the package in My Folders. Valerie has therefore overwritten Robert's external data in package A. Now, if Robert tries to run one of his reports, he encounters errors because his external data is no longer in package A.

To avoid this problem,
• save packages that contain external data with a unique name.
• apply security to your IBM Cognos applications so that users do not share the same My Folders area.

Import Data

You select the file to import from your own computer, or from your local area network.

You select which columns to import.
You then specify a namespace to use. The namespace provides a unique name to associate with the data items that you import. The namespace appears in the data tree in the **Source** tab and is used to organize the data items. By default, the namespace is the imported file name without the extension.

If you change the default name for the namespace, you are prompted to select the external data file each time you run the report. To avoid this, select the **Allow server to automatically load file** check box.

**Procedure**

1. From the **Tools** menu, click **Manage External Data**.
   
   **Tip:** You can also click the manage external data button at the top of the **Source** tab.

2. On the **Select Data** page of the wizard, under **External data file**, click **Browse** and select your external data file to import.
   
   If you want the server to load the file without prompting users when they run the report, select the **Allow the server to automatically load the file** check box.
   
   If selected, you must use the Universal Naming Convention (UNC) path, such as `\servername\filename` and you must ensure that the IBM Cognos server has access to the file.

3. Under **Data items**, select the check box for the data items that you want to import.

4. Type a name for the namespace and click **Next**.
   
   The namespace appears in the **Source** tree, and identifies the external data within the package. By default, the name is the name of your imported external data file.

5. If you do not want to link your data or change the data attributes, click **Finish** now.

**Related tasks:**

*MSR-PD-0012 error when importing external data* on page 483

When you try to import an external data file, you receive an MSR-PD-0012 error.

*MSR-PD-0013 error when importing external data* on page 483

When you try to import an external data file, you receive an MSR-PD-0013 error.

---

**Map Data**

If you want to create reports that contain data from both your external file and from your enterprise data, you must link at least one query subject from your package or from an existing report to a data item in your external data. This mapping creates a relationship between your external data and your enterprise data. For example, your external data contains information about employees, including an employee number. You map the employee number from your external data file to the employee number in your enterprise data. This ensures that your data is integrated smoothly.

**About this task**

Mapping a data item in your external data to a query subject that references other query subjects is not supported. For example, the query subject cannot contain a
calculation that references a query item from another query subject. Such a
mapping produces the following error message:

> MSR-PD-0001 Failed to relate external data to objects in the underlying package. The item
> [query item] does not lead to a query subject.

**Procedure**

1. On the **Data Mapping** page, link existing query subjects in your enterprise data
to data items in your external data file.

2. Under **Existing query subject / report**, click the ellipsis (...) button and do one
of the following:
   - If you want to select a query subject from the data tree, click **Choose Query
   Subject** and select a query subject.

   **Note:** This option is not available when working with dimensional packages.
   - If you want to select from the query subjects included in a report, click **Choose a Report**
   and select a report.

3. Click the two data items that you want to link and click the **New link** button.
   You cannot link a data item in your external data to a calculation in a query
   subject.
   A link appears linking two data items.

   **Tip:** You can create multiple links for multiple data items. To delete a link,
   select the link and click **Delete Link**.

4. Click **Next**.

**Finish Importing Data**

You can change how query items from your external data file appear after they are
imported into IBM Cognos Business Intelligence. For example, you can change the
number of decimal places or the default summary.

If you want to use numeric data from your external data source as a measure in a
crosstab, you must assign that data item a default summary other than
**Unsupported**. A specified default summary makes the data item appear as a
measure in the data tree in the **Source** tab. Otherwise, if you add the data
item with an **Unsupported** default summary as the measure in a crosstab, no
values appear.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data type</td>
<td>Identifies whether data in the file is one of the following types:</td>
</tr>
<tr>
<td></td>
<td><strong>Integer</strong>, which represents numeric values.</td>
</tr>
<tr>
<td></td>
<td><strong>Decimal</strong>, which represents integer values that are scaled by a</td>
</tr>
<tr>
<td></td>
<td>variable power of 10.</td>
</tr>
<tr>
<td></td>
<td><strong>Text</strong>, which represents values that contain letters and symbols.</td>
</tr>
<tr>
<td></td>
<td><strong>Date</strong> or <strong>Date Time</strong>, which represent dates and times.</td>
</tr>
<tr>
<td>Default summary</td>
<td>Identifies Sum, Average, Max, Min, Count or Unsupported as the</td>
</tr>
<tr>
<td></td>
<td>default type of summary for the data item.</td>
</tr>
<tr>
<td></td>
<td>Applies only to <strong>Integer</strong> and <strong>Decimal</strong> data types.</td>
</tr>
</tbody>
</table>
**Attribute** | **Description**
---|---
Decimal places | Specifies the number of decimal places for the data item. Applies only to the Decimal data type.

If you mapped links between data items in your external data and data items in your enterprise data, specify the options that define the relationships between the data items.

For each data item that you import and link, specify whether values are unique or exist more than once in both your external data and in your enterprise data. You can also specify how to handle rows that contain missing values in the report results.

**Procedure**
1. On the Data Attributes page, specify the attributes for each data item after it is imported and click Next.
   For example, if you import numeric data items, you can change the default summary and number of decimal places.
2. On the Mapping Options page, specify the relationships between the linked data items and how to process the results in the report output.
3. Click Finish.

**Publish the Package**
You can change the name and location of the package with your external data to help you differentiate between data packages.

**Procedure**
1. If you want to rename the package that will be published or change the location where it is published, do the following:
   - In the Manage External Data dialog, under Package name, click the ellipsis (...) button.
   - Type a new name for the package and select the location where to save it.
   - Click Save.
2. Click Publish.

**Results**
IBM Cognos Business Intelligence imports your external data into a new package. A namespace with the data items from your external file appears in the data tree in the Source tab.

You can now create reports with your external data.

**Edit Your External Data**
After you import your data, you can change the data mappings and options that you originally specified and republish the package.

You can
• rename the namespace, which renames the organizational folder that appears in
   the data tree in the Source tab. If you change the namespace, you are
   prompted for the external data file when you run the report.
• change which columns to import
• change the data mapping links
• change the data attributes
• change the mapping options

You can also import multiple external data files into the same package. To do this,
your data modeler must modify governors in the model and republish the package
that contains your enterprise data. For more information, see the IBM Cognos
Framework Manager User Guide.

**Procedure**

1. From the Tools menu, click Manage External Data.
2. In the Manage External Data dialog box, select the external data to edit and
   click the edit button.
3. In the left pane, select the options that you want to change.
4. Click OK and then republish the package.

**Results**

IBM Cognos Business Intelligence re-imports your external data and updates the
data items that appears in the data tree in the Source tab.

You can now create and update reports with your external data.

**Delete Your External Data**

You can delete your external data from within the package that you created if you
no longer need it.

**Procedure**

1. From the Tools menu, click Manage External Data.
2. In the Manage External Data dialog box, select the external data package to
   delete and click the delete button.
3. Click Publish.

**Results**

The external data namespace is removed from the package.

If you also no longer require the external data package or any reports created with
the package, you can delete it from within IBM Cognos Connection.
Running a Report That Contains External Data

Reports that contain external data run the same way as reports that contain only enterprise data. If you have access to the report, you will also have access to the external data included within the report.

You may be prompted to select the location of the external data file when you run the report if either
• the report author did not specify to automatically load the file.
• the IBM Cognos Business Intelligence server cannot locate the file.

However, you are not re-prompted within the same Web browser session.

You can determine whether data in a report uses external data by tracing its lineage. For more information, see “View Lineage Information for a Data Item” on page 53.

Making your Reports Public

After you create a report that uses your external data, you may want to make it public to share it with coworkers. They can run your report using your external data file that you made available on a public network drive that the IBM Cognos server can access. They can also use their own version of the file. If they use their own version, the file must contain the same columns as your original external data file that you used to import the data and create the report. In addition, you must clear the Allow server to automatically load file check box in the Select Data page of the Manage External Data wizard.

To make reports public, you must save them in the Public Folders area of the IBM Cognos portal. To save content in Public Folders, you must have the appropriate permissions. Contact your IBM Cognos administrator to obtain permissions and to inform him or her that you are sharing a package or files.

If you share your reports, ensure that you maintain the reports.
Chapter 14. Using Conditions

You can define conditions to control what users see when they run a report. Conditions can apply to specific items in a report. For example, you can define a conditional style to highlight exceptional data, such as product revenue that exceeds your target.

Conditions can also apply at the report layout level. Conditional layouts are useful for delivering reports to a multilingual audience. For example, you can have text items, such as titles and cover pages, appear in the same language as the data in the report.

**Related concepts:**
“Creating a Conditional Color Palette in a Chart” on page 111

You can create a conditional palette to color data items in your chart in different ways depending on a condition. For example, in a column chart that shows revenue per month, you want to make the columns for the months that have a revenue greater than $1000000 green.

**Highlight Data Using a Conditional Style**

Add conditional styles to your report to better identify exceptional or unexpected results. A conditional style is a format, such as cell shading or font color, that is applied to objects if a specified condition is true.

For example, you want to automatically highlight in green the departments in your organization that meet their budget quotas and highlight in red the departments that go over budget. Creating conditional styles color-codes information in your reports so that you can find areas that need attention.

You can apply multiple conditional styles to objects. For example, you can apply one style in specific cells and another style for the overall report. If multiple styles set the same property, such as font color, the last style in the list is applied.

You can apply conditional styles based on any data item in your report.

You can create the following types of conditional styles.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric Range</td>
<td>Highlights straight numerical data, such as revenues and losses.</td>
</tr>
<tr>
<td>Date/Time Range</td>
<td>Highlights data from specific dates and times.</td>
</tr>
<tr>
<td>Date Range</td>
<td>Highlights data from specific dates.</td>
</tr>
<tr>
<td>Time Range</td>
<td>Highlights data from specific times.</td>
</tr>
<tr>
<td>Interval</td>
<td>Highlights data falling between set intervals.</td>
</tr>
</tbody>
</table>
### Type Description

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Highlights specific alphanumeric items in a report. For example, you can highlight all instances of a specific word or phrase, such as Equipment. String criteria are case-sensitive. If multiple string conditions are met, only the first conditional style is applied.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Creates conditional styles that use calculations or expressions. If multiple advanced conditions are met, only the first conditional style is applied.</td>
</tr>
</tbody>
</table>

You can also apply a conditional color palette to a chart.

**Note:** Conditional styles and conditional data formatting do not work on chart axis labels. If you apply a conditional style or conditional data formatting to a chart axis, only the first style defined is applied.

You can perform a search to find objects in your report that use conditional styles. You can also view all the conditional styles used in your report to delete or modify them.

You can also use variables to highlight data. If a report contains both conditional styles and style variables, the style variables are applied first and then the conditional styles are applied.

**Tip:** The Return Quantity by Order Method sample report in the GO Data Warehouse (analysis) package includes conditional highlighting. For more information about The Sample Outdoors Company samples, see [Appendix C, “Sample Reports and Packages,” on page 507](#).

### Create a New Conditional Style

You can apply conditional styles based on any data item in your report.

**Procedure**

1. Click the object for which you want to define a conditional style and click the conditional styles button.

   **Tip:** You can also right-click the object and click **Style, Conditional Styles** or click the object, and then, in the **Properties** pane, set the **Conditional Styles** property.

2. Click the add button and click **New Conditional Style**.

3. Select the data item to determine the condition and click **OK**. The type of conditional style that you can use depends on the type of data item you select.

4. In the **Name** box, type a name for the conditional style.

5. To define a numeric value, date/time, date, time, or interval condition:
   
   - Click the new button and select a value to define a threshold.
The value appears in the **Range** column, and two ranges are created.  
• For each range, under **Style**, click one of the predefined styles to apply to the range or click the edit button and create a new style.

**Tip:** You can also define a style for the cells in your report that have missing values.  
• Repeat the steps above to add other conditions.

**Tip:** Under **Style**, pause the pointer over each range to see the condition produced for that range.

• To move a value above or below a threshold, click the arrow button next to the value.  
  For example, you insert a threshold value of five million. By default, the ranges are less than or equal to five million and greater than five million. Moving the five million value above the threshold changes the ranges to less than five million and greater than or equal to five million.

6. To define a string condition:  
• Click the new button and select how to define the condition.  
• To select more than one individual value, click **Select Multiple Values** and click the values.  
• To type specific values, click **Enter Values** and type the values.  
• To specify your own criteria, such as values that begin with the letter A, click **Enter String Criteria** and specify the condition.  
• For each condition, under **Style**, click one of the predefined styles to apply or click the edit style button and create a new style. Specify the style to apply to remaining values by clicking one of the predefined styles beside **Remaining values (including future values)**.  
• Specify the order in which to evaluate the conditions.  
  Conditions are evaluated from top to bottom, and the first condition that is met is applied.

**Reuse an Existing Conditional Style**

You can create a conditional style once and reuse it on multiple objects in your report. You can specify the order in which conditional styles are applied. You can also use existing local classes as your conditional styles.

**Procedure**

1. Click the data item for which you want to define a conditional style, and then click the conditional styles button.

**Tip:** You can also right-click the data item and click **Style, Conditional Styles** or click the data item, and then, in the **Properties** pane, set the **Conditional Styles** property.

2. Click the add button, click **Use Existing Conditional Style**, and select the style.
Create an Advanced Conditional Style

You can create advanced conditional styles that use calculations or expressions.

If multiple advanced conditions are met, only the first conditional style is applied.

Procedure

1. Click the data item for which you want to define a conditional style, and then click the conditional styles button.

   Tip: You can also right-click the data item and click Style, Conditional Styles or click the data item, and then, in the Properties pane, set the Conditional Styles property.

2. Click the add button and click Advanced Conditional Style.

3. Type a name for the conditional style.

4. Click the add button and specify the expression that defines the condition.

5. For each condition, under Style, click one of the predefined styles to apply or click the edit button and create a new style. Specify the style to apply to remaining values by clicking one of the predefined styles beside Remaining values (including future values).

6. Specify the order in which to evaluate the conditions by clicking a condition and then clicking the move up or move down arrow.
   Conditions are evaluated from top to bottom, and the first condition that is met is applied.

Manage Conditional Styles

You can view, modify, or delete the conditional styles that are applied to your report. You can also define a new conditional style.

In the IBM Cognos Report Studio options, you can specify whether to automatically delete conditional styles that are no longer used in a report.

Procedure

From the Tools menu, click Manage Conditional Styles.

Example - Add a Conditional Style to an Existing Report

You are a report writer at The Sample Outdoors Company, which sells sporting equipment. You have a report that compares current year data to previous year data and highlights negative percentage variances in red (Poor) and positive percentage variances in green (Excellent). You want to add a third conditional style to indicate percentage variances above 0 but less than 10. You create a conditional style that highlights percentage variances between 0 and 10% in yellow (Average).

Procedure

1. Open the GO Balance Sheet as at Dec 31 2012 report from the GO Data Warehouse (analysis) package.

2. Right-click any cell in the % Variance column and click Style, Conditional Styles.
3. Select **Conditional Style 1** and click the edit button.
4. Select the first advanced condition listed and click the edit button.
5. Delete =0 from the expression definition.
6. Click the **Functions** tab, and then expand the **Operators** folder.
7. Double-click **between**, and then click after **between** in the expression, add a space, and type 0.
8. Double-click **and**, and then click after **and** in the expression, add a space, type .1, and click **OK**.
9. From the **Style** box associated with this condition, click **Average**.
10. Select the second advanced condition listed and click the edit button.
11. Replace >0 in the expression with >.1 and click **OK**.
12. Leave the style associated with this condition as is.
13. Run the report.

The new conditional style appears in the **% Variance** column.

---

### Highlight Data Using a Style Variable

Highlight data in your report to better identify exceptional results. For example, you want to identify sales representatives who have exceeded their quota. You create a condition that checks whether each representative's sales for the year is greater than their quota for the year.

Style variables are useful if you are working with reports that were created in a previous version of IBM Cognos Business Intelligence or if you want to use language variables to specify conditional styles.

You can also use conditional styles to highlight data. If a report contains both conditional styles and style variables, the style variables are applied before the conditional styles.

**Procedure**

1. **Create a variable** and define the condition that determines if the data will be highlighted.
2. In the work area, click the column to highlight based on the condition.
3. In the **Properties** pane, double-click the **Style Variable** property.

4. Click **Variable**, click the variable to assign to the object, and click **OK**.

5. If you assigned a string variable, in the **Values** box, select the values for the condition to support.

   **Tip:** A default value exists for the variable, and it is always selected.

6. If you assigned a language variable, in the **Values** box, select the languages for the condition to support.

   **Tip:** A default value exists for the variable, and it is always selected.

7. Click **OK**.

8. Pause the pointer over the condition explorer button and click a value other than the default value.

   **Tip:** When you select a value in the condition explorer, the Explorer bar becomes green to indicate that conditional formatting is turned on and that any changes you make to the report apply only to the variable value.

   For example, if you created a Boolean variable, click **Yes**.

9. In the **Properties** pane, specify the formatting with which to highlight the column when the condition is satisfied.

   For example, click the **Border** property to create a thicker border around the column.

10. Repeat steps 8 to 9 for other possible values defined for the variable.

    **Tip:** To view the report with no variables applied, pause the pointer over the condition explorer button and click **(No variable)** or triple-click the Explorer bar.

**Results**

When you run the report, the report objects to which you applied the variable are highlighted when the condition is satisfied. For example, if you created a Boolean variable, the objects are highlighted when the condition is met. If the condition is not satisfied for any object, no conditional formatting is applied.

---

**Adding Conditional Rendering to a Report**

You can specify which objects are rendered when a report is run.

Before you can add conditional formatting or conditional rendering to your report, you must add a variable. You can create a variable in the condition explorer or in the **Properties** pane.

**Add a Variable from the Condition Explorer**

Before you can add conditional formatting or conditional rendering to your report, you must add a variable. You can create a variable in the condition explorer or in the **Properties** pane.

**Procedure**

1. Pause the pointer over the condition explorer button and click **Variables**.
2. From the Toolbox tab, drag one of the following variables to the Variables pane:
   - To create a variable that has only two possible values, Yes and No, drag Boolean Variable.
   - To create a variable whose values are string-based, drag String Variable.
   - To create a variable whose values are different languages, drag Report Language Variable.

3. If you created a Boolean variable, in the Expression Definition box, define the condition and click OK.
   For example, the following expression returns the value Yes if revenue is less than one million and the value No if revenue is greater than or equal to one million:
   
   \[
   \text{[Revenue]} < 1000000
   \]
   For information about creating expressions, see “Using Relational Calculations” on page 248 or “Using Dimensional Calculations” on page 302 and Appendix F, “Using the expression editor,” on page 555.

4. If you created a string variable, do the following:
   - In the Expression Definition box, define the condition and click OK.
   For example, the following expression returns the value high if revenue is greater than one million and the value low if revenue is less than or equal to one million:
   
   \[
   \text{if (}[\text{Revenue}] > 1000000) \text{ then } \text{(high)} \text{ else } \text{(low)}
   \]
   For information about creating expressions, see “Using Relational Calculations” on page 248 or “Using Dimensional Calculations” on page 302 and Appendix F, “Using the expression editor,” on page 555.
   - Click the add button in the Values pane.
   - For each value that the variable can assume, type the name of the value that corresponds with the possible outcomes defined in the expression.
   For example, in the previous expression, you must create two values for the variable, high and low.

   **Tip:** You can create a group by clicking two or more values and then clicking the group values button. For example, you can create a group that includes the available French languages.

5. If you created a language-specific variable, in the Languages dialog box, select the languages to support.

**Add a Variable from the Properties Pane**

Before you can add conditional formatting or conditional rendering to your report, you must add a variable. You can create a variable in the condition explorer or in the Properties pane.

**Procedure**

1. Select the report object.
2. In the Properties pane, under Conditional, double-click the conditional property to which to assign the variable.
   The following conditional properties are available:
<table>
<thead>
<tr>
<th>Goal</th>
<th>Conditional Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a variable based on which text can be conditionally shown. For example, you want different text to appear when a report is run in a different language.</td>
<td>Text Source Variable</td>
</tr>
<tr>
<td>Specify a variable based on which object can be conditionally rendered. For example, you want to make a revenue report smaller by not rendering rows that are below a threshold.</td>
<td>Render Variable</td>
</tr>
<tr>
<td>Specify a variable based on which object can be conditionally styled. For example, you want data that meets some criterion to appear in a different color.</td>
<td>Style Variable</td>
</tr>
<tr>
<td>Specify a variable based on which objects in a block can be conditionally rendered. Applies only to conditional block objects that you insert in a report.</td>
<td>Block Variable</td>
</tr>
</tbody>
</table>

3. In the **Variable** box, click an existing variable or one of the following variable types:
   - `<New language variable>`
   - `<New string variable>`
   - `<New boolean variable>`

4. In the **New Variable** dialog box, in the **Name** box, type the name of the variable.

5. If you created a string variable, click the add button, type the string values to define, and click **OK**.

6. If you created a language variable, select the languages to support and click **OK**.

7. In the **Expression Definition** box, define the condition.

**Hide or Show an Object**

You can hide and show objects in a report based on a condition you define.

You can also specify that an object should not be rendered based on a condition.

**Tip:** The Global Bonus Report sample report in the GO Data Warehouse (analysis) package includes hidden objects. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

**Procedure**

1. **Create a variable** and define the condition that determines if the object is shown or hidden.

   **Tip:** Create a Boolean variable to show and hide objects, as this type of variable has only two possible values.
2. From the Toolbox tab, drag a Conditional Blocks object to the work area.
3. Select the conditional block.
4. In the Properties pane, double-click the Block Variable property.
5. In the Variable box, click the variable you created and click OK.
6. Set the Current Block property to Yes.
7. From the content pane, drag the object to show or hide to the conditional block.
   For example, drag a data item from the Source tab or from the Data Items tab.
   You may need to link the report page to a query before you can add a data item to the block.

**Results**

When you run the report, the report objects to which you applied the variable are visible when the condition is satisfied and invisible when it is not.

**Add Conditional Rendering**

Add conditional rendering to specify which objects are rendered when a report is run. This is useful when your report contains sensitive data.

Conditional rendering is not the same as hiding objects. When you hide an object, the object exists but is transparent. If an object is not rendered, it is not in the report.

For a list of objects that can be rendered conditionally, see the Render Variable property in Appendix G, “Report Studio Object and Property Reference,” on page 771.

**Procedure**

1. Select the list column to be rendered conditionally.

   **Tip:** You must select the list column, not the list column body or the list column title. If the body or title is selected, as indicated in the Properties pane, click the select ancestor button and click the list column.

2. In the Properties pane, double-click the Render Variable property.

3. Click Variable and click the variable that will determine if the column will be rendered.

4. In the Render for box, select the values that the condition will support.

   **Tip:** A default value exists for the variable, and it is always selected.

**Example - Create a Conditional Report**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a report that shows orders after a date specified by the user. The report will prompt the user for a date and ask whether the user wants to see a description for each order.

**Procedure**

1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the **New** dialog box, click **List** and click **OK**.

4. From the **Source** tab, expand **Sales and Marketing (query)**, and **Sales (query)** and add data items to the list:
   - Expand **Time dimension** and add **Date**.
   - Expand **Sales order** and add **Order number**.
   - Expand **Product** and add **Product name** and **Product description**.
   - Expand **Sales fact** and add **Quantity**, **Unit price**, and **Revenue**.

5. Click **Date**, and then click the section button.

6. Select the **Order Number** column and click the group button.

7. Click **Revenue**, and then click the summarize button and click **Total**.

8. Change the title of the report to **New Orders**.

9. Pause the pointer over the page explorer button and click **Prompt Pages**.

10. Create a new prompt page by double-clicking **Page** in the **Toolbox** tab.

11. Double-click the new prompt page.

12. From the **Toolbox** tab, double-click **Text Item** and type the following text:
    
    *Enter the start date, and select if descriptions will be shown.*

13. Insert a 2 by 2 table into the prompt page by clicking the insert table button and moving the pointer until four squares are highlighted in a 2 by 2 pattern.

14. From the **Toolbox** tab, drag a **Text Item** into the upper-left cell and type the following text:
    
    **Starting Date**

15. From the **Toolbox** tab, drag a **Text Item** into the lower-left cell and type the following text:
    
    **Show Descriptions**

16. From the **Toolbox** tab, drag a **Date Prompt** into the upper-right cell.

17. In the **Prompt Wizard** window, select **Create a new parameter**, type **p_Date** in the space provided, and then click **Next**.

18. In the **Create Filter** window, select **Create a parameterized filter** with the following entries:
    - For **Package item**, click the ellipsis (...) button and open **Sales (query)** and **Time dimension**, and then click **Date**.
    - For **Operator**, click >.

19. Click **Finish**.

20. From the **Toolbox** tab, drag a **Value Prompt** into the lower-right cell.

21. In the **Prompt Wizard**, in the **Choose Parameter** window, select **Create a new parameter**, type **p_ShowDesc** in the space provided, and then click **Finish**.

22. Select the **Value Prompt** and, in the **Properties** pane, double-click **Static Choices**.

23. Click the add button.

24. In the **Edit** dialog box, type **Yes** in both the **Use** and **Display** boxes.

25. Click the add button.
26. In the Edit dialog box, type No in both the Use and Display boxes.
27. Click OK.

28. Pause the pointer over the condition explorer button and click Variables.
29. From the Toolbox tab, create a new Boolean variable by double-clicking Boolean Variable.
30. In the Report Expression dialog box, type the following in the Expression Definition window and click OK:
   ParamDisplayValue("p_ShowDesc") = 'Yes'
31. In the Properties pane, set the Name property to showDesc.

32. Pause the pointer over the page explorer button and click the report page.
33. Click the Product descriptions column.
34. In the Properties pane, click the select ancestor button and click List Column.
35. In the Properties pane, set the Render Variable property to the showDesc Boolean variable you created.
36. Run the report.

Results

The report prompts you for a date. Orders that occur after the date you entered are shown. The report also asks whether to show the Descriptions column, and the column is rendered only if you choose Yes.

Add Multiple Layouts

Add multiple layouts to show a report in different ways. For example, you can define a different layout for each language in a multilingual report. This allows you to create a single report that can be viewed by report consumers that use different regional settings.

Procedure

1. Create a variable and define the condition that will be used for each layout.
   For example, create a report language variable that includes each language that requires a conditional layout.
   Note: Expressions used in a conditional layout cannot reference a query.
2. From the File menu, click Conditional Layouts.
3. Select a variable, and then select the values that require a separate layout.

Results

A layout is created for each value you selected. Use the page explorer to navigate the different layouts. For each layout, click Report Pages to create a report page or Prompt Pages to create a prompt page and add objects.

Tip: You can create new variables from the Conditional Layouts dialog. The variables are added to the condition explorer. For more information, see “Add a Variable from the Condition Explorer” on page 370.
Setting up a Multilingual Reporting Environment

You can create reports that show data in more than one language and use different regional settings. This means that you can create a single report that can be used by report consumers anywhere in the world.

The samples databases provided with IBM Cognos store a selection of text fields, such as names and descriptions, in more than 25 languages to demonstrate a multilingual reporting environment. For information about how data is stored in the samples databases and how the samples databases are set up to use multilingual data, see the Administration and Security Guide.

Here is the process for creating a multilingual reporting environment:

- Use multilingual metadata.
  The data source administrator can store multilingual data in either individual tables, rows, or columns.
  For more information about configuring your database for multilingual reporting, see the Administration and Security Guide.

- Create a multilingual model.
  Modelers use IBM Cognos Framework Manager to add multilingual metadata to the model from any data source type except OLAP. They add multilingual metadata by defining which languages the model supports, translating text strings in the model for things such as object names and descriptions, and defining which languages are exported in each package. If the data source contains multilingual data, modelers can define queries that retrieve data in the default language for the report user.
  For more information, see the Framework Manager User Guide.

- Create multilingual maps.
  Administrators and modelers use a Microsoft Windows operating system utility named Map Manager to import maps and update labels for maps in IBM Cognos Report Studio. For map features such as country or region and city names, administrators and modelers can define alternative names to provide multilingual versions of text that appears on the map.
  For more information, see the Map Manager Installation and User Guide.

- Create a multilingual report.
  The report author uses Report Studio to create a report that can be viewed in different languages. For example, you can specify that text, such as the title, appears in German when the report is opened by a German user. You can also add translations for text objects, and create other language-dependent objects.

- Specify the language in which a report is viewed.
  You can use IBM Cognos Connection to do the following:
  - Define multilingual properties, such as a name, screen tip, and description, for each entry in the portal.
  - Specify the default language to be used when a report is run.
    Tip: You can specify the default language on the run options page, in the report properties, or in your preferences.
  - Specify a language, other than the default, to be used when a report is run.
  For more information, see the IBM Cognos Connection User Guide.

The data then appears in the language and with the regional settings specified in

- the user's Web browser options
Create a Multilingual Report in Report Studio

You can create a report in IBM Cognos Report Studio that can be viewed in different languages. For example, you can specify that text such as the title appears in German when the report is opened by a German user. You can also add translations for text objects and create other language-dependent objects.

Before you begin

If you want the report to show data in different languages, the model must also be multilingual.

Procedure

1. Create a report language variable.
2. In the work area, select the object to modify based on a language.
3. In the Properties pane, double-click the Style Variable property.
   If you are changing the language of a text string, click Text Source Variable instead.
4. Click Variable and click the language variable you created.
5. In the Values box, select the languages for the condition to support and click OK.
   Tip: A default value exists for the variable, and it is always selected.
6. Pause the pointer over the condition explorer button and a language for the variable.
   Tip: When you select a value in the condition explorer, the Explorer bar becomes green to indicate that conditional formatting is turned on and that any changes you make to the report apply only to the variable value.
7. In the Properties pane, specify the formatting for the language.
   For example, to change the language of a text string, double-click the Text property and select the new string.
8. Press Enter when you are done.
9. Repeat steps 6 to 8 for all other languages specified for the variable.
   Tip: To view the report with no variables applied, pause the pointer over the condition explorer button and click (No variable) or triple-click the Explorer bar.

Results

When you run the report, the report objects to which you applied the variable are formatted according to the browser's language.
Chapter 15. Bursting Reports

Burst a report to distribute its contents to various recipients. Bursting is the process of running a report once and then dividing the results for recipients who each view a subset of the data. For example, salespeople in different regions each need a report showing the sales target for their country or region. You use burst reports to send each salesperson only the information they need. Burst reports can be distributed by email or saved to a folder for viewing in IBM Cognos Connection.

Tip: The Bursted Sales Performance Report sample report in the GO Data Warehouse (analysis) package includes bursting. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

To burst a report against a dimensional data source, see Creating Burst Reports Using Dimensional Data Sources” on page 383.

You cannot burst crosstab or chart reports. However you can burst a report where a container includes a crosstab or chart that is part of a master detail relationship. In this situation, you can burst only HTML, PDF, and XLS output formats; you cannot burst CSV or XML output formats.

For information about avoiding disk space exhaustion when bursting charts or crosstabs, see “Master Detail or Burst Reports with Charts or Crosstabs May Result in Denial of Service” on page 499.

Procedure
1. define burst recipients
2. specify burst groups
3. set burst options
4. enable bursting

Defining Burst Recipients

Define the recipients who will receive data when the report is run. You can distribute burst reports to individual users, groups, roles, distribution lists, and contacts.

To define the recipients, you create a calculated field, create the burst table in the source database, and then import the table into a package.

Create a Calculated Field

You can use a calculated field to dynamically create burst report recipients.

Procedure
1. Pause the pointer over the query explorer button and click the query that will produce the data to distribute.
2. From the Toolbox tab, drag Data Item to the Data Items pane.
3. In the **Expression Definition** box, type the expression that will generate the list of recipients and click **OK**.

For example, typing the following expression builds the list of the employees of The Samples Outdoors Company. The expression concatenates the first letter of each employee's first name with their last name.

```plaintext
lower(substring([Employee summary (query)].[Employee by
organization].[First name],1,1) + [Employee summary (query)].[Employee by
organization].[Last name])
```

4. To give the data item a more meaningful name, in the **Properties** pane, set the **Name** property to a different name and press the Enter key.

### Creating the Burst Table in the Source Database

You can create a burst table in the source database for the list of recipients. The steps you must follow depend on the database system you are using. The burst table must contain the following columns:

- A unique identifier
  - **Tip**: Some database systems do not require a unique identifier for each table.
- A recipient column
- The data item on which to burst

You can also include other columns that provide additional information. For example, if you plan to distribute reports by email, you can add a column for the email address of each recipient.

After you create the table, add the recipients who will receive the report. You can create a mixed recipients list that includes individual users, groups, roles, contacts, distribution lists, or email addresses. For example, a burst table may contain the following recipients.

<table>
<thead>
<tr>
<th>Recipient example</th>
<th>Recipient type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAMID(&quot;Canada&quot;)</td>
<td>Group</td>
</tr>
<tr>
<td>CAMID(&quot;?:&quot;)/contact[@name='Silvano Allessori']</td>
<td>Contact</td>
</tr>
<tr>
<td>CAMID(&quot;?:&quot;)/distributionList[@name='European Partners']</td>
<td>Distribution list</td>
</tr>
<tr>
<td>CAMID(&quot;LDAP_Local_ID :uid=gbelding,ou=people&quot;)</td>
<td>Authentication provider user or group, where LDAP_Local_ID is the name of an LDAP namespace ID, and people is the name of an organizational unit</td>
</tr>
<tr>
<td><a href="mailto:c10@ibmcosognos99.com">c10@ibmcosognos99.com</a></td>
<td>Email address</td>
</tr>
</tbody>
</table>

CAMID stands for Cognos Access Manager ID, and it represents an internal search path to the recipients. Specify search paths when you want to save burst reports in a folder. You can obtain the search path in IBM Cognos Connection by opening the **Set properties** page for each recipient and clicking **View the search path**. Ensure that you use the proper syntax when adding recipients to the burst table.

In the case of NTLM namespaces, user IDs in the search path use alphanumeric or numeric characters that make them difficult to read. You can use the following alternate search path syntax:

```plaintext
directory/namespace[@name="Local NT"]//account[@userName="gbelding"]
```
where Local NT is the name of a namespace and gbelding is the name of a user. The double slash before the account element indicates that you are searching all accounts under the specified namespace.

**Note:** If you have a mixed recipients list, do not mix email address recipients and alternate path recipients. Because the alternate path syntax contains the @ symbol, it will be mistaken for an email address.

For more information about users, groups, roles, contacts, and distribution lists, see the IBM Cognos Business Intelligence Administration and Security Guide.

### Importing the Table into a Package

After you create the burst table in the source database, you must add it to the package that you will use to create the report.

For more information about importing tables and creating relationships, see the Framework Manager User Guide.

**Procedure**

1. Open the package.
2. Import the table.
3. Define the relationship between the burst table and the table containing the data item on which to burst.
   
   For example, you are bursting on country or region code. You define a relationship between country or region code in the burst table and country or region code in the Country or Region table.
4. Save and publish the package.

### Specify a Burst Group

Specify burst groups to set how the report will be distributed. Burst groups are defined by a data item that you create in the report or that you add from the burst table.

**Procedure**

1. Pause the pointer over the query explorer button and click the query that will produce the data to distribute.
2. If you are creating a data item, do the following:
   
   • From the Toolbox tab, drag Data Item to the Data Items pane.
   • In the Expression Definition box, type the expression that defines the burst key.
     
     For example, the following expression builds an email address for each sales representative in The Sample Outdoors Company. The expression incorporates the calculated field that was previously created, which is named userID below, with ibmcognos99.com as the domain name.

     ```
     [userID]+'@ibmcognos99.com'
     ```

     **Tip:** To give the data item a more meaningful name, in the Properties pane, set the Name property to a different name and press Enter.
3. To specify a burst table column as the data item, do the following:
• From the Source tab, expand the burst table.
• Drag the data item to the Data Items pane.
  For example, if you are bursting reports by email, drag the data item containing email addresses.

Set Burst Options

Set burst options for the report to indicate the data item on which to burst and the recipients.

Before you begin

Before you set burst options, ensure that the data item on which you intend to burst is in the report and is grouped. The grouped column will create the appropriate subsets of data. In addition, you must associate the burst key with this level of grouping.

Procedure

1. From the File menu, click Burst Options.
2. Select the Make report available for bursting check box.
3. Under Burst Groups, in the Query box, click the query that contains the data item on which to burst.
   
   Tip: You can choose a query that does not appear in the layout. This is useful to distribute the same report to all burst recipients.
4. In the Label box, click the data item with which to label each burst report.
5. Click the edit button.
6. In the Data Items box, drag the data item on which to burst to the Groups folder and click OK.
   
   Tip: You can specify the sort order of data within each group by dragging data items to the Sort List folder and then clicking the sort order button.
7. Under Burst Recipient, in the Query box, click the query that contains the data item to be used as the distribution list.
8. In the Data Item box, click the data item that contains the recipients.
9. In the Type box, choose the method to burst the report:
   • Click Automatic to let IBM Cognos Business Intelligence determine from the data item whether to email reports or send them to folders in IBM Cognos Connection.
   • Click Email addresses to distribute reports by email.
   • Click Directory entries to distribute reports to folders that recipients can access in IBM Cognos Connection.

Note: To burst reports to multiple mobile device users, you must choose to distribute reports to folders. You can choose Directory entries, or you can choose Automatic if the data item returns directory entries instead of email addresses. To view the reports, the recipients must have IBM Cognos Mobile installed on their mobile devices. For more information about IBM Cognos Mobile, see the IBM Cognos Mobile Installation and Administration Guide.
When recipients log into IBM Cognos BI, they will see only the report that is specific to them.

10. If the report contains two nested data containers, such as a list and a chart, click the ellipsis (...) button beside Master detail relationships and define the relationship between the data containers.

For information about master detail relationships, see "Create a Master Detail Relationship" on page 292.

Enable Bursting

When the report is ready to be distributed, enable bursting for the report in IBM Cognos Connection.

Procedure

1. Locate the report in IBM Cognos Connection.
2. Under Actions, click the run with options button.
3. Click advanced options on the right of the screen.
4. Under Time and mode, click Run in the background.
5. Select the Burst the report check box.
6. If you are distributing reports by email, select the Send the report by email check box.

Tip: If you are bursting the report to a folder, you can also send the report by email if the Send the report by email check box is selected. Reports will be emailed if the recipient's email address is stored in the authentication source you are using or if they are entered in the recipient's personal information in IBM Cognos Connection.

7. If the burst report contains a drill-through link to another report and you are distributing the burst report by email, do the following:
   - Click Edit the options.
   - Select the Include a link to the report check box.

If you do not select the check box, the drill-through links in the burst report will not work.

8. Run the report.

Results

Allow a few moments for the report to run. If you are an administrator, you can view all outputs for the report. Under Actions, click View the output versions for this report. When burst recipients log into IBM Cognos Connection or access their email accounts, they will see only the data that is meant for them.

Creating Burst Reports Using Dimensional Data Sources

You can burst a report using a dimensional data source by using burst information that is stored in the data source. Because you do not want to append bursting information to existing dimensional data sources, you can create a relational data source that contains the burst information.

Report bursting is limited when the underlying data source is a cube (MOLAP data source such as IBM Cognos PowerCube, Microsoft Analysis Services, Oracle Essbase, or IBM DB2/OLAP). The burst report must be a grouped report, and the
burst is restricted to the outermost grouping in the report. For example, if you have a report grouped on Country or Region and State, then you can burst the report only on Country or Region.

**Procedure**

1. In IBM Cognos Framework Manager, include both the dimensional data source that is the basis for reporting and the relational burst table in the model.
   
   For more information about models, see the Framework Manager User Guide.

2. In IBM Cognos Report Studio, create a master detail report in which the master query drives the report and the detail query contains the burst information.

   You must group the master query on the data item on which you are bursting. This data item must have a corresponding data item in the relational burst table.

   Author the detail query against the relational burst table. The burst table must contain two columns: the data item corresponding to the data item used in the master report for bursting and the data item that contains the recipient information. The recipient can be an email address or an expression that results in a search path to an object in IBM Content Manager, such as an account, group, role, contact, or distribution list.

   For more information about master detail queries, see "Create a Master Detail Relationship" on page 292.

3. Ensure that the detail query, which must be evaluated by IBM Cognos Business Intelligence when the report is executed, is not visible:
   
   • Place a list that is based on the detail query in a conditional block with a box type of None.
   
   • Link the master and detail queries using the following expression:

     
     \[
     \text{[Master Burst Key]} = \text{[Detail Burst Key]}
     \]

**Results**

When you set the burst options for the report, the master query provides the data items for the burst key and the detail report provides the data items for the burst recipients.

---

**Example - Burst a Report**

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a report that lists product sales for each sales representative. The report is to be emailed to each sales representative, but they only need to see the data that pertains to them. You create a list report that you burst to each sales representative.

**Procedure**

1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click List and click OK.
4. From the Source tab, expand Sales and marketing (query) and Sales (query). Add the following data items:
   
   • Expand Employee by region and add Employee name.
   
   • Expand Products and add Product line, Product type, and Product.
5. Group the Employee name, Product line, and Product type columns.

6. Click the Employee name column, click the headers and footers button, and then click Create Header.
   Employee name appears as a header in the list. You no longer need to keep the data item as a list column.

7. In the list, click the Employee name column and click the delete button.

8. Click Revenue, and then click the summarize button and click Total.

9. Pause the pointer over the query explorer button and click Query1.
10. From the Source tab, drag Email from the Employee by region folder to the Data Items pane.
11. From the File menu, click Burst Options.
12. Select the Make report available for bursting check box.
13. Under Burst Groups, in the Query box, click Query1.
14. In the Label box, click Employee name.

15. Click the edit button.

16. In the Data items box, drag Employee name to the Groups folder and click OK.
17. Under Burst Recipient, in the Query box, click Query1.
18. In the Data Item box, click Email.
19. In the Type box, click Email addresses.
20. Save the report.
21. Locate the report in IBM Cognos Connection.

22. Under Actions, click Run with options.
23. Click advanced options on the right of the screen.
24. Under Time and mode, click Run in the background.
25. Select the Burst report check box.
26. Select the Send the report by email check box.
27. Run the report.

**Results**

When sales representatives access their email accounts, they will see reports with only the data that is meant for them.
Chapter 16. Format Data

Format data in a report to improve readability. For example, you can show all date values in the order year, month, and day. If you do not set Data Format properties here, data is formatted according to the properties set in the model. If the properties were not set in the model, data is formatted according to the International Components for Unicode (ICU) formats.

Special cells, such as overflow or underflow, errors, or not applicable values, appear with two dash (--) characters unless you override them.

You can also format data based on a condition or specify the format for a particular object.

If you create a calculation that uses mixed currency values, an asterisk character (*) appears as the unit of measure. To remove the asterisk character, change the format of the corresponding row or column.

Set the Default Data Formats

Set the default data properties for each type of data, including text, number, currency, percent, date, time, date/time, and time interval.

Procedure

1. From the Data menu, click Default Data Formats.
2. In the Format type box, click a format type.
   The properties that you can set for the selected format type appear in the Properties box.
3. If you clicked the Currency format type and require different currencies in the report, click the add button and select currency check boxes.
   For example, you may have one column with values are in euros and another column whose values are in US dollars.
4. Set the properties.
   If you added currencies in step 3, click each one and set the properties. If you did not add any currencies, any properties you set will apply to all currencies.
   For properties in which you type meta-characters that represent certain types of information, such as YYYY-MM-DD for dates, the meta-characters that are required depend on the authoring language specified for the report. For more information, see “Using Patterns to Format Data” on page 393.
   If you set a value for the Pattern property, all other formatting properties are ignored with the following exceptions:
   • Missing Value Characters
   • Zero Value Characters
   • Negative Pattern
   Some properties are language-sensitive and should be changed only with caution.
Results

The data formatting properties you set are applied to objects only in the current layout. If a data item contains values in multiple currencies but only a subset of those currencies have defined formats, the default format for the locale in which you are working is applied to values with no specified format.

Specify the Data Format for an Object

Specify the format for a particular object if you are not getting the results you want.

For example, you add a measure to a report and you want to see two decimals when you run the report. You set the number of decimals to two for the Number format type for the current layout. However, when you run the report, you see more than two decimals for the measure. To get the results you want, you must map the measure to the Number format type.

Data formats are not applied in delimited text (CSV) and XML report outputs.

Procedure

1. Click the object.
2. In the Properties pane, double-click the Data Format property.
3. Under Format type, click the format type to apply to the object.
4. To override any of the properties of the format type that were defined for the current layout, in the Properties box, click the property and specify its value.

Specifying the Number of Decimals in Numbers

When specifying the number of decimals, IBM Cognos Business Intelligence uses the IEEE 754 default rounding mode known as half even. With half even rounding, numbers are rounded toward the nearest truncated value, unless both truncated values are equidistant, in which case the value ending in an even digit is chosen, as follows:

- If the digit immediately after the precision to be shown is greater than 5, the number is rounded up.
- If the digit immediately after the precision to be shown is less than 5, the number is rounded down.
- If the digit immediately after the precision is a 5, the number is rounded down when the preceding digit is even and rounded up when the preceding digit is odd.

For example, the number 78.5 is rounded to 78, while the number 73.5 is rounded to 74.

In addition, if the maximum number of decimals is lower than the actual number of decimals in the number, the number is rounded to the maximum number of decimals.

Digit shaping in charts and maps

When working with bidirectional content, you cannot specify digit shaping at the chart or map level. You can specify digit shaping for the objects in charts and maps.
To understand how digit shaping is applied to charts and maps, you must know which chart and map objects are considered text and which objects are considered numeric values.

The following table describes the chart and map objects that are considered text.

Table 1. Chart and map objects that are considered text

<table>
<thead>
<tr>
<th>Container</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chart</td>
<td>chart title, subtitle, footer, notes, legend item, legend title, regression label, marker label, background image label, axis titles, discrete axis label (for example, category axis label, x-axis), baseline labels</td>
</tr>
<tr>
<td>Map</td>
<td>map title, subtitle, footer, legend title, notes, axis labels</td>
</tr>
</tbody>
</table>

The following table describes the chart and map objects that are considered numeric values.

Table 2. Chart and map objects that are considered numeric values

<table>
<thead>
<tr>
<th>Container</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chart</td>
<td>numeric axis label (for example, measure axis label, y-axis), numeric values that appear in the legend (you can select to show the first, last, maximum, minimum value for each item in legends), numeric values that appear in the chart</td>
</tr>
<tr>
<td>Map</td>
<td>legend items</td>
</tr>
</tbody>
</table>

The following list describes the different digit shaping options and how digit shaping is applied to charts and maps.

- The digit shape options for text objects, which include titles, footers, and labels, are Default, None, National, and Contextual. Digit shape options for numeric values are Default, None, and National. There is no Contextual option for numeric values.
- By default, digit shaping is not applied to numbers that appear on an object that is considered text. For example, if a chart shows year data on the x-axis and the content language specified in IBM Cognos Connection is Arabic (Egypt), no digit shaping is applied because the x-axis label is considered a string.

The following list describes the default digit shaping behavior for charts and maps when the content language implies non-European digits, such as Arabic (Egypt).

- Numeric values that have a specific format, such as decimal or currency, are displayed as Arabic-Indic.
- Except for matrix charts, numeric axis labels for charts and maps are displayed in European digits. This is due to the way that charts and maps are rendered. The engine used to render charts and maps does not perform any formatting for numeric axis labels. No locale information is passed, and consequently the labels are displayed as European digits. To display the labels as Arabic-Indic, select the digit shaping option National.
- Matrix charts are rendered using International Components for Unicode (ICU) to format numbers. Locale information is passed and the numbers are shaped accordingly. For example, when the content language is Arabic (Egypt), ICU displays numbers as Arabic-Indic.
- Numbers within text labels are displayed as is. That is, no digit shaping is applied.
The following list describes the default digit shaping behavior for charts and maps when the content language is Thai.

- Numeric values that have a specific format, such as decimal or currency, are displayed as European digits.
- Numbers within text labels are displayed as is. That is, no digit shaping is applied.

**Related concepts:**

“Support for bidirectional languages” on page 56

You can author reports that support bidirectional languages. You can specify base text direction, digit shaping, and container direction.

**Related information:**

“Digit Shaping” on page 970

Specifies the digit shaping option to apply. When National is selected, digit shapes are determined from the user’s content language. When Contextual is selected, digit shapes are determined from adjoining characters in the value. For example, if the most recent strongly directional character before the numerical character is left-to-right, the number is displayed as a European number. If the most recent strongly directional character before the numerical character is right-to-left, the number is displayed in Arabic-Indic format. If there are no strongly directional characters before the numerical character, the number is displayed according to the base text direction of the field. When None is selected, no shaping is performed, and the value as it appears in the data source is shown. The default depends on the type of value. When the value is a string, the default is None.

---

**Locale-sensitive Properties**

IBM Cognos Report Studio contains an extensive library of customized properties adapted to users from different regions who speak different languages. For example, if a modeler specifies that a given data item is a currency, only the proper currency symbol must be specified. When reports are created, Report Studio automatically adapts the format of the currency numbers to each user according to the content language specified in IBM Cognos Connection.

When modelers or report authors specify properties, these properties override user preferences and risk creating inconsistent formatting for users of other cultures. It is usually much easier and safer to let Report Studio take care of formatting. For example, for the date format type, different regions use different characters to represent the date separator. If you specify a date separator, you may confuse users in other regions.

The following data formatting properties are locale-sensitive:

- Currency Symbol Position
- Date Separator
- Date Ordering
- Calendar Type
- Time Separator
- Display AM/PM Symbols
- Clock
- Decimal Symbol
- Negative Sign Position
- Thousands Separator
- Group Size (digits)
Secondary Group Size (digits)
Missing Value Characters
Zero Value Characters
Pattern
Negative Pattern

Suppress Empty Cells

Sparse data may result in crosstabs showing empty cells. For example, a crosstab that matches employees with products results in many rows of empty values for the revenue measure if the employee does not sell those products.

You can suppress rows, columns, or rows and columns based on divide by zero, missing, and overflow values. Suppressing rows or columns without data gives you a more concise view of your report.

Calculations are performed before suppression is applied. If you have multiple crosstabs or charts, you must select one in order to access suppression options.

Within a list report, suppression applies only to rows and is based on non-grouped data items. If details of a group are null, then the header and footer are also suppressed.

Suppression can also be applied to charts, repeaters, and repeater tables.

Access to the suppression feature depends on the settings in your modeling component, IBM Cognos Transformer, IBM Cognos Framework Manager, and IBM Cognos Administration.

Procedure

1. From the Data menu, click Suppress and click Suppression Options.
2. Under Suppress, choose what sections to suppress.
3. Under Suppress the following, choose which values to suppress.

Related concepts:
“Limitations When Formatting Empty Cells in SAP BW Data Sources” on page 548

When working with SAP BW data sources, if the SAP BW server administrator configured custom formatting for empty cells on the SAP BW server, this custom format does not appear in IBM Cognos Business Intelligence reports. Ask your administrator to configure the formatting of empty cells in IBM Cognos BI.

Suppress Null Cells Using Filters

You can also use filters to suppress null cells in your reports. Using filters ensures that calculations take suppression into account. You may also obtain better report performance because the filtering is done at the data source.

If your report includes more than one measure or fact expression, it is best to base the filter on only one underlying fact or measure.

Relational-style Reports

In relational-style reports, you can use detail and summary filters.
**Dimensional-style Reports**

In dimensional-style reports, you can use the filter function.

For example, insert a set expression in your crosstab and use the expression filter (descendants ([Set]) is not null). If your crosstab includes three or more levels within the same dimension on an edge, use the expression filter (descendants (currentMember([Hierarchy]) is not null).

If your crosstab includes nested sets, filter the sets using a cascading approach to improve performance. For example, first filter the outermost (or highest nested level) set, and then filter the remaining sets proceeding inward.

**Example - Suppress Zeros in Rows and Columns in an Existing Report**

You are a report writer at The Sample Outdoors Company, which sells sporting equipment. You have a report that compares current year data to previous year data. You want to suppress zeros in the report to make the report more concise. You use the zero suppression tool to set the level of suppression.

**Procedure**

1. Open the GO Balance Sheet as at Dec. 31, 2012 report.
2. From the Data menu, click Suppress and click Suppression Options.
3. Under Suppress the following, click Rows and columns.
4. Under Suppress the following type of values, select the Zero values check box.
5. Run the report.
   - Rows and columns containing zeros are hidden.

![Great Outdoors Company balance sheet as at Dec 31, 2012 (with prior year comparative data)](Image)

**Specify what appears for data containers that contain no data**

You can specify what appears in a data container when no data is available from the database.

When no data is available, you can show one of the following options:

- An empty data container, such as a blank list.
• Alternate content, such as another data container or an image. You can insert any object from the Toolbox tab.

• Text, such as There is no data available for this month. If you show text, you can format it. By default, the text No Data Available appears.

You can specify what appears when no data is available for the following data containers: lists, crosstabs, charts, maps, repeaters, repeater tables, and tables of contents.

If your report includes multiple data containers, you can specify different no data contents for each container.

Tip: The No Data sample report in the GO Sales (query) package includes data containers that have no data. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Procedure
1. Select a data container.
2. In the Properties pane, click the select ancestor icon and click the data container type.
3. Click the ellipsis (...) button next to the No Data Contents property and select what should appear for data containers that contain no data:
   • To show an empty data container, click No Contents.
   • To show alternate content, click Content specified in the No data tab.
     Two tabs appear at the top of the data container and the No Data Contents tab is selected automatically.
     From the Toolbox tab, insert the objects to appear when there is no data available into the No Data Contents tab.
     • To show text, click Specified text and type the text that you want to appear.

Using Patterns to Format Data

You can format data so that it matches any pattern of text and numbers when default formats are not appropriate. For example, you can format dates to use full text including the era, or you can format them to only use numbers and show the last two digits of years to save space.

Using symbols and patterns can provide similar results as basic data formatting tasks. For example, you can set how many digits appear after the decimal point. You can achieve these types of results with a pattern, or you can set the No. of Decimal Places property. Patterns allow flexibility for more complex requirements.

Each supported content language code requires a specific set of symbols to be used in patterns. For each language code, there are two tables you will need; one for date and time symbols, and one for decimal symbols. The decimal symbols are the same for all locales, however, date and time symbols are grouped into six locale groups. Check the Date and Time Symbol section to see which locale group is used for your locale.

To define patterns, open the Data Format dialog box, and edit the Pattern property for each format type. Use the symbols that are defined in the language code tables, and follow these guidelines.
Pattern Guidelines

When you define a pattern, the number of symbols you use affects how the data will be shown. There are different rules for text, numbers, and values that can take the form of text or numbers.

Text

You can specify whether text is produced in full or abbreviated form.

<table>
<thead>
<tr>
<th>Number of symbols</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 or more</td>
<td>Full text form</td>
<td>EEEE produces Monday</td>
</tr>
<tr>
<td>Less than 4</td>
<td>Abbreviated form</td>
<td>EEE produces Mon</td>
</tr>
</tbody>
</table>

Numbers

The number of symbols you use in a pattern sets the minimum number of digits that are produced in a report. Numbers that have fewer digits than specified are zero-padded. For example, if you specify mm for minutes, and the database value is 6, the report will show 06.

Note: The year value is handled differently. If you specify two symbols for year, the last two digits of the year value is produced. For example, yyyy produces 2013, and yy produces 13.

Text and Numbers

For values that can produce text or numbers, such as months, you can specify whether text or numbers are produced, and whether words are abbreviated.

<table>
<thead>
<tr>
<th>Number of symbols</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 or more</td>
<td>Text</td>
<td>MMMM produces January</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MMM produces Jan</td>
</tr>
<tr>
<td>Less than 3</td>
<td>Numbers</td>
<td>MM produces 01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>M produces 1</td>
</tr>
</tbody>
</table>

Date and Time Symbols

Date and time symbols are divided into locales, each of which is detailed below.

Locale Group A

Locales: af-za, en, en-au, en-be, en-bw, en-ca, en-gb, en-hk, en-ie, en-in, en-mt, en-nz, en-ph, en-sg, en-us, en-vi, en-zh, fo-fo, gl-gw, id, id-id, is, is-is, it, it-ch, it-it, kk-kz, ms, ms-br, ms-my, nb-no, nl, nl-be, nl-nl, no, no-no, om-et, om-so, pl, pl-pl, pt, pt-br, pt-pt, so-dj, so-et, so-ke, so-so, sv, sv-fi, sv-se, sv-ke, sv-tz
<table>
<thead>
<tr>
<th>Meaning</th>
<th>Symbol</th>
<th>Presentation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Era</td>
<td>G</td>
<td>Text</td>
<td>AD</td>
</tr>
<tr>
<td>Year</td>
<td>y</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Year (of 'Week of Year')</td>
<td>Y</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Month in year</td>
<td>M</td>
<td>Text and number</td>
<td>July and 07</td>
</tr>
<tr>
<td>Week in year</td>
<td>w</td>
<td>Number</td>
<td>27</td>
</tr>
<tr>
<td>Week in month</td>
<td>W</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in year</td>
<td>D</td>
<td>Number</td>
<td>189</td>
</tr>
<tr>
<td>Day in month</td>
<td>d</td>
<td>Number</td>
<td>10</td>
</tr>
<tr>
<td>Day of week in month</td>
<td>F</td>
<td>Number</td>
<td>2 (2nd Wed in July)</td>
</tr>
<tr>
<td>Day of Week (1=first day)</td>
<td>e</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in week</td>
<td>E</td>
<td>Text</td>
<td>Tuesday</td>
</tr>
<tr>
<td>a.m. or p.m. marker</td>
<td>a</td>
<td>Text</td>
<td>pm</td>
</tr>
<tr>
<td>Hour in day (1 to 24)</td>
<td>k</td>
<td>Number</td>
<td>24</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (0 to 11)</td>
<td>K</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (1 to 12)</td>
<td>h</td>
<td>Number</td>
<td>12</td>
</tr>
<tr>
<td>Hour in day (0 to 23)</td>
<td>H</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Minute in hour</td>
<td>m</td>
<td>Number</td>
<td>30</td>
</tr>
<tr>
<td>Second in minute</td>
<td>s</td>
<td>Number</td>
<td>55</td>
</tr>
<tr>
<td>Millisecond</td>
<td>S</td>
<td>Number</td>
<td>978</td>
</tr>
<tr>
<td>Time zone</td>
<td>z</td>
<td>Text</td>
<td>Pacific Standard Time</td>
</tr>
<tr>
<td>Escape used in text</td>
<td>'</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Single quote</td>
<td>&quot;</td>
<td>n/a</td>
<td>'</td>
</tr>
</tbody>
</table>
### Locale Group B

Locales: be-by, bg-bg, el, el-gr, fi, fi-fi, hr, hr-hr, hu, hu-hu, ja, ja-jp, ko, ko-kr, ro, ro-ro, ru, ru-ua, ru-ru, sh-yu, sk, sk-sk, sl-si, sq-al, sr-sp, th, tr, tr-tr, uk-ua, zh, zh-cn, zh-hk, zh-mo, zh-sg, zh-tw

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Symbol</th>
<th>Presentation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Era</td>
<td>G</td>
<td>Text</td>
<td>AD</td>
</tr>
<tr>
<td>Year</td>
<td>a</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Year (of 'Week of Year')</td>
<td>A</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Month in year</td>
<td>n</td>
<td>Text and number</td>
<td>July and 07</td>
</tr>
<tr>
<td>Week in year</td>
<td>w</td>
<td>Number</td>
<td>27</td>
</tr>
<tr>
<td>Week in month</td>
<td>W</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in year</td>
<td>D</td>
<td>Number</td>
<td>189</td>
</tr>
<tr>
<td>Day in month</td>
<td>j</td>
<td>Number</td>
<td>10</td>
</tr>
<tr>
<td>Day of week in month</td>
<td>F</td>
<td>Number</td>
<td>2 (2nd Wed in July)</td>
</tr>
<tr>
<td>Day of Week (1=first day)</td>
<td>e</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in week</td>
<td>E</td>
<td>Text</td>
<td>Tuesday</td>
</tr>
<tr>
<td>a.m. or p.m. marker</td>
<td>x</td>
<td>Text</td>
<td>pm</td>
</tr>
<tr>
<td>Hour in day (1 to 24)</td>
<td>h</td>
<td>Number</td>
<td>24</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (0 to 11)</td>
<td>K</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (1 to 12)</td>
<td>k</td>
<td>Number</td>
<td>12</td>
</tr>
<tr>
<td>Hour in day (0 to 23)</td>
<td>H</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Minute in hour</td>
<td>m</td>
<td>Number</td>
<td>30</td>
</tr>
<tr>
<td>Second in minute</td>
<td>s</td>
<td>Number</td>
<td>55</td>
</tr>
<tr>
<td>Millisecond</td>
<td>S</td>
<td>Number</td>
<td>978</td>
</tr>
<tr>
<td>Time zone</td>
<td>z</td>
<td>Text</td>
<td>Pacific Standard Time</td>
</tr>
<tr>
<td>Escape used in text</td>
<td>'</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>
### Locale Group C


<table>
<thead>
<tr>
<th>Meaning</th>
<th>Symbol</th>
<th>Presentation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single quote</td>
<td>&quot;</td>
<td>n/a</td>
<td>'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Symbol</th>
<th>Presentation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Era</td>
<td>G</td>
<td>Text</td>
<td>AD</td>
</tr>
<tr>
<td>Year</td>
<td>u</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Year (of 'Week of Year')</td>
<td>U</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Month in year</td>
<td>M</td>
<td>Text and number</td>
<td>July and 07</td>
</tr>
<tr>
<td>Week in year</td>
<td>w</td>
<td>Number</td>
<td>27</td>
</tr>
<tr>
<td>Week in month</td>
<td>W</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in year</td>
<td>D</td>
<td>Number</td>
<td>189</td>
</tr>
<tr>
<td>Day in month</td>
<td>t</td>
<td>Number</td>
<td>10</td>
</tr>
<tr>
<td>Day of week in month</td>
<td>F</td>
<td>Number</td>
<td>2 (2nd Wed in July)</td>
</tr>
<tr>
<td>Day of Week (1=first day)</td>
<td>e</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in week</td>
<td>E</td>
<td>Text</td>
<td>Tuesday</td>
</tr>
<tr>
<td>a.m. or p.m. marker</td>
<td>a</td>
<td>Text</td>
<td>pm</td>
</tr>
<tr>
<td>Hour in day (1 to 24)</td>
<td>h</td>
<td>Number</td>
<td>24</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (0 to 11)</td>
<td>K</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (1 to 12)</td>
<td>k</td>
<td>Number</td>
<td>12</td>
</tr>
<tr>
<td>Hour in day (0 to 23)</td>
<td>H</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Minute in hour</td>
<td>m</td>
<td>Number</td>
<td>30</td>
</tr>
<tr>
<td>Second in minute</td>
<td>s</td>
<td>Number</td>
<td>55</td>
</tr>
</tbody>
</table>
### Locale Group D

Locales: de, de-at, de-be, de-ch, de-de, de-lu

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Symbol</th>
<th>Presentation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Era</td>
<td>G</td>
<td>Text</td>
<td>AD</td>
</tr>
<tr>
<td>Year</td>
<td>j</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Year (of 'Week of Year')</td>
<td>J</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Month in year</td>
<td>M</td>
<td>Text and number</td>
<td>July and 07</td>
</tr>
<tr>
<td>Week in year</td>
<td>w</td>
<td>Number</td>
<td>27</td>
</tr>
<tr>
<td>Week in month</td>
<td>W</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in year</td>
<td>D</td>
<td>Number</td>
<td>189</td>
</tr>
<tr>
<td>Day in month</td>
<td>t</td>
<td>Number</td>
<td>10</td>
</tr>
<tr>
<td>Day of week in month</td>
<td>F</td>
<td>Number</td>
<td>2 (2nd Wed in July)</td>
</tr>
<tr>
<td>Day of Week (1=first day)</td>
<td>e</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in week</td>
<td>E</td>
<td>Text</td>
<td>Tuesday</td>
</tr>
<tr>
<td>a.m. or p.m. marker</td>
<td>a</td>
<td>Text</td>
<td>pm</td>
</tr>
<tr>
<td>Hour in day (1 to 24)</td>
<td>h</td>
<td>Number</td>
<td>24</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (0 to 11)</td>
<td>K</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (1 to 12)</td>
<td>k</td>
<td>Number</td>
<td>12</td>
</tr>
<tr>
<td>Hour in day (0 to 23)</td>
<td>H</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Meaning</td>
<td>Symbol</td>
<td>Presentation</td>
<td>Example</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Minute in hour</td>
<td>m</td>
<td>Number</td>
<td>30</td>
</tr>
<tr>
<td>Second in minute</td>
<td>s</td>
<td>Number</td>
<td>55</td>
</tr>
<tr>
<td>Millisecond</td>
<td>S</td>
<td>Number</td>
<td>978</td>
</tr>
<tr>
<td>Time zone</td>
<td>z</td>
<td>Text</td>
<td>Pacific Standard Time</td>
</tr>
<tr>
<td>Escape used in text</td>
<td>'</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Single quote</td>
<td>&quot;</td>
<td>n/a</td>
<td>'</td>
</tr>
</tbody>
</table>

**Locale Group E**

Locales: fr, fr-be, fr-ca, fr-ch, fr-fr, fr-lu

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Symbol</th>
<th>Presentation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Era</td>
<td>G</td>
<td>Text</td>
<td>AD</td>
</tr>
<tr>
<td>Year</td>
<td>a</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Year (of 'Week of Year')</td>
<td>A</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Month in year</td>
<td>M</td>
<td>Text and number</td>
<td>July and 07</td>
</tr>
<tr>
<td>Week in year</td>
<td>w</td>
<td>Number</td>
<td>27</td>
</tr>
<tr>
<td>Week in month</td>
<td>W</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in year</td>
<td>D</td>
<td>Number</td>
<td>189</td>
</tr>
<tr>
<td>Day in month</td>
<td>j</td>
<td>Number</td>
<td>10</td>
</tr>
<tr>
<td>Day of week in month</td>
<td>F</td>
<td>Number</td>
<td>2 (2nd Wed in July)</td>
</tr>
<tr>
<td>Day of Week (1=first day)</td>
<td>e</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in week</td>
<td>E</td>
<td>Text</td>
<td>Tuesday</td>
</tr>
<tr>
<td>a.m. or p.m. marker</td>
<td>x</td>
<td>Text</td>
<td>pm</td>
</tr>
<tr>
<td>Hour in day (1 to 24)</td>
<td>h</td>
<td>Number</td>
<td>24</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (0 to 11)</td>
<td>K</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Meaning</td>
<td>Symbol</td>
<td>Presentation</td>
<td>Example</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Hour in a.m. or p.m.</td>
<td>k</td>
<td>Number</td>
<td>12</td>
</tr>
<tr>
<td>(1 to 12)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hour in day (0 to 23)</td>
<td>H</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Minute in hour</td>
<td>m</td>
<td>Number</td>
<td>30</td>
</tr>
<tr>
<td>Second in minute</td>
<td>s</td>
<td>Number</td>
<td>55</td>
</tr>
<tr>
<td>Millisecond</td>
<td>S</td>
<td>Number</td>
<td>978</td>
</tr>
<tr>
<td>Time zone</td>
<td>z</td>
<td>Text</td>
<td>Pacific Standard Time</td>
</tr>
<tr>
<td>Escape used in text</td>
<td>'</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Single quote</td>
<td>&quot;</td>
<td>n/a</td>
<td>&quot;</td>
</tr>
</tbody>
</table>

**Locale Group F**

Locales: ga-ie

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Symbol</th>
<th>Presentation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Era</td>
<td>R</td>
<td>Text</td>
<td>AD</td>
</tr>
<tr>
<td>Year</td>
<td>b</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Year (of 'Week of Year')</td>
<td>B</td>
<td>Number</td>
<td>2013</td>
</tr>
<tr>
<td>Month in year</td>
<td>M</td>
<td>Text and number</td>
<td>July and 07</td>
</tr>
<tr>
<td>Week in year</td>
<td>t</td>
<td>Number</td>
<td>27</td>
</tr>
<tr>
<td>Week in month</td>
<td>T</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in year</td>
<td>l</td>
<td>Number</td>
<td>189</td>
</tr>
<tr>
<td>Day in month</td>
<td>L</td>
<td>Number</td>
<td>10</td>
</tr>
<tr>
<td>Day of week in month</td>
<td>F</td>
<td>Number</td>
<td>2 (2nd Wed in July)</td>
</tr>
<tr>
<td>Day of Week (1=first day)</td>
<td>e</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Day in week</td>
<td>E</td>
<td>Text</td>
<td>Tuesday</td>
</tr>
<tr>
<td>a.m. or p.m. marker</td>
<td>a</td>
<td>Text</td>
<td>pm</td>
</tr>
<tr>
<td>Meaning</td>
<td>Symbol</td>
<td>Presentation</td>
<td>Example</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>Hour in day (1 to 24)</td>
<td>u</td>
<td>Number</td>
<td>24</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (0 to 11)</td>
<td>K</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Hour in a.m. or p.m. (1 to 12)</td>
<td>k</td>
<td>Number</td>
<td>12</td>
</tr>
<tr>
<td>Hour in day (0 to 23)</td>
<td>U</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>Minute in hour</td>
<td>n</td>
<td>Number</td>
<td>30</td>
</tr>
<tr>
<td>Second in minute</td>
<td>s</td>
<td>Number</td>
<td>55</td>
</tr>
<tr>
<td>Millisecond</td>
<td>S</td>
<td>Number</td>
<td>978</td>
</tr>
<tr>
<td>Time zone</td>
<td>c</td>
<td>Text</td>
<td>Pacific Standard Time</td>
</tr>
<tr>
<td>Escape used in text</td>
<td>’</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Single quote</td>
<td>&quot;</td>
<td>n/a</td>
<td>’</td>
</tr>
</tbody>
</table>

### Decimal Format Symbols

*All locales*

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A digit that is shown even if the value is zero.</td>
</tr>
<tr>
<td>#</td>
<td>A digit that is suppressed if the value is zero.</td>
</tr>
<tr>
<td>.</td>
<td>A placeholder for decimal separator.</td>
</tr>
<tr>
<td>,</td>
<td>A placeholder for thousands grouping separator.</td>
</tr>
<tr>
<td>E</td>
<td>Separates mantissa and exponent for exponential formats.</td>
</tr>
<tr>
<td>;</td>
<td>Separates formats for positive numbers and formats for negative numbers.</td>
</tr>
<tr>
<td>-</td>
<td>The default negative prefix.</td>
</tr>
<tr>
<td>%</td>
<td>Multiplied by 100, as percentage.</td>
</tr>
<tr>
<td>%00</td>
<td>Multiplied by 1000, as per mille.</td>
</tr>
<tr>
<td>.Currency Symbol</td>
<td>The currency symbol. If this symbol is present in a pattern, the monetary decimal separator is used instead of the decimal separator.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>€</td>
<td>The international currency sign. It will be replaced by an international currency symbol. If it is present in a pattern, the monetary decimal separator is used instead of the decimal separator.</td>
</tr>
<tr>
<td>X</td>
<td>Other characters that can be used in the prefix or suffix.</td>
</tr>
<tr>
<td>'</td>
<td>Used to quote special characters in a prefix or suffix.</td>
</tr>
<tr>
<td>/u221E</td>
<td>Infinity symbol.</td>
</tr>
<tr>
<td>/uFFFD</td>
<td>Not a Number symbol.</td>
</tr>
</tbody>
</table>
Chapter 17. Using Query Macros

A macro is a fragment of code that you can insert in the Select statement of a query or in an expression. For example, add a macro to insert a new data item containing the user’s name.

You can include references to session parameters, parameter maps, and parameter map entries in macros. Parameter values are set when you run the query. For example, you can use the language session parameter to show only the data that matches the language setting for the current user.

Macros can be used in these different ways:
- They can be inserted in the SQL.
  An example is `Select * from Country where Country.Name = #$myMap{$runLocale}#`
- They can supply an argument to a stored procedure query subject.
  If a value is not hard-coded for the argument, the stored procedure query subject can be used to return different data.
- They can be inserted in expressions, such as calculations and filters.
  An example is a filter `[gosales].[Sales staff].[Staff name] = #$UserLookUpMap{$UserId}#`
- They can be used as a parameter wizard.
  Parameters can reference other parameters. An example is `Map1, Key = en-us, Value = #$myMap{$UserId}#`

You can also add query macros to the IBM Cognos Framework Manager model. For more information, see the Framework Manager User Guide.

Support for query macros in IBM Cognos Report Studio includes the same capabilities as macros used in Framework Manager. However, Report Studio query macros do not extend to the layout. Therefore, when making changes to the query using macros, you must bear in mind the side-effects on the layout. For example, if a macro removes a column from the query that the layout refers to, a run-time error will occur.

Syntax

Use the following syntax to reference session parameter and parameter values.

<table>
<thead>
<tr>
<th>Object</th>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session key</td>
<td>$session_key</td>
<td>#$my_account#</td>
</tr>
<tr>
<td>Parameter map key</td>
<td>$map{&lt;key&gt;}</td>
<td>#$map_one{'abc'}#</td>
</tr>
<tr>
<td>Parameter map entry whose key is</td>
<td>$map{$session_key}</td>
<td>#$map_one{$my_account}#</td>
</tr>
<tr>
<td>defined by a session parameter</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can add the following elements to further define the macro expression.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Single quotation marks ' | Delineates a literal string that has a single quotation mark as part of the string. 

If the single quotation mark displays in a string, such as a query item, the string must be enclosed in a single quotation mark on both sides of the string and the single quotation mark must be doubled. For example, ab'c is written as 'ab''c' 

If the single quotation mark displays in a macro, the string must be enclosed in square brackets. For example, ab'c is written as [ab'c] 

If the single quotation mark displays in a prompt, there is no need to enclose the string. 

To escape a single quotation mark in an expression, use &apos; |
| Square brackets [ ] | Encloses model objects, such as a namespace or query subject and macro names that contain restricted characters, such as a number sign, hyphen, or space. |
| Curly brackets, also known as braces { } | Calls a function that is unknown to the parser, such as dateadd in DB2, and whose first argument is a keyword. 

Example: 

    dateadd ((month),2,<date expression>) |
| + operator | Concatenates two strings, such as 'abc' + 'xyz' |
| Single quote function (sq) | Surrounds the result of a string expression with single quotation marks. If the single quotation mark displays in a string, such as a query item, the string must be enclosed in a single quotation mark on both sides of the string and the single quotation mark must be doubled. You can use this function to build clauses to test against literal parameter-driven values. 

Here is an example: 

    #sq($my_sp)# 

If a session parameter (my_sp) has the value ab'cc, the result is 'ab"cc' |
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Double quote function (dq) | Surrounds the result of a string expression with double quotation marks. You can use this function to refer to table and column names with non-identifier characters, such as a blank space or a percent sign (%).  
Here is an example:  
#dq
('Column' + $runLocale)#  
If runLocale=en-us, the result is  
"Column en-us" |
| Square bracket function (sb) | Inserts a set of square brackets into the argument to build object references in a model query and model expressions, such as filters and calculations.  
Here is an example:  
#sb
('my item in ' + $runLocale)#  
If runLocale=en-us, the result is [my item in en-us] |

For information about functions, see Appendix F, “Using the expression editor,” on page 555.

Add a Query Macro

Add a query macro to allow run-time changes to be made to SQL queries.

About this task

When you reference a parameter, you must do the following:

- Use a number sign (#) at the beginning and end of each set of one or more parameters.  
Everything between the number signs is treated as a macro expression, which is processed at runtime. Framework Manager removes anything that is outside the number signs.
- Precede each parameter map entry with a dollar sign ($)  
- Use a name that starts with an alpha character (a..z, A..Z)

Do not insert macros between existing quotation marks or square brackets because IBM Cognos Report Studio does not execute anything within these elements.

Tip: For a list of supported macro functions, see Appendix F, “Using the expression editor,” on page 555.

Procedure

1. To add a macro to an expression, in the Expression Editor, click the macros tab ☰️.
2. Pause the pointer over the query explorer button and click Queries.

3. If the query to which you are adding a macro was built by using your own SQL, do the following:
   - Click the SQL object that is linked to the query.
   - In the Properties pane, double-click the SQL property.
   - In the SQL dialog box, type the macro.

4. If the query to which you are adding a macro was not built by using your own SQL, do the following:
   - Click the query.
   - In the Properties pane, double-click the Generated SQL/MDX property.
   - In the Generated SQL/MDX dialog box, click Convert.
   - In the SQL dialog box, type the macro.

---

### Creating Prompts Using Query Macros

You can create mandatory and optional prompts in reports using query macros.

Use the prompt macro functions prompt and promptmany to create single-value and multiple-value prompts. You can use prompt macro functions when working with a relational data source or a dimensionally-modeled relational (DMR) data source.

If you want to use a prompt macro in an expression such as a calculation, you must specify the data type when using an overloaded operator, such as a plus sign (+). You can use the plus sign to concatenate two items and to add two items.

If you want to define a filter on a dimension level and have the filter use the prompt or promptmany macro, you must provide the data type as memberuniquename and a default value.

Here is an example:

```plaintext
members( [MS_gosales].[New Dimension].[PRODUCTLINE].[PRODUCTLINE])
in { set { #promptmany('what', 'memberuniquename', '[MS_gosales].[PROD1].[PRODUCTLINE].[PRODUCTLINE] ->[all].[1]')# } )
```

Here is an example of a mandatory prompt:

```plaintext
select
   COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
   COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
   COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
   COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
gosales.gosales.dbo.COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY = #prompt('CountryName')#
```

When default values are specified in the syntax of macro prompts, you may see an error. Use prompt syntax such as where Country = ?Enter Country?.

The prompt and promptmany functions have the following mandatory and optional parameters. All argument values must be specified as strings.
Name

This mandatory parameter is the name of the prompt. Name can also refer to the name of a parameter on a user-created prompt page, in which case the user-created prompt page appears when the report is run instead of the default prompt page that the macro would generate.

Datatype

This optional parameter is the prompt value data type. The default value is string. Prompt values are validated. In the case of strings, the provided value is enclosed in single quotation marks and embedded single quotation marks are doubled.

Values include the following:
- boolean
- date
- datetime
- decimal
- double
- float
- int
- integer
- interval
- long
- memberuniquename
  Memberuniquename is not an actual data type. This value must be used when the data type of the parameter is member unique name (MUN).
- numeric
- real
- short
- string
- time
- timeinterval
- timestamp
- token
  Token is not an actual data type. It is a way to pass SQL. A token does not pass values.
- xsddate
- xsddatetim
- xsddatetime
- xsddecimal
- xsddouble
- xsdduration
- xsdfloat
- xsdint
- xsdlong
- xsdshort
- xsdstring
- xsdttime
**DefaultText**

This optional parameter is the text to be used by default. If a value is specified, the prompt is optional.

If you use a space and no values are provided in the Prompt Value dialog box, a Where clause is usually not generated.

If you use text and no values are provided in the Prompt Value dialog box, a Where clause is usually generated using the default value.

Ensure that the text you provide results in a valid SQL statement.

**Note:** If the data type is memberuniquename, a value for the DefaultText parameter must be provided. For example:

```
(#prompt('WhichLevel', 'memberuniquename',
'[goSalesAgain].[PRODUCT1].[PRODUCT].[PRODUCT(All)]->[all]')#)
```

**Text**

This optional parameter is text that precedes any user-provided values, such as 'and column1 = '.

**QueryItem**

This parameter is optional. The prompt engine can take advantage of the Prompt Info properties of the query item. Descriptive information can be shown, although the prompt value is a code.

**TextFollowing**

This optional parameter is the closing parenthesis that is used most often for the promptmany function. This parameter is also useful when the prompt is optional and is followed by hardcoded filters in the SQL statement.

**Examples - selecting country or region prompts**

When a report is run, you want your users to be prompted to choose the country or region for which they want to see data. The following code examples describe how you can use macros to create different prompts.

**Mandatory prompt with no data type specified**

Note the following:

- The Datatype argument is not specified. Therefore, it is a string, which is correct in this case.
- The DefaultText argument is not specified. Therefore, it is a mandatory prompt.

```
select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY = #prompt('CountryName')#
```
Mandatory prompt with the data type specified

Note the following:
• This prompt requires a valid integer value as response.
• The DefaultText argument is not specified. Therefore, it is a mandatory prompt.

```sql
select COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
       COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
       COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
       COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY_CODE >
    #prompt('Starting CountryCode',
    'integer',
    '',
    '',
    '[gosales].[COUNTRY_MULTILINGUAL].[COUNTRY_CODE]')#
```

Optional prompt and mandatory filter with the data type and default value specified

Note the following:
• This prompt allows the user to supply a valid integer response.
• The DefaultText argument is specified. Therefore, the user may omit entering a value, in which case the value 10 is used. This makes it an optional prompt, but not an optional filter.

Example 1:
```sql
select COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
       COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
       COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
       COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY_CODE >
    #prompt('Starting CountryCode',
    'integer',
    '10')#
```

Example 2:
```sql
[gosales].[COUNTRY].[COUNTRY] = #prompt('countryPrompt','string',''Canada'')#
Result 2:
[gosales].[COUNTRY].[COUNTRY] = 'Canada'
```

Note the following:
• The defaultText parameter must be specified such that it is literally valid in the context of the macro, because no formatting takes place on this value. See details below.
• The default string Canada in Example 2 is specified as a string using single quotes, in which the embedded single quotes are doubled up, thus 3 quotes. This results in the string being properly displayed within single quotes in the expression.
• As a general rule for the string datatype, the defaultText should always be specified as in the previous note, except in the context of a stored procedure parameter.
• For the defaultText of types date or datetime, a special format should be used in the context of SQL. Examples of these formats are 'DATE ''2012-12-25''' and 'DATETIME ''2012-12-25 12:00:00'''$. In all other contexts, you use the date/datetime without the keyword and escaped single quotes (e.g., '2012-12-25').

**Prompt that appends text to the value**

Note the following:

• The DefaultText argument is specified as a space character. In this case, the generated text is just the space character, which eliminates the WHERE clause from the query.

• The Text argument is specified, which is written into the generated SQL before the user-provided prompt value.

```sql
select COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
#prompt('Starting CountryCode',
  'integer',
  ' ',
  // <= this is a space
  'where COUNTRY_MULTILINGUAL.COUNTRY_CODE >'
)#
```

**Syntax substitution**

Note the following:

• The Datatype argument is set to token, which means that the user-provided value is entered without any checking on the provided value.

  Because checking is not performed on the value, the expression editor may indicate that the expression is not valid. When a valid user-provided value is supplied or if you provide a valid default value, expression editor will interpret the expression as valid.

• Token should be used only if there is a list of pick-values for the user.

• The DefaultText argument is specified. Therefore, this is an optional prompt and group by COUNTRY is used in the generated SQL.

```sql
Select COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
#prompt('Sort column',
  'token',
  'group by COUNTRY',
  'group by '
)#
```

**Examples - creating prompts that use parameter maps**

When a report is run, you want your users to select a language for the data in the report. The following examples describe several ways you can do this.
Prompt that uses session variables

Note the following:

- The name of the prompt is specified using a lookup in the parameter map named PromptLabels. The key value is the session variable $language.
- The WHERE clause is using a parameterized column.

```sql
select ORDER_METHOD.ORDER_METHOD_CODE as ORDER_METHOD_CODE,
       ORDER_METHOD.ORDER_METHOD_#$language# as ORDER_METHOD_EN
from gosales.gosales.dbo.ORDER_METHOD ORDER_METHOD
#prompt($PromptLabels{$language},
   ''
   ,''
   ,where ORDER_METHOD.ORDER_METHOD_.' + $language + '>
#)
```

A parameter map that nests prompts

Note the following:

- In the model, there is a parameter map DynPromptLabels with
  #$PromptLabels{$language}#  
- Part of the prompt information is run from a parameter map instead of being coded directly inside the SQL.
- The whole macro containing the prompt can be a value in a parameter map.

```sql
select ORDER_METHOD.ORDER_METHOD_CODE as ORDER_METHOD_CODE,
       ORDER_METHOD.ORDER_METHOD_#$language# as ORDER_METHOD_EN
from gosales.gosales.dbo.ORDER_METHOD ORDER_METHOD
#prompt($DynPromptLabels{'ex9'},
   ''
   ,''
   ,where ORDER_METHOD.ORDER_METHOD_.' + $language + '>
#)
```

Examples - creating a multiple value prompt

When a report is run, you want your users to select one or more values. The following examples describe several ways you can do this.

Prompt with a required minimum

Note the following:

- The user must enter at least a single value.
- This resembles the first example on prompting for a country or region
  "Mandatory prompt with no data type specified" on page 408.

```sql
select COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
       COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
       COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
       COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY IN (#promptmany('CountryName')#)
```
Prompt with a required minimum with the data type specified

Note the following:
- This resembles the second example on prompting for a country or region.

```
select COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
       COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
       COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
       COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY_CODE IN (
    #promptmany('Selected CountryCodes',
        'integer',
        '',
        '',
        '[gosales].[COUNTRY_MULTILINGUAL].[COUNTRY_CODE]')#
)
```

Optional prompt with the data type and default value specified

The In clause and both parentheses are part of the SQL statement.

```
select COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
       COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
       COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
       COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY_CODE IN (
    #promptmany('Selected CountryCodes',
        'integer',
        '10'
    )#
)
```

Prompt that adds text before the syntax

Note the following:
- This example uses the TextFollowing argument.

```
select COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
       COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
       COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
       COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
#promptmany('Selected CountryCodes',
    'integer',
    '',
    '/<==this
    is a space
    'where COUNTRY_MULTILINGUAL.COUNTRY_CODE IN
    (',
    ')
')#
```

Optional prompt that adds text before the syntax

```
Select COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
       COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
       COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
       COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from _
```
gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL,
gosales.gosales.dbo.COUNTRY XX
where COUNTRY_MULTILINGUAL.COUNTRY_CODE = XX.COUNTRY_CODE
#promptmany('Selected CountryCodes',
    'integer',
    '','
    ' and COUNTRY_MULTILINGUAL.COUNTRY_CODE IN (','
    ','
    ')
)="#
Chapter 18. Using Drill-through Access

Using drill-through access, you can move from one report to another within a session while maintaining your focus on the same piece of data. For example, you select a product in a sales report and move to an inventory report about that product.

Drill-through access helps you to build business intelligence applications that are bigger than a single report. Drill-through applications are a network of linked reports that users can navigate, retaining their context and focus, to explore and analyze information.

Drill-through access works by defining the target report or object and then using information from the source report to filter the target. IBM Cognos BI can match data from the source to metadata in the target report, or you can define parameters in the target for greater control.

Note: In the dimensional reporting style, you can also enable drilling up and drilling down. For more information about drilling up and drilling down, see “Create a Drill-up and Drill-down Report” on page 308.

What You Should Know

For a drill-through link to work, it is necessary to know

- the source
- the target
- how the data in the packages that contain these objects is related
  Depending on the underlying data, you may create a drill-through definition and have IBM Cognos BI match the data (dynamic drill-through) or create parameters in the target (parameterized drill-through).
- whether to run the target report or to open it
  The target of drill-through access is usually a saved report definition. The report can be created in IBM Cognos Report Studio, IBM Cognos Query Studio, IBM Cognos PowerPlay Studio, or IBM Cognos Analysis Studio. The target of drill-through access can also be a package that contains an IBM Cognos PowerCube, in which case a default view of the PowerCube is created.
- whether the users of the drill-through link in the source report have the appropriate permissions to view or run the target report
- if the target is being run, in what format to run it, and what filters to use on the target

If you do not want to run the target report on demand, you may link to a bookmark in the saved output instead of running the report. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

Sources and Targets

In IBM Cognos BI, there are many different examples of source and target. For example, you can drill through
• between reports created in different packages against different data source types, such as from an analysis against a package that contains a PowerCube to a detailed report against a package based on a relational data source
• from one existing report to another report using definitions created in Report Studio
• between IBM Cognos Viewer reports authored in Report Studio, Query Studio, and Analysis Studio
• to and from a package built on a PowerCube
• from IBM Cognos Series 7 to IBM Cognos BI reports
• from Metric Studio to other IBM Cognos BI reports by passing parameters using URLs

For more information, see the IBM Cognos Metric Studio User Guide.

Tip: In the GO Data Warehouse (analysis) package, the Positions to Fill sample report is a drill-through target report and the Recruitment Report sample report is a drill-through source report. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Understanding Drill-through Concepts

Before you set up drill-through access, you must understand the key concepts about drilling through. Knowing these concepts will help you to avoid errors so that report consumers drill through as efficiently as possible.

Drill-through Paths

You can create a drill-through path in a source report in IBM Cognos Report Studio, or using Drill-through Definitions in IBM Cognos Connection. A drill-through path is the definition of the path that is taken when moving from one report to another, including how the data values are passed between the reports.

Using Drill-through Definitions, you can create a drill-through path from any report in the source package to any target report in any other package in IBM Cognos Connection. This type of drill-through definition is stored in the source package. Users of any report in the package can use the drill-through definition to drill between any combination of IBM Cognos Analysis Studio, IBM Cognos Query Studio, IBM Cognos PowerPlay Studio, or IBM Cognos Viewer reports in any package.

For any target report that contains parameters, you should map the target parameters to the correct metadata in the drill-through path. This ensures that the values from the source report are passed to the correct parameter values, and that the target report is filtered correctly. If you do not map parameters, then the users may be prompted for values when the target report is run.

A report-based drill-through path refers to a path created and stored in a Report Studio source report. This type of drill-through path is also called authored drill through. The path is associated with a specific data column, chart, or cross tab in the source report, and is available only when users select that area of the report. If an authored drill-through definition is available, a hyperlink appears in the source report when it is run.

Report-based drill-through is limited to Report Studio source reports and any target reports. Use this type of drill-through access when you want to pass data
item values or parameter results from within a source report to the target report, pass the results of a report expression to a target report, or use URL link as a part of the drill-through definition.

**Selection Contexts**

The selection context represents the structure of the values selected by the user in the source.

In IBM Cognos Analysis Studio, this includes the context area. When a package drill-through definition is used, the selection context is used to give values for mapped parameters (parameterized drill-through) or also to map the appropriate data items and values.

Drill-through links can also be defined to open the target object at a bookmark. The content of this bookmark may also be specified by the selection context.

Drill-through access is possible between most combinations of the IBM Cognos Business Intelligence studios. Each studio is optimized for the goals and skills of the audience that uses it, and in some cases for the type of data source it is designed for. Therefore, you may need to consider how the various studios manage the selection context when you drill through between objects created in different studios, and how the data sources are conformed. During testing or debugging, you can see how source values are being mapped in different contexts using the drill-through assistant.

**Drilling Through to Different Report Formats**

The settings in the drill-through definition determine the format in which users see the report results.

For example, the users may see the reports in IBM Cognos Viewer as an HTML Web page, or the reports may open in IBM Cognos Query Studio, IBM Cognos PowerPlay Studio, or IBM Cognos Analysis Studio. If your users have PowerPlay Studio, they may also see the default view of a PowerCube.

Reports can be opened as HTML pages, or as PDF, XML, CSV, or Microsoft Excel spreadsheet software formats. When you define a drill-through path, you can choose the output format. This can be useful if the expected use of the target report is something other than online viewing. If the report will be printed, output it as PDF; if it will be exported to Excel for further processing, output it as Excel or CSV, and so on.

If you define a drill-through path to a report that is created in Analysis Studio, PowerPlay Studio, or Query Studio, consumers can open the report in its studio instead of in IBM Cognos Viewer. This can be useful if you expect a consumer to use the drill-through target report as the start of an analysis or query session to find more information.

For example, if an application contains a dashboard style report of high-level data, you can define a drill-through link to Analysis Studio to investigate items of interest. The Analysis Studio view can then be drilled through to a PDF report for printing.

**Note:** IBM Cognos Report Studio does not display data results.
Drilling Through Between Packages

You can set up drill-through access between packages.

The two packages can be based on different types of data source, but there are some limits. The following table shows the data source mappings that support drill-through access.

Table 3. Data source mappings that support drill-through access

<table>
<thead>
<tr>
<th>Source data source</th>
<th>Target data source</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLAP</td>
<td>OLAP</td>
</tr>
<tr>
<td></td>
<td>Note: OLAP to OLAP drill through is supported only if the data source type is the same, for example, SSAS to SSAS.</td>
</tr>
<tr>
<td>OLAP</td>
<td>Dimensionally modeled relational</td>
</tr>
<tr>
<td>OLAP</td>
<td>Relational data</td>
</tr>
<tr>
<td></td>
<td>Note: For more information, see &quot;Business Keys&quot; on page 421</td>
</tr>
<tr>
<td>Dimensionally modeled relational</td>
<td>Dimensionally modeled relational</td>
</tr>
<tr>
<td>Dimensionally modeled relational</td>
<td>Relational</td>
</tr>
<tr>
<td>Relational</td>
<td>Relational</td>
</tr>
</tbody>
</table>

Bookmark References

When you drill through, the values that you pass are usually, but not always, used to filter the report. IBM Cognos Business Intelligence supports bookmarks within saved PDF and HTML reports so that a user can scroll a report to view the relevant part based on a URL parameter.

For example, you have a large inventory report scheduled to run daily or weekly during off hours because of resource considerations. Your users may want to view this report as a target because it contains detailed information, but you want them to view the saved output rather than run this large report. Using this Action option and bookmark settings, users can drill through from another source location based on products to open the saved report to the page that shows the product they want to focus on.

When a bookmark in the source report is used in a drill-through definition, it provides the value for the URL parameter. When report consumers drill through using this definition, they see the relevant section of the target report.

Bookmark references are limited to previously run reports that are output as PDF or HTML and contain bookmark objects.

Members and Values

Dimensionally modeled data, whether stored in cubes or stored as dimensionally modeled relational (DMR) data, organizes data into dimensions. These dimensions contain hierarchies. The hierarchies contain levels. And the levels contain members.

An example of a dimension is Locations. A Locations dimension may contain two hierarchies: Locations by Organization Structure and Locations by Geography. Either of these hierarchies may contain levels like Country or Region and City.
Members are the instances in a level. For example, New York and London are members in the City level. A member may have multiple properties, such as Population, Latitude, and Longitude. Internally, a member is identified by a Member Unique Name (MUN). The method by which a MUN is derived depends on the cube vendor.

Relational data models are made up of data subjects, such as Employees, which are made up of data items, such as Name or Extension. These data items have values, such as Peter Smith.

In IBM Cognos Business Intelligence, the methods of drilling through available are:
- Dimensional (member) to Dimensional (member)
- Dimensional (member) to Relational (data item value)
- Relational (data item value) to Relational (data item value)

If the target parameter is a member, the source must be a member. The source and target should usually be from a conformed dimension. However, if the data supports it, you may also choose to define a mapping using different properties of the source metadata item.

If the target parameter is a value, the source can be either a value or a member. If the source is a dimensional member, you must ensure that the level or dimension is mapped to the target data item correctly in the drill-through definition. The business key from which the member is sourced should usually match the relational target value, which is most often the business key. However, if the data supports it, you may also choose to define a mapping from the caption of the source metadata item.

**Member Unique Names**

The member unique name (MUN) is a unique identifier for a member in IBM Cognos reports. It is stored in the report specification when the member is referenced in the report directly.

The MUN is used in drill-through between OLAP data sources. The member keys in the MUN for the different OLAP data sources must match.

The MUN is used to find the member in the data source, which is similar to how business keys are used to find records in a table. For example, when you create OLAP dimension Products, you use the Product Line database column as a label for the members in your Product Line level. However, you use the Product Line Code business key from the database table to ensure that all the Product lines are unique in that level. The source value that you used to create the members is used in combination with the data source name, hierarchy, and level information in the member unique name.

If the MUN changes, members that are directly referenced in expressions, filters, or reports are no longer found. Changes to the MUN may be related to other changes. For example, changes to the hierarchy and level structures may change the level unique name, and changes to the business key values may change the member key path. Other factors that can affect the MUN are application changes during the design stage or over time, IBM Cognos PowerCube category codes that are unpredictably unique, the production environment that has more members than the test environment, or removing the member from the data source.
To avoid potential problems, we recommend the following best practices when you build OLAP data sources:

- Use unique codes and keys within a dimension for the member keys.
- Define your OLAP and relational packages using unique conformed values for the source values (business keys) within similar dimensions or data values where drill-through between applications may be required.
- Ensure that the business keys and dimension metadata structure are the same in the production and test environments.
- Do not change the business keys in IBM Cognos Framework Manager in the production environment.
- Resolve the non-unique keys in a dimension in the data source before you build the cube.

  Ensure that there are no duplicate source values in all levels of a dimension before you build a PowerCube. We do not recommend using the tilde character (~) in the category codes.

  For more information, see the section about uniqueness in the IBM Cognos Series 7 Step-by-Step Transformer.

For information about PowerCubes migrated from IBM Cognos Series 7, see the IBM Cognos PowerPlay Migration and Administration Guide.

Conformed Dimensions

If you work with more than one dimensional data source, you may notice that some dimensions are structured the same, and some are not.

The reason that dimensions can be structured differently is that the data sources may serve different purposes.

For example, a Customer dimension appears in a Revenue data store, but not in an Inventory data store. However, the Products dimension and the Time dimension appear in both data stores.

Dimensions that appear in multiple data stores are conformed if their structure is identical for all of the following:
- hierarchy names
- level names
- level order
- internal keys

Drilling through is possible between different dimensional data stores only if the dimensions are conformed, and if the dimension data store is of the same vendor type, such as IBM Cognos PowerCube as the source and the target. For example, in two data stores for Revenue and Inventory that contain Products and Time dimensions, it is possible to define the Products and Time dimensions differently for each data store. However, for drill-through between the Products and Time dimensions to work, their structures must be identical in each data store.

If you are not sure whether your dimensions are conformed, then you should check with the data modeler to ensure that the drilling through will produce meaningful results.

IBM Cognos Business Intelligence does not support conformed dimensions generated by IBM Cognos Framework Manager for SAP BW data sources.
**Dimensionally modeled Relational Data Sources**

Ensure that each level contains a business key that has values that match your PowerCube or other DMR models. Also, you must also ensure that the **Root Business Key** property is set and uses the business key of the first level in the hierarchy. This helps to ensure that you have a conformed member unique name when attempting to drill through using members from this dimension.

**Related concepts:**

"Data Does Not Appear in a Target Report or the Wrong Data Appears" on page 496

If no data appears when you drill through to a target report or if the wrong data appears, the problem might be data source conformance. The business keys might be different or might be mismatched.

**Related tasks:**

"Unexpected or Empty Results When Drilling Through" on page 503

When you drill from a source report to a target report, there might be no data returned. This might be the correct result if there is no data that corresponds to the drill-through selections or if you do not have permission to view the data.

**Business Keys**

When drill-through access is defined from a member to a relational value, the business key of the member is passed by default.

This means that your relational target parameter must be set up using the data item with a matching value, which is most often the business key data item. You can also choose to pass the caption of the source metadata item.

For example, employees are usually uniquely identified by an employee number, not by their name, because their name is not necessarily unique. When you drill through from a dimensional member to a relational data item, the value provided is the business key. Therefore, the parameter in the target report must be defined to accept a business key value. The exact logic used to define the business key value supplied depends on the cube vendor. For IBM Cognos PowerCubes, the business key value is the **Source** property defined for the level in IBM Cognos Transformer. IBM Cognos Series 7 Transformer PowerCubes pass the source value if the drill-through flag was enabled before the cube was built. Otherwise, the category code is used.

In IBM Cognos Report Studio, you can determine what the member business key is using an expression such as `roleValue('_businessKey',[Camping Equipment])`. This expression is case-sensitive.

SSAS 2005 multipart business keys are not supported in drill-through operations.

**Tip:** When other users run your drill-through report, you may not want them to be prompted for a business key. In Report Studio, you can build a prompt page with a text that is familiar to the users, but filters on the business key. Your IBM Cognos Framework Manager modeler can also set the **Display Item Reference** option for the **Prompt Info** property to use the business key when the data item is used in a prompt.
Related concepts:
“Data Does Not Appear in a Target Report or the Wrong Data Appears” on page 496
If no data appears when you drill through to a target report or if the wrong data appears, the problem might be data source conformance. The business keys might be different or might be mismatched.

Related tasks:
“Unexpected or Empty Results When Drilling Through” on page 503
When you drill from a source report to a target report, there might be no data returned. This might be the correct result if there is no data that corresponds to the drill-through selections or if you do not have permission to view the data.

Scope

Scope is specific to drill-through definitions created using Drill-through Definitions in IBM Cognos Connection (package drill-through definitions). The scope you set defines when the target report is shown to the users, based on the items they have in the source report.

Usually, you define the scope of a drill-through path to match a parameter that it passes. For example, if a target report contains a list of employees, typically you want to display the report as an available drill-through choice only when a user is viewing employee names in a source report. If employee names are not in the source report and the scope was set on the employee name in the drill-through definition, the employee report does not appear on the list of available drill-through target reports in the Go To page. You can set the scope to a measure or to an item in the report.

In report-based drill-through access, where the drill-through path is associated with a specific report column, the column serves as the scope.

Mapped Parameters

Drill-through targets may contain existing parameters or you can add parameters to the target for greater control over the drill-through link.

You usually map all parameters in a drill-through target to items from the source.

When you map source items that are OLAP or DMR members to target parameters, you can select from a set of related member properties to satisfy the requirements of the target parameter. For a dimensional target, a dimensional source item uses the member unique name by default. For a relational target, a dimensional source item uses the business key by default.

For example, you could change the source member property that is used for a mapping to the member caption instead of the business key to match the parameter in a relational target. For a dimensional target, you could define a parameter that accepts a particular property (such as business key or parent unique name), then pass the appropriate source property to satisfy that target.

Note: If you define drill through between non-conformed dimensions, you should test carefully to ensure that the results behave as expected.

If you do not specify parameter mappings, then by default, you will be prompted for any parameters required in the target when you use the drill-through link. To customize this behavior, use the display prompt pages setting.
When the action is set to **Run using dynamic filtering**, then additional filtering is applied if names from the context in the source report match names of items in the target. Use this action as well when there are no parameters defined in the target.

If parameters are not mapped correctly, then you may receive an empty report, the wrong results, or an error message.

The source and target cannot contain identical parameter names when they are from different packages, even if the data structure is conformed. If the source and target are from the same package, there is no restriction.

If you have the necessary permissions, you can use the drill-through assistant to look at what source parameters are passed, and what target parameters are mapped for a given drill-through link.

You can change the dynamic drill-through filter behavior if you want drill-through to generate a filter using the Member Business Key instead of the default Member Caption. For more information, see Changing Drill-Through Filter Behavior in the *IBM Cognos Administration and Security Guide*.

### Drilling Through on Dates Between PowerCubes and Relational Packages

The usual method of drilling through from OLAP to relational packages requires that the target report parameter is set using the business key in the relational data, which does not work well for dates.

OLAP data sources typically view dates as members, such as Quarter 1 2012, while relational data sources view dates as ranges, such as 1/Jan/2012 to 31/March/2012.

A special feature exists for drilling through between PowerCubes and relational packages. Ensure that the target report parameter is set up using `in_range`. The parameter must be of type date-time, and not integer.

An example follows:

```plaintext
[gosales_goretailers].[Orders].[Order date] in_range ?Date?
```

Also ensure that the drill-through definition maps the parameter at the dimension level and that the PowerCube date level is not set to suppress blank categories. Enabling the option to suppress blank categories in the Transformer model before you build the cube may cause the drill-through on dates to be unsuccessful. This happens because there are missing values in the range.

### Set Up Drill-through Access in a Report

Set up drill-through access in a source report to link two reports containing related information.

You can then access related or more detailed information in one report (the target) by selecting one or more data item values from another report (the source). IBM Cognos passes values from the source to the target and uses the passed values to filter the target object. If the data in the source and target is conformed or if the data item names are the same, then the system may map the source values to filter the target (dynamic drill-through). If you require greater control, you may define parameters in the target object (parameterized drill-through).
For instructions about creating parameters in other kinds of targets, see the IBM Cognos Business Intelligence Administration and Security Guide.

When you define the drill-through path in IBM Cognos Report Studio, you can pass a value from a different data item that is hidden from the user for display purposes but still in the query. For example, users see the Product Name data item and can drill through on that item, but the drill-through definition passes the Product Number value for the product name the user chose. You can also define the target parameter to accept multiple values or a range of values from the data item in the drill-through source report or to accept a value from a parameter in the source report.

Drill-through definitions that have been authored inside a Report Studio report appear in the source report as blue underlined links. Users click the hyperlink to select the value they want passed to the target and to drill through to the target report. If a user selects multiple items within a single column then when the drill-through target report is run, the values from each selected row are passed to the target parameters. This occurs as an and condition.

You can also drill through within the same report by creating bookmarks. As well, you can create drill-through definitions in the source package instead of the Report Studio report. Users can use package drill-through definitions to navigate to a target report from an IBM Cognos Analysis Studio analysis, an IBM Cognos Query Studio report, an IBM Cognos PowerPlay Studio report or IBM Cognos PowerCube package, or a Report Studio report. For more information, see the Administration and Security Guide.

If you have the IBM Cognos Software Development Kit, you can use URLs to set up drill-through access to and from third-party sources.

You can also drill through from a map.

Related concepts:
“Example - Creating Drill-through Access in a Legacy Chart” on page 154

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a chart that shows the revenue for each product line and allows the reader to drill through from the revenue chart to view the product details for any item selected. You create a drill-through report to link two reports containing related information. You can then access related or more detailed information in one report by selecting a value in the chart. You create two reports: a target list report that contains the details for the item and a source report that contains the chart that shows the product line revenue.

Create a parameter in the target

Set up drill-through access in a source report to link two reports containing related information. You can then access related or more detailed information in one report (the target) by selecting one or more data item values from another report (the source).

If you are using an SAP BW data source for the target report and the target report contains a variable for a hierarchy node, values from the source report can be values only of the data item representing the leaf-level identifier of the hierarchy.
Before you begin

Before you begin, ensure that you have a report that will serve as the source report and another report that will serve as the target report.

Procedure

1. In IBM Cognos Report Studio, open the target report.
2. Create a parameter that will serve as the drill-through column or that will filter the report.
   For example, to drill through or filter Product line, create the following parameter:
   
   \[
   \text{[Product line]}=?\text{prodline\_p}?
   \]

   **Tip:** Use the operators `in` or `in_range` to enable the target report to accept multiple values or a range of values.

3. In the **Usage** box, specify what to do when a value for the target parameter is not passed as part of a drill through:
   - To specify that users must select a value, click **Required**.
     If a value for the target parameter is not passed, users are prompted to choose a value.
   - To specify that users do not need to select a value, click **Optional**.
     Users are not prompted to choose a value and so the value is unfiltered.
   - To specify not to use the parameter, click **Disabled**.
     The parameter is not used during the drill-through. It will also not be used in the report for any other purposes.

   **Tip:** If the parameter is needed in the report for other reasons, then you can also specify not to use it in the drill-through definition (Parameters table, Method, Do not use parameter).

Create a drill-through definition

Create a drill-through definition to link two reports containing related information (the source report and the target report).

Drill-through definitions that have been authored inside an IBM Cognos Report Studio report appear in the source report as blue underlined links. Users click the hyperlink to select the value they want passed to the target and to drill through to the target report. If a user selects multiple items within a single column then when the drill-through target report is run, the values from each selected row are passed to the target parameters. This occurs as an *and* condition.

Before you begin

Before you begin, ensure that you have a report that will serve as the source report and another report that will serve as the target report.

Procedure

1. Check the drill-through target:
   - Confirm that the drill-through users have access to the target.
   - If necessary, check what parameters exist in the target.
2. Open the source report.
3. Click the element in the report that will serve as the starting point for the drill-through link.
   You can select a data item that your report users are likely to choose to drill on for more detailed information, such as an Employee Name data item.

   **Tip:** If you are passing only parameter values to the target report, you do not have to drill on a data item. Instead, you can drill on any object in the report, such as the report title. Therefore, you can drill from outside the context of a query.

4. Click the drill-through definitions button, or, from the **Properties** pane, double-click the **Drill-Through Definitions** property.

5. Click the new drill-through definition button.
   A drill-through definition is created.

   **Tip:** To change the drill-through name, click the rename button, type the new name, and click **OK**.

6. On the **Target Report** tab, click the ellipsis (...) button next to the **Report** box and select the drill-through target report.

7. In the **Action** box, decide how users will view the target report when they click the drill-through link in the parent report:
   - To view the latest data in IBM Cognos Viewer, select **Run the report**.
   - To edit an IBM Cognos Query Studio, IBM Cognos Analysis Studio, or IBM Cognos PowerPlay Studio target, select **Edit the report**.
   - To have IBM Cognos Business Intelligence match values from the selection context with data in the target (dynamic filtering), select **Run the report using dynamic filtering**.
     **Note:** If you chose this option, any parameters in the target report are still used for the drill-through access.
   - To view the most recently saved output version of the target report from IBM Cognos Connection, select **View the most recent report**.
   - To use the default action specified for the report in IBM Cognos Connection, select **(Default)**.

8. If the target report contains parameters, then, under the **Parameters** box, click the edit button.
   Each required and optional parameter defined in the target report appears in the **Parameters** dialog box.

9. For each parameter, specify the **Method** of treatment:
   - To specify not to pass any value, click **(Default)**.
   - To specify not to use this parameter, click **Do not use parameter**.
     The target report will not be filtered by this parameter.
   - To pass values from a data item, click **Pass data item value**, click **Value**, and then click the data item.
     Values for the selected data item are passed to the target report.
   - To pass values from a source report parameter, click **Pass parameter value**, click **Value**, and then click the parameter.
     Values for the selected parameter are passed to the target report.
   - If you choose to pass parameters, and the source report is based on a dimensional package, then click an item from the drop-down list in the
Source metadata item properties column. You can pass the default property of the source metadata item (business key) or another property, such as the member caption.

10. Click OK.
11. If you chose to run the target report, in the Format box, click the output format for your report.

   Tip: Click (Default) to run the report using the default format specified for the report in IBM Cognos Connection.

12. To open the target report in a new window, select the Open in new window check box.

13. In the Display prompt pages box, choose how to display prompt pages:
   • To always display prompt pages when the drill-through action occurs regardless of whether values were specified for the required parameter values, click Always.
   • To use the prompt settings of the target report (specified by the Prompt for Values check box in IBM Cognos Connection, Report Properties, Report tab) to determine whether to show the prompt pages, click Based on the default prompt settings of the target report.
   • To not display prompt pages when the required parameter values are provided by the drill-through definition, click Only when required parameter values are missing. This is the default setting.

Results

The drill-through text appears as a blue hyperlink in text items in the non-chart areas of the report. Report consumers can also start the drill-through action by clicking the Go To button or by right-clicking the item and clicking Go To, Related links. If you have the necessary permissions, you can view which parameters were passed from the source and how they are mapped in the target object from the Go To page using the drill-through assistant.

For more information, see the Administration and Security Guide.

Defining independent drill-through definitions for crosstabs

By default, you can create only one drill-through definition for fact cells in a crosstab, regardless of how many facts there are. This topic describes how to define different drill-through definitions for different measures in crosstab fact cells.

For example, a crosstab has Quantity and Revenue as measures. For the Quantity cells, you want to define a drill-through definition to a target report that contains more information about quantities sold. For the Revenue cells, you want to define a drill-through definition to a different target report that shows details about revenue generated.

Procedure

1. Open the source report.
2. On the toolbar, click the unlock icon.
3. In the crosstab, click the white space inside one of the fact cells of the measure for which you want to define a drill-through definition.
Do not click the measure text. You want to select the crosstab intersection and not the text item. In the Properties pane title bar, ensure that Crosstab Intersection appears.

4. In the Properties pane, set the Define Contents property to Yes. The text item in the crosstab intersection disappears.

5. Click the Data Items tab.

6. From the crosstab query, drag the measure that disappeared in step 4 to the empty crosstab intersection.

7. Right-click the measure in the crosstab intersection and then click Drill-Through Definitions.

8. Create the drill-through definition.

Results

A drill-through definition is defined for the selected measure. Repeat the steps to define different drill-through definitions for other measures in the crosstab. When users run the report, clicking values from different measures takes them to a different target report.

Specify the Drill-through Text

You can specify the drill-through text that appears when users can drill through to more than one target.

For example, if users from different regions view the report, you can show text in a different language for each region.

Procedure

1. Right-click the drill-through object and click Drill-Through Definitions.

2. If more than one drill-through definition exists for the object, in the Drill-Through Definitions box, click a drill-through definition.

3. Click the Label tab.

4. To link the label to a condition, in the Condition box, do the following:
   - Click Variable and click an existing variable or create a new one.
   - Click Value and click one of the possible values for the variable.

5. In the Source type box, click the source type to use.

6. If the source type is Text, click the ellipsis (...) button that corresponds to the Text box and type text.

7. If the source type is Data Item Value or Data Item Label, click Data Item and click a data item.

8. If the source type is Report Expression, click the ellipsis (...) button that corresponds to the Report Expression box and define the expression.

9. If the label is linked to a condition, repeat steps 5 to 8 for the remaining possible values.

Results

When users run the source report and click a drill-through link, the Go to page appears. The drill-through text you specified appears for each target. If you did not specify the drill-through text for a target, the drill-through name is used.
Example - Drill Through to a Hidden Report from a Report Studio Report

You want to set up a drill-through link from an employee satisfaction report created in IBM Cognos Report Studio to a hidden list report about compensation, also created in Report Studio.

The source report (Employee Satisfaction 2012) is based on the package GO Data Warehouse (analysis) which is modeled on a DMR data source. The target report (Compensation (hidden)) is based on the package GO Data Warehouse (query). You set up this drill-through connection from within Report Studio (report-based, or authored drill through) because you do not want to make a report about compensation available for drill through from any source report in the package. The target report is already hidden in the portal, so that it is unlikely to be run by anyone who does not use the drill through link.

You must have the IBM Cognos Business Intelligence samples from the deployment zip file IBM_Cognos_DrillThroughSamples installed to follow this exercise, and you must have access to Report Studio.

The Compensation report is a hidden report. You may be able to set whether hidden reports are visible (My Preferences, General tab) and whether you can hide reports. This capability is set by your administrator.

Check the Target Report
Check the target report to make sure the drill-through will work.

Procedure
1. Open the target report:
   • In IBM Cognos Connection, go to Public Folders, Samples, Models, GO Data Warehouse (query), Report Studio Report Samples.
   • Locate the report Compensation (hidden) and open it in Report Studio.

   Tip: If you do not see the report, go to IBM Cognos Connection and confirm that you can view hidden reports (My Preferences, General tab).

2. In Report Studio, from the Data menu, click Filters and check what filter parameters are available.
   You want to filter from the source report on department, not time, so you will only use the pPosition parameter in the drill-through definition.

3. In the report body, select the list column body Position-department (level 3) and review the data item properties.
   Because the drill-through definition goes from DMR to relational, the data item values will need to match.


Create and Test the Drill-Through Definition
Create and test the drill-through definition to make sure it works.

Procedure
1. Open the source report:
   • In IBM Cognos Connection, go to Public Folders, Samples, Models, GO Data Warehouse (analysis), Report Studio Report Samples.
• Locate the Employee Satisfaction 2012 report and open it in Report Studio.

2. Save the Employee Satisfaction 2012 report with a new name, such as Employee Satisfaction 2012 New.
   This is to keep the original report and drill-through definition intact for comparison.

3. In the table Employee rankings and terminations by department, select the column Position-department (level 3).

4. In the properties pane, review the data item properties, to confirm that the data item names match values in the target report.

5. In the properties pane, under Data, double-click Drill-through definitions.

6. Select the definition DrilltoHiddenRep and delete it.
   **Note:** In the following steps, you recreate the drill-through definition. For comparison, use the original sample report.

7. In the Drill-through Definitions box, click the new drill-through definition button.

8. Click the rename button, and type a name for the drill-through definition.
   **Tip:** This is the name that consumers see when they select from a list of possible drill-through definitions in the final report.

9. In the Target Report tab, select the target report:
   • Under Report, click the ellipsis button (...).
   • Navigate to GO Data Warehouse (query), Report Studio Report Samples, and select the Compensation (hidden) report.
     **Tip:** If you do not see the report, go to IBM Cognos Connection and confirm that you can see hidden reports (My Preferences, General tab).

10. Under Action, select Run the report.

11. Under Parameters, click the edit button.
    A table of parameters available in the target report appears, showing the parameter **pPosition**.

12. Map the parameter from the Compensation (Hidden) report to the metadata in the Employee Satisfaction 2012 report:
    • In the Method column, select Pass data item value, because the target report is based on a relational data source.
    • In the Value column, select Position-department (level 3).

    **Tip:** In this report, you pass values from the column where the drill-through is defined. In other cases, you might pass a related parameter. For example, you could drill through on employee name, but pass the employee number.

13. Save the report.

14. Run the report, and click a department to test the drill-through definition.

**Results**

When you test the drill-through link, the Compensation (hidden) report appears, filtered by the department you selected. The report appears as a drill-through target whether or not it is hidden in IBM Cognos Connection.

If your administrator has given you the Drill Through Assistant capability, then you can see additional information you right-click on the link and select Go To see a list of drill-through targets. From the Go To page, you can see what source
values are passed, and what target parameters are mapped.

Try it yourself - create a report with drill-through access to itself

You can create a report that drills through to itself so that users can view detailed information in the same report.

In this topic, you learn how to create a report that shows revenue by each retailer. Users can access detailed order information for a particular retailer.

It should take 20-25 minutes to complete this topic, and your report will look like this.

![Figure 50. A report with drill-through access to itself](image)

**Procedure**

1. Create a new blank report that uses the GO Data Warehouse (query) package.
2. Add a table with two columns and one row.
3. Add a list object to each column in the table.
4. Open Sales and Marketing (query), then open Sales (query). Add these data items to the first list:
   - Retailer name in Retailers
   - Revenue in Sales fact
5. Set the sort order for the Retailer name column to Sort Ascending.
6. Add these data items to the second list:
   - Retailer name in Retailers
   - Order number in Sales order
   - Date in Time
   - Product in Products
   - Quantity in Sales fact
   - Revenue in Sales fact

   Tip: To simultaneously add all the data items to the list, Ctrl+click the items
   before dragging them to the list.
7. In the second list, select the Retailer name list column and in the Properties
   pane, set its Render property to No.
8. In the second list, create the following parameterized filter:
   [Retailer name]=?Selected retailer?
9. Save the report.
10. In the left list, right-click the Retailer name column (and not the column title)
    and click Drill-Through Definitions.
11. Create a new drill-through definition.
12. Specify the report as the target report.
13. Under Parameters, click the edit button.
14. Set the Selected retailer parameter to pass data item values using the Retailer
    name data item.
15. In the Toolbox tab, add a value prompt under the table.
    Because you are setting up drill-through access to the same report, you must
    add a prompt with a default value so that users are not prompted when they
    run the report.
16. In the Prompt Wizard, specify that the prompt is to use the existing parameter
    named Selected retailer, and click Finish.
17. Click the value prompt.
18. In the Properties pane, double-click the Default Selections property and add
    a simple selection named NoRetailer.
19. Set the following properties:
    - Required to No
    - Hide Adornments to Yes
    - Visible to No
20. Create a title for the report.
    a. In the Toolbox tab, add a block above the table.
    b. Click the block and, in the Properties pane, set the class to Report title area.
    c. Insert a text item above the table. Type the following text, adding a blank space at the end:
       Order Details:
    d. In the Toolbox tab, add the following layout calculation immediately following the text item:
if(ParamDisplayValue('Selected retailer')='NoRetailer')
then 'All Retailers' else (ParamDisplayValue('Selected retailer'))

When the report runs, the report title changes to reflect the retailer selected by the user. If no retailer is selected, 'All Retailers' appears.

e. Click the text item and the layout calculation and, in the Properties pane, set the class to Report title text.

21. Pause the pointer over the condition explorer button and click Variables.
Use conditional formatting to show a message above the second list when users have not selected a retailer from the first list.

22. Create a Boolean variable named HighLevel:
   \texttt{ParamDisplayValue('Selected retailer')}
   \texttt{<> 'NoRetailer'}

23. Pause the pointer over the page explorer button and click Page1.

24. In the Toolbox tab, add a block above the second list.

25. Insert a text item in the block with the following text:
   \texttt{Select a retailer in the Retailer list to view order details}

26. Set the font style for the text to bold.

27. Click the text item and, in the Properties pane, set the Style Variable property to the HighLevel variable.

28. Pause the pointer over the condition explorer and click the Yes value for the HighLevel variable.

29. Set the Visible property for the text item to No.

30. Triple-click the explorer bar to turn off conditional formatting.

31. Run the report to view what it will look like for your users.

**Results**

A list appears on the left that shows revenue for each retailer. When users click a retailer, order information for the selected retailer appears in the second list. To return to the original report, in IBM Cognos Viewer, click Previous Report.

For more information, see the following references:
- "Using a table to control where objects appear" on page 339
- Add Relational Data to a Report
- Create a Parameter to Produce a Prompt
- Set Up Drill-through Access in a Report
- Create a Prompt Directly in a Report Page
- Using Calculations
- Calculation Components
- Add a Variable
- Hide and Show Objects

**Drilling Through from IBM Cognos Series 7 to IBM Cognos BI**

You can set up drill-through access from IBM Cognos Series 7 to IBM Cognos Business Intelligence. Specifically, you can drill through to an IBM Cognos BI report from an IBM Cognos Series 7 PowerPlay Web report or IBM Cognos Series 7 Visualizer report. Drill through is supported for both PowerCubes and other cubes.

You must complete steps in both IBM Cognos Series 7 and IBM Cognos BI to enable drill through. Refer to the IBM Cognos Series 7 documentation for the
specific steps you must complete in the IBM Cognos Series 7 components. Cross references to the appropriate IBM Cognos Series 7 documents are included below.

**Setting Up Drill-through Access from IBM Cognos Series 7 Visualizer**

Setting up drill-through access from IBM Cognos Series 7 Visualizer to IBM Cognos Business Intelligence involves setting up the target report.

To set up drill-through access, you must do the following:

- Specify the IBM Cognos BI target and select the filters to add to the target report.
  
  You must configure drill through to IBM Cognos BI for individual IBM Cognos Visualizer reports. For more information, see the IBM Cognos Visualizer User Guide.

- Create and test the target report.

**Setting Up Drill-through Access from PowerPlay Web**

Setting up drill-through access from IBM Cognos Series 7 PowerPlay Web to IBM Cognos Business Intelligence involves setting up target reports.

To set up drill-through access, you must do the following:

- For PowerCubes, specify drill-through targets for IBM Cognos BI reports in the Transformer model.
  
  For more information, see the Transformer documentation.

- For other cubes, specify drill-through targets for IBM Cognos BI reports in IBM Cognos Series 7 PowerPlay Connect.
  
  For more information, see the PowerPlay OLAP Server Connection Guide.

- Configure drill-through access in PowerPlay Server Administration.
  
  In addition to enabling drill-through access to IBM Cognos BI, you must specify the location of the IBM Cognos BI server and the IBM Cognos BI folder that contains the target reports. For more information, see the PowerPlay Enterprise Server Guide.

- Select the filters to add to the target report.
  
  In PowerPlay Enterprise Server Administration, enable and use IBM Cognos BI Assistance to identify the filter expressions required in the target report. For more information, see the PowerPlay Enterprise Server Guide.

- Create and test the target report.

**Create and Test the Target for a Series 7 Report**

You can create and test an IBM Cognos Series 7 report target to ensure the drill-through works properly.

When you create the target report, ensure that the names of the parameters you add are identical to the parameter names listed in the Drill Through Assistant page in IBM Cognos Series 7. However, the metadata item that you use in the target report for that parameter name does not have to be the identical label. The data values between the target parameter and the source value shown in the drill assistant must match. You may also need to change the type of operator in the target parameter from what is recommended in the Drill Through Assistant. For example, if the assistant recommends an = operator but you want to pass a date range, you should change the parameter operator in the target to in_range.
Before you begin

The target report must be based on a published package that contains the metadata items that you want to filter on, or contains items that are mapped to those metadata items.

Procedure

2. Add the data items and other objects you want.
3. From the Data menu, click Filters.
4. In the Detail Filters tab, click the add button.
5. In the Expression Definition box, create the parameterized filter you want by typing the filter expression.
6. Click OK.
7. In the Usage box, click Optional.
   If you do not make the filter optional, a prompt page appears when you drill through to the report.
8. Repeat steps 4 to 7 for other parameterized filters you want to add.
9. Save the report.
   The report name must match what you specified as a target in the PowerCube, other cube, or IBM Cognos Series 7 Visualizer report.
10. Test the drill through in the IBM Cognos Series 7 PowerPlay report or IBM Cognos Visualizer report.
Chapter 19. Working with Multiple Pages

Divide a report into multiple pages and add navigation elements to make it easier to use.

Note: The pages that you see in the Page Explorer differ from physical, printed pages. For example, if you create page breaks in a long report, the Page Explorer could show only one page, but your report could include several physical pages when you print it.

Add a Page to a Report

Reports may contain pages with a variety of content. Pages may be added to create a multiple-page report. For example, you are creating a sales report and you want to include general information about the company. You decide to create an introduction page to show this information. When you run the report, page 1 of the report is the introduction page and the following pages contain data.

Tip: The TOC Report sample report in the GO Data Warehouse (query) package includes multiple pages. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

You can also create page sets to control the rendering of pages in a report.

Procedure

1. Pause the pointer over the page explorer button and click Report Pages.

   Tip: Click Prompt Pages to add a prompt page.

2. From the Toolbox tab, drag Page to the work area.
3. To change the order of pages, drag pages above or below other pages.
4. To associate a query to the page, in the Properties pane, set the Query property to the query.
   Associate the page to a query to add data-related objects directly to the page header, body, or footer, such as data items and filters.
5. Double-click the page to open it in page design view.
6. Add objects to the page.

Create a Page Break or Page Set

You can create simple page breaks, or you can create advanced page sets to associate pages with a query structure to force page breaks.

For example, a query contains the data item Product line, which is grouped. Creating a page set that is associated with this query adds page breaks for each product line. When working with dimensional data, you can use the Page layers area to show values on a separate page for each member.

When you are working with dimensional data, IBM Cognos Report Studio does not support page breaks or page sets that are based on custom groups.
Tip: The Global Bonus Report sample report in the GO Data Warehouse (analysis) package includes page sets. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Create simple page breaks
Simple page breaks use data items to define where your report starts a new page.

Procedure
1. Click the data item at which the report should break to a new page.
2. From the Structure menu, do one of the following:
   a. To create a page break without creating a master detail relationship, click Set Page Break.
      This option only works if the data item is in a list or repeater. If the data item is in a crosstab or chart, IBM Cognos Report Studio creates a page break using a master detail relationship. Setting page breaks without using master detail relationships can improve performance when running the report.
   b. To create a page break using a master detail relationship, click Set Page Break Using Master/Detail.
      Report Studio confirms the page break setting.
      Tip: To modify simple page break settings, pause the pointer over the page explorer button and click Report Pages, and then select and modify the page set.

Create advanced page sets
You can use advanced page sets to define sets of pages within your report, and then customize the grouping and sorting for each page set.

Procedure
1. Pause the pointer over the page explorer button and click Report Pages.
2. Create the report pages.
3. From the Toolbox tab, drag the Page Set object to the Report Pages pane.
4. In the Properties pane, set the Query property to the query to associate with the page set.
5. Organize the pages in the report by dragging report pages to page sets.
6. Insert the page containing details in the Detail Pages folder.
   Detail pages are the pages that repeat based on the items by which you group in the following step.
   Tip: You can insert multiple detail pages into the same page set and link them using a master detail relationship.
7. Define the grouping structure for the page set:
   a. Click the page set.
   b. In the Properties pane, double-click the Grouping & Sorting property.
   c. In the Data Items pane, drag the data item by which to group data items to the Groups folder in the Groups pane.
   d. To sort the data within each group, in the Data Items pane, drag the data item by which to sort to the Detail Sort List folder, and then click the sort
order button to specify the sort order. For more information about sorting data, see “Sorting Relational Data” on page 235 or “Sorting Dimensional Data” on page 281.

Note: Grouping an item for a page set is not the same as grouping a column in the layout. Grouping a column in the layout visually shows groups in a report. Grouping an item for a page set groups the item in the query. If you want to use an item that is already grouped in the layout, you must still perform step 7.

8. Repeat steps 3 to 7 to create other page sets.

Tip: You can nest page sets and join them by defining a master detail relationship. Create nested page sets to have pages occur within other pages. For example, you want pages containing product type information to occur within pages containing product line information.

Join Nested Page Sets

If you have nested page sets in your report, define a master detail relationship between them to see data in the nested page set that is related to the data in the parent page set.

For example, you have a page set that shows pages of product line information. The page set contains a nested page set that shows pages of product type information. For each product line page, you want to see the related product type pages, as shown below:

Product line_1
  • Product type_1
  • Product type_2
  • Product type_3

Product line_2
  • Product type_4
  • Product type_5
  • Product type_6

For more information about master detail relationships, see “Create a Master Detail Relationship” on page 292.

Procedure

1. Pause the pointer over the page explorer button and click Report Pages.
2. In the Report Pages pane, click the nested page set.
3. In the Properties pane, double-click the Master Detail Relationships property.
4. Click the New Link button.
5. In the Master Query box, click the data item that provides the primary information.
6. Link the master data item to the details by doing one of the following:
   • To link to another data item in the detail query, in the Detail Query box, click the data item that provides the detailed information.
To link to a parameter, in the Parameters box, click the parameter that provides the detailed information.

7. Repeat steps 4 to 6 to create other links.

Tip: To delete a link, select it and press the Delete key.

Example - Preparing a Product List Report

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to create a multiple-page report showing all products sold by the company. You are asked to create title and end pages and to have each product line appear on a new page preceded by a header page and followed by a footer page.

Procedure
1. Open IBM Cognos Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click List and click OK.
4. From the Source tab, expand Sales and Marketing (query), Sales (query), and Product, and double-click the following:
   - Product line
   - Product type
   - Product name
5. Expand Sales fact and double-click Product cost.
6. Group the Product line and Product type columns.
7. Pause the pointer over the page explorer button and click Report Pages.
8. From the Toolbox tab, drag Page Set to the work area and associate it to Query1.
9. From the Toolbox tab, drag Page to the work area four times to add four new pages.
10. For each page, click the page, and in the Properties pane, set the Name property as follows:
   - Title page
   - End page
   - Product Line Header
   - Product Line Footer
11. For each page you just created, add objects.
    For example, add a text item to each page to uniquely identify it.
12. Rename the Page1 page, which contains the list, to List.
13. Click the page set and, in the Properties pane, double-click the Grouping & Sorting property.
14. In the Data Items box, drag Product line to the Groups folder in the Groups box and click OK.
15. Organize the report pages into the following hierarchy by dragging them to the appropriate location.
Results

When you run the report, the following pages appear:

- Title page
- Product line header page
- A page for Product line_1
- Product line footer page
- Product line header
- A page for Product line_2
- Product line footer
- ...
- End page

Add a Bookmark

Add a bookmark so that users can quickly move from one part of a report to another. For example, a list report contains many rows of data. You add bookmarks so that users can move to specific rows.

You can also drill through to another report.

If you want to use page numbers to move from one part of a report to another, you can also create a table of contents.

Bookmarks work for reports produced in HTML format or PDF. In HTML format, they work best when viewing saved report outputs, as the entire report appears in a single HTML page. When reports are run interactively, more than one HTML page may be generated, and a bookmark works only if the target exists in the page currently being viewed.

If you run a saved report using a URL, and you specify a bookmark within the URL, you always go to the first page of the report. The bookmark defined in the URL is not honored. For more information about using URLs to run reports, see the Administration and Security Guide.

Tip: You can reduce the number of HTML pages generated when a report is run interactively by specifying a value for the Rows Per Page property for a data container in the report.
Tip: The Briefing Book sample report in the GO Sales (analysis) package includes bookmarks. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Procedure

1. From the Toolbox tab, drag a Bookmark object to the report.
2. Click the bookmark and, in the Properties pane, set the Source Type property to a source type.

<table>
<thead>
<tr>
<th>Source type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Creates a static bookmark using a text value.</td>
</tr>
<tr>
<td></td>
<td>For example, a list has sections and you want users to jump from each section to the top of the report.</td>
</tr>
<tr>
<td>Report Expression</td>
<td>Creates a dynamic bookmark whose values are derived from an expression that you define.</td>
</tr>
<tr>
<td>Data Item Value</td>
<td>Creates a dynamic bookmark that has data as possible values. This is useful for creating a context-based bookmark.</td>
</tr>
<tr>
<td></td>
<td>For example, a list has sections and you want users to jump from the top of the report to a specific section. Note: This source type appears only if the bookmark is inserted next to a data item.</td>
</tr>
<tr>
<td>Data Item Label</td>
<td>Creates a bookmark that has the label of a data item as its value. Use this source type to jump to the first occurrence of a data item label.</td>
</tr>
<tr>
<td></td>
<td>For example, a list is divided into sections using Product line. You want users to jump to the first product line section that appears in the list rather than to a specific section, such as Camping Equipment. Note: This source type appears only if the bookmark is inserted next to a data item.</td>
</tr>
<tr>
<td>Member Caption</td>
<td>In a crosstab, creates a dynamic bookmark that uses member captions as possible values.</td>
</tr>
<tr>
<td>Cell Value</td>
<td>In a crosstab, creates a dynamic bookmark that uses cell values as possible values.</td>
</tr>
</tbody>
</table>

3. Set the property that appears under the source type you chose to the bookmark value.
   For example, if the source type is Data Item Value, set the Data Item Value property to the data item.
4. Right-click the object that will jump to the bookmark and click Drill-Through Definitions.
   For example, right-click a data item, image, chart, or text item.

Tip: The object can exist in a different report, so you can jump from one report to another.
5. Click the new drill-through definition button.

6. Click the Bookmark tab.

7. Click Source type and click one of the source types described in step 3. Click the source type to use to produce the value needed to jump to the bookmark. For example, click Data Item Values if you want the value to come from a data item such as Product line.

8. If you clicked one of the following source types, specify the value to use to jump to the bookmark.
   - For Text, click the ellipsis (...) button next to Text and type a text value.
   - For Data Item Value or Data Item Label, in the Data item list, choose a data item.
   - For Report Expression, click the ellipsis (...) button next to Report expression and define the expression.

Create a Table of Contents

You can create a table of contents that appears in the rendered output of your report. A table of contents is useful for reports that include sectioned items, grouped items, or multiple pages in the layout. The report output indicates page numbers and allows for easy navigation.

You can include multiple tables of contents in your report, which are useful if your report contains grouped lists. You can also add section numbers in front of entries in the table of contents by modifying the heading level property.

Table of contents entries are logical markers placed anywhere in a report. For example, you can place entries at the top of a page or in a list group header to mark each grouped data value. Although table of contents entries are visible in IBM Cognos Report Studio, they cannot be seen when a report is run.

A table of contents works only for reports produced in PDF or non-interactive HTML format. In HTML format, they work best when viewing saved report outputs, as the entire report appears in a single HTML page. When reports are run interactively, more than one HTML page may be generated, and a table of contents works only if the target exists in the page currently being viewed.

Tip: You can reduce the number of HTML pages generated when a report is run interactively by specifying a value for the Rows Per Page property for a data container in the report.

If you want to quickly move from one part of a report to another without using page numbers, you can add bookmarks.

Tip: The Briefing Book sample report in the GO Sales (analysis) package includes a table of contents. For more information about The Sample Outdoors Company samples, see Appendix C, “Sample Reports and Packages,” on page 507.

Before you begin

You must first create a table of contents before adding entries in the report. All entries must be inserted after the table of contents in the report layout.
Procedure

1. From the Toolbox tab, drag the Table of Contents object to the new location, which can be anywhere in the report.
   
   A table of contents placeholder appears.

2. Drag the Table of Contents Entry object to the location of your first table of contents marker.
   
   Tip: You can also click Insert Table of Contents Entry from the Structure menu.
   
   The new entry appears in the table of contents.

3. Double-click the Double click to edit text box of the new marker.

4. In the Text box, type the text to appear in the table of contents and click OK.

5. To edit the heading level of a table of contents entry, click the entry and, in the Properties pane, set the Heading Level property to the level.
   
   The heading level is used to insert section numbers in front of entries in the table of contents using layout calculations.

6. When you finish creating the table of contents, run the report. By clicking the arrow to the right of the run report button, you can specify whether to run the report as HTML or PDF.

Example - Add a Table of Contents to a Report

You are a report author at The Sample Outdoors Company, which sells sporting equipment. You are requested to add a table of contents to an existing report so that users can more easily navigate your report.

Procedure

1. Open IBM Cognos Report Studio with the GO Data Warehouse (analysis) package.

2. Open the Budget vs. Actual sample report from the Report Studio Report Samples folder.

3. Create the report pages:
   
   - Pause the pointer over the page explorer button and click Report Pages.
   - Click Page1, and then, in the Properties pane, set the Name property to Budget vs. Actual Sales.
   - Click the Budget vs. Actual Sales page and, from the Edit menu, click Copy.
   - From the Edit menu, click Paste to paste the copy of the page in the Report Pages pane.
   - Select the new page, and in the Properties pane, set the Name property to Table of Contents.
   - In the Report Pages pane, drag the Table of Contents page to the top of the list.

4. Pause the pointer over the page explorer button and click Table of Contents.

5. Delete the crosstab object:
   
   - Click somewhere in the report page.
   - In the Properties pane, click the select ancestor button and click Crosstab.
Tip: You can also click the container selector (three orange dots) of the crosstab to select it.

- Click the delete button .

6. Select the Camping Equipment block object and click the delete button.

7. Double-click the report title, type Table of Contents, and click OK.

8. From the Toolbox tab , drag a Table of Contents object onto the page.

9. Pause the pointer over the page explorer button and click Budget vs. Actual Sales.

10. From the Toolbox tab, drag a Table of Contents Entry object to the left of each region.

11. Pause the pointer over the page explorer button and click Table of Contents.

12. Double-click the first entry in the table of contents, type Americas, and click OK.

13. Rename the other table of contents entries as Asia Pacific, Northern Europe, Central Europe, and Southern Europe.

14. Save the report.

15. Number the table of contents entries:
   - From the Toolbox tab, drag a Layout Calculation object to just before the word Americas.
     - In the Report Expression dialog box, on the Functions tab , expand the Report Functions folder and double-click the TOCHeadingCount expression.
     - At the end of the expression definition, type 1) and click OK.
     - Repeat the above three steps to add layout calculations before the other table of contents entries.
     - Ctrl+click only the five table of contents entries and not the five layout calculations.
     - In the Properties pane, double-click the Padding property and set the left padding to 10 px.

16. Change the color of table of contents entries:
   - Ctrl+click the five layout calculations, the five table of contents entries, and the five page number calculations.
   - In the Properties pane, double-click the Foreground Color property and set the foreground color to blue.

17. Add links to the table of contents from another page:
   - From the Toolbox tab, drag a Bookmark object to just before the title.
     - Select the bookmark and, in the Properties pane, double-click the Label property, type TOC, and click OK.
     - Pause the pointer over the page explorer button and click Budget vs. Actual Sales.
     - From the Toolbox tab, drag a Text Item object to the right of the crosstab object.
     - In the Text dialog box, type Return to the Table of Contents and click OK.
     - Right-click the text object and click Drill Through Definitions.

- Click the add button .
• On the Bookmark tab, in the Source Type list, click Text.
• Click the ellipsis (...) button, type TOC, and then click OK twice.

18. Save the report.

19. Click the arrow to the right of the run report button and click Run Report - PDF.

**Results**

On the first page, the table of contents appears. Clicking a region brings you to the corresponding page in the report. You can return to the table of contents by clicking Return to the Table of Contents at the end of the last page.

---

**Insert Page Numbers in a Report**

You can insert page numbers in a report and specify the number style to use.

You can select a predefined page numbering scheme or create a custom scheme. You can easily insert page numbers using the Page Number object.

You can also manually create an expression to insert page numbers by inserting a layout calculation and using the different page report functions in the expression editor.

**Procedure**

1. From the Toolbox tab, drag Page Number to the report.

   **Tip:** When you create a new report using one of the existing report layouts, Page Number is already inserted in the page footer.

2. Right-click the page number symbol and click Edit Number Style.

3. Choose the style to use.

   The first three choices apply only to vertical page numbers. The remaining choices specify how vertical and horizontal page values appear.

   **Note:** The 1 of 3 number style works only for reports produced in PDF or non-interactive HTML format. In HTML format, the 1 of 3 number style works when viewing saved report outputs, as the entire report appears in a single HTML page.

4. If you want to customize the choice that you made in the previous step, click the edit button, make your changes, and click OK.

   A custom number style is created. If you later choose a different number style, the custom style is removed from the list.
Tip: In the **Custom Number Style** dialog box, when you pause the pointer over a box, a tooltip describes how that box affects page numbers. For example, the **Separator Text** box contains the text, such as a hyphen, that separates page values for both vertical and horizontal pages.

## Control Page Breaks and Page Numbering

You can control page breaks and page numbering in a list, crosstab, table, or report page by choosing any of these options.

The options that are available depend on which object you have selected. All the options for all the objects are described in the following table.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep with header</td>
<td>Keeps all headers on the same page with the number of detail rows specified.</td>
</tr>
<tr>
<td>Keep with footer</td>
<td>Keeps all footers on the same page with the number of detail rows specified.</td>
</tr>
<tr>
<td>Keep with previous</td>
<td>Keeps the object with the specified number of preceding objects on the same page, if space permits.</td>
</tr>
<tr>
<td>Keep with next</td>
<td>Keeps the object with the specified number of subsequent objects on the same page, if space permits.</td>
</tr>
<tr>
<td>Reset page count</td>
<td>Resets the page count after a page break to the value specified.</td>
</tr>
<tr>
<td>Reset page number</td>
<td>Resets the page number after a page break to the value specified.</td>
</tr>
<tr>
<td>Repeat every page</td>
<td>If the report renders multiple pages, this object is repeated on every page.</td>
</tr>
<tr>
<td>Allow contents to break across pages</td>
<td>Allows contents to break across pages. In lists and crosstabs, controls whether a cell is broken across pages, which is useful when there is a lot of text.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Allow row contents to break across pages</td>
<td>In tables, allows the contents of a row to break across pages. For example, if a row contains four lines of text, the first two lines from the row appear on the first page, and the last two lines appear on the next page.</td>
</tr>
<tr>
<td>Repeat table rows on page break</td>
<td>In tables, if a row breaks across pages, repeats the rows that were previously rendered on each page. By default, table rows are repeated. Note: This option applies to saved reports only. In interactive HTML reports, table rows are always repeated even if this option is not selected.</td>
</tr>
</tbody>
</table>

You can also specify the style to use for page numbers.

**Procedure**

1. Click an object.
2. In the Properties pane, double-click the Pagination property.
3. Specify the page break and numbering options.

---

**Create Page Layers**

When working with dimensional data, you can create page layers in a report to show values for each member on a separate page. For example, your report contains payroll information for the entire company. You want to view values for each department on a separate page.

After you create page layers, a caption appears in the header to indicate the contents of each page. You can navigate between the different pages using links below the report.

Creating page layers is similar to filtering using context. However, with context filters, values are filtered according to the member you add to the Context filter area. With page layers, the report is split into a separate page for each child of the member you add to the Page layers area.

To create more complex page layers in your reports, such as a report book with title and end pages, create page sets.

**Procedure**

1. In the source tree, select or search for one or more items on which to filter.
2. Drag the item into the Page layers section of the overview area.
   
   The crosstab shows the results for the children of the selected item on separate pages, and a list appears under Page layers.

   **Tip:** To navigate between pages, click Page down and Page up below the report.

3. To replace the page breaks with items from the same dimension, select an item from the list. To delete the page breaks, from the list, click Delete. To delete all the page breaks, right-click the Page layers area and click Delete All.
Chapter 20. Creating Report Templates

A report template is a pattern you use to build reports. Create your own report templates when you frequently produce the same type of report.

Tip: You can add your own report templates to the New dialog box that appears when you open IBM Cognos Report Studio. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

Convert a Report to a Template

Convert a new or existing report to a template so it can be reused. You can use sample reports provided with IBM Cognos Report Studio.

Procedure

1. Create a new report or open an existing report.
2. Add objects to the work area.
3. From the File menu, click Convert To Template.
   Any query-related information in the original report, such as data items, calculations, and filters, is removed from the template.
4. From the File menu, click Save As to save the template as a new file and keep the original report intact.

Create a New Template

Create a new template to provide report authors with a layout that they can use to create reports or they can apply to existing reports. IBM Cognos Query Studio and IBM Cognos Analysis Studio users can also use these templates. You can also use the template in Query Studio to define a layout for prompt pages.

When you create a new template, the Source tab is empty; you can add only report objects.

Procedure

1. From the File menu, click New.
2. Click Report Template and click OK.
3. Pause the pointer over the page explorer button and click the report page or prompt page to format.
   Tip: To create a new report page or prompt page, click the Report Pages or Prompt Pages folder and drag the page to the Report Pages or Prompt Pages pane.
4. From the Toolbox tab, add the objects to the work area.
   Note: If you add objects that are not supported by Query Studio, the objects will be ignored when you apply the template.
5. Save the template.
Chapter 21. Managing Existing Reports

After you have created a report, you can make changes or enhancements, such as setting up the report for bursting.

Before you modify an existing report, ensure that you have the proper security permissions. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

Copy a Report to the Clipboard

You can copy a report specification to the clipboard so you can open it from the clipboard later.

This process is different for the Microsoft Internet Explorer and Mozilla Firefox Web browsers because the clipboard works differently in each Web browser.

Copy a Report to the Clipboard in Internet Explorer

This process is different for the Microsoft Internet Explorer and Mozilla Firefox Web browsers because the clipboard works differently in each Web browser.

Procedure

From the Tools menu, click Copy To Clipboard.

Copy a Report to the Clipboard in Firefox

This process is different for the Microsoft Internet Explorer and Mozilla Firefox Web browsers because the clipboard works differently in each Web browser.

Procedure

1. From the Tools menu, click Copy To Clipboard.
2. In the Copy Report to Clipboard dialog box, copy the entire report specification.
3. Open a text editor and paste the report specification.
4. Copy all the text from the text editor report specification.
   Now the text is saved on your computer's clipboard.

Open a Report from the Clipboard

You can open a report specification that was previously copied to the clipboard.

To copy a report to the clipboard, see "Copy a Report to the Clipboard." This is useful for importing an XML report specification from outside the IBM Cognos Business Intelligence environment.

Although IBM Cognos Report Studio attempts to validate the report specification, it is your responsibility to ensure that it is correct. For more information, see the IBM Cognos Software Developer Kit Developer Guide.
**Procedure**

From the **Tools** menu, click **Open Report from Clipboard**.

---

**Open a File from Another Studio**

You can open reports, templates, or analyses that were created in IBM Cognos Query Studio or IBM Cognos Analysis Studio in IBM Cognos Report Studio. All the capabilities of Report Studio are available to you so you can change formatting, layout, calculations, and queries.

You can also open a report, template, or analysis in Report Studio from IBM Cognos Connection with the **Open with Report Studio** action.

**Note:** If you make and save changes to a Query Studio report or an Analysis Studio analysis in Report Studio, the report or analysis can no longer be opened in Query Studio and Analysis Studio.

**Procedure**

1. From the **File** menu, click **Open**.
2. Click the report, template, or analysis.
3. Click **Open**.

---

**Analysis Studio Query Specification**

IBM Cognos Analysis Studio defines each group of rows and columns as a set. When you import an analysis into IBM Cognos Report Studio, the report will have one query that processes all the sets found on the crosstab. Each set is defined by data items that segment and summarize the base set definition. To maintain reports converted from Cognos Analysis Studio, you must understand what each of these items represents and how they relate to each other.

Before you modify any of the data items in the Cognos Analysis Studio set definitions, we recommend that you fully understand each data item and its dependencies. Modifying the data items may cause unpredictable results and may slow the performance of your report.

The data items for a set specify the following:
- the set definition
- set segments, including which members were excluded and hidden individually
- filter rules for defining which members are to be retrieved
- calculations for Subtotal (N items), More & Hidden, Subtotal (included), Subtotal (excluded), and the total

For more information, see the *IBM Cognos Analysis Studio User Guide*.

- which subtotals should appear
- any user-defined calculations
The default measure identifies which measure is the default for the crosstab or chart. If no default measure is specified, the default measure may be empty.

**Set Definitions**

The following table shows the definitions and dependencies for the data items in the set definition.

<table>
<thead>
<tr>
<th>Data item</th>
<th>Definition</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;set name&gt; (base)</td>
<td>Defines the set of members to be used for sorting, filtering, and summary operations. This data item provides a generic reference for all other data items and may reference one of the other set definitions.</td>
<td>None</td>
</tr>
<tr>
<td>&lt;set name&gt; (level)</td>
<td>Identifies the level for a level-based set.</td>
<td>None</td>
</tr>
<tr>
<td>&lt;set name&gt; (list)</td>
<td>Defines the list of members in a selection-based set.</td>
<td>None</td>
</tr>
<tr>
<td>&lt;set name&gt; (depth N)</td>
<td>Defines the set of members at N, number of levels down.</td>
<td>None</td>
</tr>
<tr>
<td>&lt;set name&gt; (named set)</td>
<td>References a predefined set.</td>
<td>None</td>
</tr>
</tbody>
</table>

**Set Segments Definitions**

The following table shows the definitions and dependencies for the data items in the set segment definition.
Data item | Definition | Dependencies
---|---|---
<set definition> (hidden list) | Lists the members that are manually hidden using the Hide command in Cognos Analysis Studio. This set appears when you try to unhide a member. | <set definition>

<set definition> (included set) | Lists the set of members after filters are applied and hidden items are excluded, but before the More limit is applied. Sort or order operations, if any, are defined in this data item. | <set definition> (hidden list)

<set definition> (visible items set) | Limits the set to show the number of members according to the More limit with a small tolerance. The tolerance allows showing the last two members if they are all that remain in the More portion of the set. The choice of visible members can be based on a sort order. | <set definition> (included set)

<set definition> (excluded list) | Identifies members that were manually excluded from the analysis. If no members were manually excluded, the expression defines an emptySet. This set appears when you try to remove a member from the Excluded items list in the Properties pane. | <set definition>

### Filters

The following table shows the definitions and dependencies for the data items in the filter.

<table>
<thead>
<tr>
<th>Data item</th>
<th>Definition</th>
<th>Dependencies</th>
</tr>
</thead>
</table>
<set definition> (filter rules) | Specifies the user-defined filter rules to reduce the set definition using operators such as greater than or less than based on measures, calculations, or attributes. | <set definition> |
### Data item Definition Dependencies

<table>
<thead>
<tr>
<th>Data item</th>
<th>Definition</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;set definition&gt;</code> (excluded filters)</td>
<td>Removes members that were manually excluded from the results after applying the user-defined rules.</td>
<td><code>&lt;set definition&gt;</code> (excluded list), <code>&lt;set definition&gt;</code> (filter rules)</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (filter top bottom)</td>
<td>Focuses on the members based on top / bottom / first n where n can be a count or a percentile. For more information about Top/Bottom filters, see the IBM Cognos Analysis Studio User Guide.</td>
<td><code>&lt;set definition&gt;</code> (excluded filters) and totals for sets on the opposite axis</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (filter rules)</td>
<td>Specifies the user-defined filter rules to reduce the set definition using operators such as greater than or less than based on measures, calculations, or attributes.</td>
<td><code>&lt;set definition&gt;</code></td>
</tr>
</tbody>
</table>

### Subtotals and Related Conditions

The following table shows the definitions and dependencies for the data items in the subtotals and related conditions.

<table>
<thead>
<tr>
<th>Data item</th>
<th>Definition</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;set definition&gt;</code> (subtotal)</td>
<td>Aggregates the visible items in the set.</td>
<td><code>&lt;set definition&gt;</code> (visible items set)</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (subtotal display)</td>
<td>Shows the subtotal if the number of included items is greater than the number of visible items.</td>
<td><code>&lt;set definition&gt;</code> (visible items set), <code>&lt;set definition&gt;</code> (included set), <code>&lt;set definition&gt;</code> (subtotal)</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (more and hidden subtotal)</td>
<td>Calculates the More &amp; hidden subtotal by subtracting the aggregation of the included members from the aggregation of those members that were manually hidden or hidden by exceeding the maximum display limit. Not available for selection-based sets.</td>
<td><code>&lt;set definition&gt;</code> (included set), <code>&lt;set definition&gt;</code> (visible items set), <code>&lt;set definition&gt;</code></td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (more and hidden subtotal as set)</td>
<td>Converts the More &amp; hidden subtotal member to a set for use in set operations.</td>
<td><code>&lt;set definition&gt;</code> (more and hidden subtotal)</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (more and hidden subtotal as member)</td>
<td>Generically references the More &amp; hidden subtotal.</td>
<td><code>&lt;set definition&gt;</code> (more and hidden subtotal)</td>
</tr>
<tr>
<td>Data item</td>
<td>Definition</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (more and hidden subtotal display)</td>
<td>Shows the More &amp; hidden subtotal if the number of items hidden or clipped is greater than zero.</td>
<td><code>&lt;set definition&gt;</code> (more and hidden subtotal)</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (included subtotal)</td>
<td>Calculates the Subtotal (included). For selection-based sets, this data item references the summary data item for the set.</td>
<td><code>&lt;set definition&gt;</code> (included set)</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (included subtotal display)</td>
<td>Shows the Subtotal (included) if any members passed the filter criteria.</td>
<td><code>&lt;set definition&gt;</code> (included set)</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (excluded subtotal)</td>
<td>Calculates the value for Subtotal (excluded). This data item is not available for selection-based sets.</td>
<td><code>&lt;set definition&gt;</code> (total), <code>&lt;set definition&gt;</code> (included subtotal), <code>&lt;set definition&gt;</code> (excluded subtotal)</td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (excluded subtotal display)</td>
<td>Shows Subtotal (excluded) if the number of members in the filtered set is less than the base set. This means that the filter rule is filtering out something.</td>
<td><code>&lt;set definition&gt;</code> (excluded subtotal), <code>&lt;set definition&gt;</code></td>
</tr>
<tr>
<td><code>&lt;set definition&gt;</code> (total)</td>
<td>Identifies the total for the set. This value is calculated directly from the data store. The expression used to calculate the summary depends on the set type: • Details-based set: The selected member. • Level-based and depth-based sets: The root member for the hierarchy. • Selection-based set: The aggregate, typically the sum, of the selected members.</td>
<td><code>&lt;set definition&gt;</code></td>
</tr>
</tbody>
</table>

### Managing Changes in the Package

If changes were made to the package that was used to create a report, the report must be updated.

When you open a report, IBM Cognos Report Studio automatically checks to see if the package has changed. If it has, a message appears indicating that the report will be updated to the latest version of the package. However, you may need to make additional changes to the report if
Update a Name Reference

If the namespace name or query subject and data item names in the package have changed, you must update reports that were created with the package to reflect the change.

The names of data items in a report are a concatenation of the namespace name, the query subject name, and the data item name. For example, if you add Order number from the GO Data Warehouse (query) sample package to a filter expression, you will see [Sales (query)].[Sales order].[Order number] in the expression. Similarly, package filter names are a concatenation of the namespace name and the filter name.

Procedure
1. To update a data item name reference:
   - Click the data item.
   - In the Properties pane, double-click the Expression property.
   - In the Expression Definition box, update the data item name reference.
2. To update a package filter name reference:
   - From the Data menu, click Filters.
   - Double-click the package filter.
   - In the Expression Definition box, update the namespace name.

Change the Package

If the name of the package that were used to create a report has changed, you must change the package connection to update the report.

At the same time, you can also change the authoring language for the report.

Procedure
1. From the File menu, click Report Package.
2. Click Another package and choose the new package.
3. To change the authoring language, click the ellipsis (...) button beside the Language box and choose a different report language.
   You may need to update the report to reflect the change. For example, any filter expressions in the report must be modified to reflect the syntax rules of the new language.
4. Click OK.
   The new package is loaded and the report is validated. If errors are found, the Validation Errors dialog box appears, showing the elements that must be updated to reflect the package change.
5. Click Close.
6. Make any required changes in the report to support the new package.
   For example, you may need to link the data items in the report to the new package.

Tip: You can use Query Explorer to make the changes.
Chapter 22. Upgrading Reports

When you open a report that was created in a previous version of IBM Cognos Business Intelligence, it is automatically upgraded.

Any problems detected during the upgrade process appear as information messages and error messages in the Upgrade Information dialog box. You must fix any errors in the report and then validate the report before you can run it. In some cases, the information or error message is linked to the location of the issue in your report. To go to the location of the issue, click the message, and then click Select. If only warnings and information appear in the dialog box, these will disappear when you click OK.

Tip: To view this dialog box again, from the File menu, click Upgrade Information.

After you upgrade a report to the most recent version of IBM Cognos BI, you can no longer open it with a previous version.

Lifecycle Manager

You can download IBM Cognos Lifecycle Manager from [http://www.ibm.com/](http://www.ibm.com/) to help you test your reports. Lifecycle Manager is a verification tool that checks that your reports run and produce the same results in the new environment.

Lifecycle Manager is a Microsoft Windows operating system-based application for auditing upgrades to the latest version of IBM Cognos BI from IBM Cognos ReportNet 1.1 MR3 or MR4, and from IBM Cognos 8 versions 8.2, 8.3, or 8.4.

It provides a verification feature that validates, executes, and compares report results from two different IBM Cognos BI releases. This helps to identify upgrade and compatibility issues between releases. User interface design and status reporting functionality provide both a proven practice process and support for upgrade project planning and status reporting. Lifecycle Manager also automates much of the process of bundling the required files, such as reports and models, for the test case. For more information, see the Lifecycle Manager User Guide.

Upgrading Reports from IBM Cognos BI Version 8.4

When you upgrade to IBM Cognos Business Intelligence version 10 from IBM Cognos 8 BI version 8.4, some reports may look or behave differently after the upgrade. This section describes changes that you may encounter in your reports.

Upgrading Report Styles

IBM Cognos Business Intelligence includes a new default report style with updated colors and gradients. If your report uses a custom report template, your report will appear the same in this version of IBM Cognos Report Studio as it did in previous versions. By default, new reports and new report objects, such as lists and crosstabs, appear in the new report style.
If you want to continue to work with the previous 8.x report style, set the **Override 10.x styles with 8.x styles on new reports** option (Tools, Options, Advanced tab).

You can update the style of an upgraded report to use the new 10.x style (File, Report Properties, Report styles, 10.x styles).

For more information about report styles, see “Create and Modify Report and Object Styles” on page 350.

**Upgrading Legacy Charts**

IBM Cognos Report Studio version 10.1.0 includes a new default chart technology. You can continue to use and work with the legacy charts or upgrade your legacy charts to the current default charts.

When you open a report that uses the legacy charts, the legacy charts is not upgraded to the current default charts automatically. You can upgrade your legacy charts one at a time to the current default chart. For more information, see “Convert Charts From One Type to Another” on page 102.

When you add a new chart to a report, Report Studio adds the current default charts. If you want to add new legacy charts or continue to work with existing legacy charts, set the **Use legacy chart authoring** option (Tools, Options, Advanced tab).

**New Default for Hidden or Deleted Axis Titles**

The default for rendering hidden or deleted axis titles has changed. This can change the size or placement of chart elements in your chart report output.

In previous versions of IBM Cognos Report Studio, hidden or deleted axis titles were defined and rendered as empty spaces in the report.

In Report Studio, Version 10.1.0, hidden or deleted axis titles are not rendered at all, so the space that was reserved for the title is available for other chart elements. This can affect the placement or size of chart elements, such as labels, the chart body, or the bars in a bar chart.

To make the charts appear the way they did in the previous version, set the x-axis’ **Title** property to Show and then set the x-axis title’s **Default Title** property to No. This adds the space for a custom axis title to the report specification but leaves the title blank.

**Upgrading Reports with SAP BW Prompt Variables**

When you upgrade reports that use SAP BW data sources and contain variables as prompts from version 8.3 SP2 to version 8.4 or 10.1.0, the upgrade may fail. Prompts that contained SAP BW variables are now empty. This occurs because a default setting in an IBM Cognos configuration file changed in version 8.4.

To successfully upgrade these reports, do one of the following:

1. Fully qualify all of the variables in the report by changing `variable_name` to `[infoquery].variable_name`.
2. Ask your IBM Cognos administrator to change the value for the DetectSAPVariableUniqueness parameter in the configuration file, as follows:
   - In the `c10_location\configuration` directory, locate the `qfs_config.xml` file.
• Find the DetectSAPVariableUniqueness parameter and change its value to false.
• Save the qfs_config.xml file.
• Restart the IBM Cognos service.

If more than one variable with the same name exists in the package, this change may cause problems.

---

**Upgrading Reports from IBM Cognos BI Version 8.1 or 8.2**

When you upgrade IBM Cognos Business Intelligence, some features in IBM Cognos BI may behave differently after the upgrade. When you upgrade reports, for example, changes in behavior may cause validation errors.

**Members Containing No Data Are Not Suppressed for SAP BW**

If you upgrade a crosstab report that uses a SAP BW data source and includes calculations, filters, or nesting, the suppression of null values may not occur as expected. You may see additional empty rows and columns.

By default, null values are suppressed for list and grouped list reports.

To effectively remove null values in crosstabs that use a SAP BW data source, insert individual members to create the report. You can also ensure that SAP BW members are assigned to proper dimension hierarchies and levels within the BW cube.

**Thousands Separators Missing**

You may encounter missing thousands separators in your reports in the following circumstances:
• The report contains an item that does not specify an explicit data format.
• The report item refers to another item in the same report.
• That second report item refers to an item in the IBM Cognos Framework Manager model with Usage property set to Identifier or Attribute.
• The model item does not specify an explicit format.

To restore the data formats, specify an explicit format either in the report item properties or in the model item properties.

**RQP-DEF-0177 Error When Upgrading Reports**

When you upgrade your report from IBM Cognos Business Intelligence, version 8.2, you encounter the following errors:

RQP-DEF-0177 An error occurred while performing operation 'sqlPrepareWithOptions' status='-120'.

UDA-SQL-0458 PREPARE failed because the query requires local processing of the data. The option to allow local processing has not been enabled.

This error occurs when the query requires local processing because some or all of its constructs are not supported by the database vendor. In IBM Cognos BI, version 8.2 and earlier, IBM Cognos BI proceeded with local processing. In version 8.3 and later versions, you must explicitly set the query Processing property to Limited Local in order to enable local processing.
Procedure

1. In IBM Cognos Report Studio, pause the pointer over the query explorer button and click the query.

2. In the Properties pane, set the Processing property to Limited Local.

Upgrading Reports from IBM Cognos ReportNet

When you upgrade from IBM Cognos ReportNet, some reports may look or behave differently after the upgrade. The upgrade does not account for the following issues.

Undocumented and Unsupported Features

If advanced report authors used undocumented and unsupported features such as JavaScript that refer to IBM Cognos HTML objects, they may have to recreate the features to complete the upgrade of the report.

No Sort Order

Data may appear in a different order after upgrading. If sort order is important, ensure that the report or model specifies a sort order before upgrading.

Layout Errors

Layout errors are suppressed by default in ReportNet. IBM Cognos Business Intelligence does not suppress layout errors. Users may have to correct or remove report layout expressions that cause errors.

Report Format

If you used the default format in ReportNet, upgraded reports will retain that format. However, new reports in IBM Cognos Business Intelligence will use the IBM Cognos BI format. If you want a consistent style across all reports, you must edit the styles property in each report and select or deselect the Use 1.x report styles option.

Customized Style Sheets

If you edited the ReportNet style sheet (default_layout.css) or the IBM Cognos Business Intelligence stylesheet (globalreportstyles.css), reports will lose the formatting after upgrading. You must reapply the changes to the IBM Cognos BI stylesheets and copy the stylesheets to the IBM Cognos BI server and the Web server.

PDF Reports

PDF rendering in IBM Cognos Business Intelligence behaves like HTML rendering. After upgrading, PDF reports may have different font size, column wrapping, or word wrapping. You may have to change each affected report or change the default font in the IBM Cognos BI style sheet.

IF-THEN-ELSE Statements

If you use assignments of different data types after THEN and ELSE in ReportNet, the reports will generate invalid coercion errors after upgrading. You may have to recast the variables or change the assignments in the affected reports.
Solve Order

In previous versions of IBM Cognos Business Intelligence, you could specify the solve order for objects such as crosstab node members. Solve order is now specified for data items. When you upgrade a report, solve orders specified in the report are moved to data items.

For more information about solve order, see “Resolve Multiple Calculations for Crosstabs and Charts” on page 540.

Chart Behavior

In ReportNet, if a chart is created with a user-specified minimum value and all data values are below the minimum value, the chart starts at the user-specified minimum value and contains no data.

In IBM Cognos Business Intelligence, the same parameters result in a chart that ignores the user-specified minimum value and uses a range that includes all of the data values for the chart.

Database Only Processing of Queries

If you specified in ReportNet that the processing for a query should be Database Only, when you upgrade your report to IBM Cognos Business Intelligence, some processing may now occur locally.

Changes in the Behavior of Functions Between ReportNet 1.1 and IBM Cognos BI

In ReportNet 1.1, double counting could occur when applying count or count distinct to a query item. This occurred primarily when querying multiple query subjects that were joined 1-to-N in the model if the counted item existed in the query subject on the 1 side of the join. The explanation is that count or count distinct was applied after the join operation.

Count distinct worked only when the item to which count distinct was applied was the column used in the join. In this case, the column to be counted could have identical values for the different values used in the join condition.

In IBM Cognos Business Intelligence, improvements to the count functionality cannot be handled through an automatic upgrade. The new approach avoids double counting on the 1 side of a 1-to-N join. The explanation is that count or count distinct is now applied before the join operation.

There is now a lesser need to use count distinct and, when used, it will be successful more often. Count distinct is no longer required to overcome double counting. Instead, it can be used as intended to select distinct values that exist in a query subject.

When you want to count repeated occurrences of a value, we recommend that you do one of the following:

• Apply a count operation on a column based on a query in which an explicit join occurs.
  This applies the count after the join.
• Count rows in a report by using a layout calculation object or by counting the literal value 1.
Appendix A. Accessibility Features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products. See the IBM Accessibility Center (http://www.ibm.com/able) for more information about the commitment that IBM has to accessibility.

Accessibility features in IBM Cognos Report Studio

There are several accessibility features in IBM Cognos Report Studio.

The major accessibility features are described in the following list:

- You can use command keys, or shortcut keys, to navigate through Report Studio. Shortcut keys directly trigger an action and usually make use of the Ctrl keys.
- Report Studio uses Web Accessibility Initiative—Accessible Rich Internet Applications (WAI-ARIA). This means that people with limited vision can use screen-reader software, along with a digital speech synthesizer, to listen to what is displayed on the screen.

Note: To take full advantage of the accessible features of Report Studio, use Mozilla Firefox version 4.0 or higher and Freedom Scientific JAWS version 12.0.

Keyboard Shortcuts

This product uses some standard Microsoft Windows and accessibility shortcut keys.

<table>
<thead>
<tr>
<th>Action</th>
<th>Shortcut keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables or disables the accessibility feature - an IBM Cognos Report Studio restart is required. The default is disabled.</td>
<td>Ctrl+Shift+Tab</td>
</tr>
<tr>
<td>Opens the context menu for the selected item, if available.</td>
<td>Shift+F10</td>
</tr>
<tr>
<td>Switches focus from or to the main menu bar and the main worksheet.</td>
<td>Ctrl+F10</td>
</tr>
<tr>
<td>Opens the online help.</td>
<td>F1</td>
</tr>
<tr>
<td>Closes the Web browser window.</td>
<td>Alt+F4</td>
</tr>
<tr>
<td>Copies objects.</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Pastes objects.</td>
<td>Ctrl+V</td>
</tr>
<tr>
<td>Closes objects, such as a dialog box, menu, drop-down list, or page.</td>
<td>Esc</td>
</tr>
<tr>
<td>Cycles through objects, such as the tabs in a dialog, the objects in a page, the cells in a list, the rows in a crosstab, or check boxes.</td>
<td>Tab, Shift+Tab, Arrow keys</td>
</tr>
<tr>
<td>Selects the first or last item in a list.</td>
<td>Home/Pg Up or End/Pg Dn keys</td>
</tr>
</tbody>
</table>
**Considerations to Improve Report Accessibility**

Creating accessible reports ensures access of information to all users, with all levels of ability.

For example, people with a visual impairment may use screen reading technology to access the information in a report.

The following are some design considerations for creating accessible reports:

- Avoid using visual cues, such as bold text or color, to convey important information.
- Avoid using pictures and OLE Objects in PDF documents, as these items are tagged as artifacts and ignored by the screen reader.
- Avoid using conditional formatting to convey important information.
- When selecting color palettes for report objects, choose patterns or shades of gray.
- Ensure that there is a table corresponding to chart types that are rendered as images because the screen reader ignores this information.
- Deliver reports in HTML format, which is the most supported output format for most screen readers.
- Ensure that the report has a title.
- Gain an understanding for screen reading technology.
- Avoid spelling and grammatical errors, as they cause the screen reading software to misinterpret the information.
- Avoid using features like calendar boxes and up and down selections on time controls. Instead use prompts such as check boxes, radio buttons, combo boxes, and multi-select boxes.
- Ensure that the target application is accessible when using embedded Web applications or drill-through paths.
- Avoid using large, complex list or crosstab reports.
  Displaying the information in multiple simple lists or crosstab reports is more manageable for assistive technology users.
- Add alternate text to images, charts, and other visual objects so that screen readers can provide context for them.
- When using tables, add summary text to provide context for the table content. If the top cells in a table behave as headers, designate these cells as headers so that screen readers can identify the relationships.
- Avoid using range prompt controls that contain radio buttons.
When users tab to a radio group, the focus should be in the selected radio button. In the case of range prompts, for both the From and To sections, the radio button groups start first in the tab order. However, the secondary input elements, which can be text boxes, drop-down lists, date edit boxes, and so on, are located before the selected radio button in the tab order. Instead of going to the selected radio button, the focus goes to the secondary input element. As a result, users using a screen reader do not know that the input element is one of two options.

Enable Accessible Report Outputs

If you want to include accessibility features, such as alternate text, summary text, designated cell headers in tables and accessible conditional layouts, you must enable these accessibility features in the report output.

You can enable accessible report outputs in one of the following ways:
- in the IBM Cognos Report Studio run options, so that the report has accessibility features enabled when you run the report from within Report Studio.
- in IBM Cognos Connection as a run option (Run with options, Enable accessibility support), so that report consumers can run the report once with accessibility features.
- in IBM Cognos Connection as a property (Set properties, Report tab), so that report consumers can always run the report with accessibility features.
- in IBM Cognos Connection, as a user preference (My area, My preferences), so that report consumers can enable accessibility features for all of their reports. Accessibility settings in the report properties overwrite this setting.
- in IBM Cognos Administration, as a server-wide option, so that all reports for all IBM Cognos users have accessibility features enabled. Accessibility settings in the user preferences and report properties overwrite this setting.

Administrators can also change a system-level setting that forces the accessibility features on or off regardless of any other settings.

For information about the last four options, see the IBM Cognos Connection User Guide or the IBM Cognos Administration and Security Guide.

Procedure

In Report Studio, from the Run menu, click Run Options and select the Include accessibility features check box.

Alternate text and summary text

To make reports accessible, you can add alternate text or summary text to objects such as images, charts, and tables. When a screen reader encounters one of these objects, it reads the text that you added to the object.
Related tasks:

“Customize Prompt Text” on page 323

You can customize the instructional text that appears around prompts. For example, a value prompt with multiple selections includes a **Select all** link below the choices that you can customize to text other than **Select all**. You can also specify translated text in many languages for international report users.

Add Alternate Text to Images and Charts

You can add alternate text for images, maps, and charts to make your reports accessible. When a screen reader encounters one of these objects, it reads the alternate text that you added to the object.

You can add translations for the text to support users in multiple languages. When users run a report, IBM Cognos Business Intelligence uses the alternate text in the appropriate language.

If you use images only for visual spacing in your report, leave the **Alternate Text** property empty.

**Before you begin**

For the report output to contain the alternate text, you must enable the accessibility features for the report. For more information, see “Enable Accessible Report Outputs” on page 469.

**Procedure**

1. Select the image or chart object.
2. In the **Properties** pane, click the select ancestor button and select the **Image**, **Map**, or chart object.
3. Double-click the **Alternate Text** property.
4. Select **Specified text** and click the ellipsis (...) button.
5. In the **Default text** box, type a description for the object, and click the add icon .
6. In the **Languages** dialog box, select the languages that apply to your text.
7. Double-click a language and type the translation of the text for each language and click **OK**.

**Add Summary Text to Tables**

You can provide summary text for crosstabs, lists, repeater tables, and table objects. This text provides context for the entire object to make your reports accessible. When a screen reader encounters one of these objects in HTML report outputs, it reads the description that you added to the object.

The table summary is not displayed in visual Web browsers. Only screen readers and speech browsers use the summary text. The summary text is usually read immediately before the table caption.

You can add translations for the text to support users in multiple languages. When users run a report, IBM Cognos Business Intelligence uses the summary text in the appropriate language.
If you use tables for report layout, leave the summary empty to indicate to screen readers that the table is used exclusively for visual layout and not for presenting tabular data.

**Before you begin**

For the report output to contain the summary text, you must enable the accessibility features for the report. For more information, see “Enable Accessible Report Outputs” on page 469.

**Procedure**

1. Select the crosstab, list, or table.
2. In the Properties pane, click the select ancestor button and select the Crosstab, List, Repeater Table, or Table object.
   
   **Tip:** You can also click the container selector (three orange dots) of the container to select it.
3. Double-click the Summary Text property.
4. Select Specified text and click the ellipsis (...) button.
5. In the Default text box, type a description for the object, and click the add icon.
6. In the Languages dialog box, select the languages that apply to your text.
7. Double-click a language and type the translation of the text for each language.

**Designate Cells Headers in Tables**

You can specify whether specific table cells are table headers. This allows screen readers and speech browsers to identify the relationships between the cells in your tables.

**Before you begin**

For the report output to contain the cell headers, you must enable the accessibility features for the report. For more information, see “Enable Accessible Report Outputs” on page 469.

**Procedure**

1. Select the table cells.
2. In the Properties pane, set the Table Header property to Yes.

**Headings and emphasis**

You can apply certain global classes to text objects to make the text easier to read for visually impaired users.

The following global classes are available to improve text readability for headings and blocks of text:

- Heading 1 through 6

  **Tip:** In HTML output, these classes are converted to <h1> to <h6> tags.

- Emphasize Text and Strong Text

  **Tip:** In HTML output, these classes are converted to <em> and <strong> tags.
You apply one of these global classes to text by accessing the **Classes** property of the text and choosing the class that you want to apply.

**Prompt control labels**

You can associate labels to prompt controls to make the prompts accessible.

There are several methods that you can use to associate a label to a prompt control:

- Associate a text item to the prompt control.
- Insert the prompt control in a field set and use the caption of the field set to specify the label.
- Use HTML items to define a field set.

When a text item is associated to a prompt control, the text is read by screen readers when users tab to the control. When a prompt control is inserted in a field set, screen readers read the label of the field set when the focus moves to any input element within the field set.

**Associating a text item to a prompt control in IBM Cognos Report Studio**

Associate a text item to a prompt control to create a label for the prompt.

**Before you begin**

You must specify a name for the prompt control before you can associate a text item to the control.

**Procedure**

1. To specify a name for the prompt control, select the control, and in the Properties pane, type the name in the **Name** property.
2. Create a text item or select an existing text item.
3. In the Properties pane, in the **Label For** property, type the name of the prompt control.

**Associating labels with prompt controls using field sets**

You can associate labels with prompt controls using field sets.

**Procedure**

1. From the **Toolbox** tab, drag **Field Set** to the report.
2. Double-click the caption of the field set and type the text that you want to appear as the prompt control label.
3. From the **Toolbox** tab, drag a prompt control to the field set.
4. Optional: To hide the field set border so that only the prompt control border appears in the prompt page, set the border to none.
   a. Select the field set.
   b. In the **Properties** pane, click the **Border** property.
   c. In the **Style** box, click **None**.
   d. Under **Preview**, click the apply all borders icon and click **OK**.
**Associating labels with prompt controls using HTML items**

Instead of using the field set object, you can associate labels with prompt controls using HTML items. You use the HTML items to create the field sets.

**Procedure**

1. From the Toolbox tab, drag HTML Item to the left of the prompt.
2. Click the HTML item.
3. In the Properties pane, double-click the HTML property.
4. In the HTML window, type the following text and click OK.
   
   ```html
   <fieldset style="border:0;">
   <legend>
   Prompt label
   </legend>
   </fieldset>
   ```

5. From the Toolbox tab, drag HTML Item to the right of the prompt.
6. Click the HTML item.
7. In the Properties pane, double-click the HTML property.
8. In the HTML window, type the following text and click OK.
   
   ```html
   </fieldset>
   ```

**Example - Conditionally Show a List Below a Chart for an Accessible Report**

Charts are rendered as images in report outputs, such as HTML and PDF. As a result, they are difficult to navigate for visually impaired users and screen readers cannot convey the information shown in charts. To make your reports accessible, you can add a conditional layout that shows list or crosstab equivalents of the chart when the accessibility features are enabled for the report output.

To set up this conditional layout, use a conditional block that contains the list or crosstab and the report function IsAccessible. The IsAccessible function is a Boolean function that returns Yes (or true) when the accessibility features are enabled for the report output and No (or false) when the accessibility features are not enabled.

Therefore, the list or crosstab becomes conditional and appears only when the accessible report output option is enabled. In addition, you should add alternate text for the chart object so that screen readers can let users know that an accessible list or crosstab follows.

**Procedure**

1. Open IBM Cognos Report Studio with the sample GO Sales (query) package.
2. Create a new report with a pie chart.
3. Add data items to the chart. From the Source tab, do the following:
   - Expand Sales (query) and Sales and then insert Quantity in the Default measure area.
   - Expand Products and insert Product line in the Series (Pie Slices) area.
4. Run the report to see your chart.
5. Add alternate text for the pie chart:
   - Select the chart and, in the Properties pane, double-click the Alternate Text property.
Select Specified text and click the ellipsis (...) button.

Type default text for the chart, such as

A pie chart shows the quantity of sales by product line as a percentage of total sales. An equivalent list is below.

If you want to add the alternate text in additional languages, click the add button.

6. From the Toolbox tab, insert a Conditional Blocks object below the chart.

7. Select the conditional block and, from the Properties pane, double-click the Block Variable property.

8. Under Variable, select New boolean variable and type a name for the variable, such as accessibility_condition.

9. In the Report Expression dialog box, specify the expression that defines the variable:

   - From the Functions tab, expand Report Functions and insert the IsAccessible function to the Expression Definition box.
   - Type a closing bracket at the end of the function IsAccessible () and click OK.

10. Select the conditional block and in the Properties pane, change the Current Block property to Yes.

11. From the Toolbox tab, add a list inside the conditional block.

12. Add the same data items from the chart to the list. From the Source tab, do the following:

    - Expand Sales (query) and Products and insert Product line in the list.
    - Expand Sales and insert Quantity as the second column in the list.

13. Add summary text for the list

    - Select the list and in the Properties pane, double-click the Summary Text property.
    - Select Specified text and click the ellipsis (...) button.
    - Type default text for the list, such as

        The first column lists all product lines and the second column lists sales quantities.

        If you want to add the alternate text in additional languages, click the add button.

14. Select the conditional block and in the Properties pane, change the Current Block property to No.

15. From the Run menu, click Run Options and select the Include accessibility features check box.

16. Run the report.

Results

Because we just chose to include the accessibility features, you see both the chart and the list. Also, when a screen reader encounters the image for the chart, it reads the alternate text that you added.
If you clear the **Include accessibility features** check box and run the report again, you see only the chart.

![Pie chart and list](image)

*Figure 54. Report showing both a pie chart and a list*

**IBM and accessibility**

See the IBM Accessibility Center for more information about the commitment that IBM has to accessibility.

The IBM Accessibility Center is available online at [http://www.ibm.com/able](http://www.ibm.com/able)
Appendix B. Troubleshooting

This appendix describes some common problems you may encounter.

For more troubleshooting problems, see the IBM Cognos Business Intelligence Troubleshooting Guide.

Problems Creating Reports

The topics in this section document problems you may encounter when creating reports.

Division by Zero Operation Appears Differently in Lists and Crosstabs

If you have a list that accesses a relational data source, a calculation containing a division by zero operation appears as a null value, such as an empty cell. In a crosstab, the division by zero operation appears as /0. This happens when the Avoid Division by Zero property is set to Yes, which is the default.

To have a consistent display of null values in lists and crosstabs, define an if-then-else statement in the expression in the crosstab cell that changes the value /0 to the value null.

Application Error Appears When Upgrading a Report

When upgrading a report, the following error appears if the report contains data items in the page layout that are not in a data container:

RSV-SRV-0040 An application error has occurred. Please contact your Administrator.

This error occurs when IBM Cognos BI cannot determine the query reference for a data item. Such data items are identified by a small red circle with a white x icon that appears in the lower left corner.

To correct the error, drag the data items into a container. If the container is a list, we recommend that you drag the data items into the list page header or footer, or the overall header or footer. If you want to see the first row of the item on each page or in the overall report, drag the item to the list page header or overall header. If you want to see the item's last row on each page or in the overall report, drag the item to the list page footer or overall footer.

Tip: If a header or footer does not exist, create it.

Nested List Report Containing a Data Item That is Grouped More Than Once Does Not Run After Upgrade

When you upgrade a nested list report that contains a data item that is grouped in both lists, the report does not run.

The following error occurs when the report is run against a dimensional data source and both lists are using the same query. This error does not occur if the report is run against a relational data source.
OP-ERR-0199: The query is not supported. The dimensions on the edge are inconsistent. The dataItems from dimension="[Product line]" must be adjacent.

For example, you have a list that contains the grouped items Product line and Product type and a nested list that contains the data items Year, Quarter, Product line, and Unit sale price. Year, Quarter, and Product line are grouped items in the nested list.

To resolve the issue, delete the data item that is grouped in both lists from the inner list.

**Procedure**

1. Click anywhere in the report.
2. In the Properties pane, click the select ancestor button and click the List link that represents the inner list.
3. Double-click the Grouping & Sorting property.
4. In the Groups pane, select the data item that you want and click the delete button.

**Background Color in Template Does not Appear**

When creating a Query Studio template in Report Studio, if you add a list object and change its background color, the color change does not appear when you apply the template to a Query Studio report.

To work around this issue, do one of the following:

- Edit the style sheet (CSS) classes for lists in Report Studio.
- Do not add any objects to the page body when you are creating a Query Studio template.
- Leave the page body blank.

**Subtotals in Grouped Lists**

When using an IBM Cognos PowerCube that contains a ragged hierarchy, if you group on the first level in the hierarchy, subtotals may appear in the wrong place or show wrong values.

To resolve the issue, group on the second level.

**Chart Labels Overwrite One Another**

In Report Studio and Query Studio, if you define a chart and render it in HTML or PDF format using the default sizes, the axis labels of the chart may overwrite each other.

To avoid this problem, make the chart wider or taller by modifying the height and width properties of the chart or enable the Allow Skip property.

**Chart Shows Only Every Second Label**

You create a report that includes a chart. The Allow Skip option is set to false, but when you run the report, labels are skipped.
This can occur if there is not enough room for all labels and the options Allow 45 Degree Rotation, Allow 90 Degree Rotation, and Allow Stagger are also set to false. IBM Cognos BI has no options for making the labels fit, so it skips every second label.

The solution is to select either Allow 45 Degree Rotation, Allow 90 Degree Rotation, or Allow Stagger.

**Chart Gradient Backgrounds Appear Gray in Internet Explorer**

In Report Studio, you can define a custom palette for a chart that includes a gradient. When the chart is rendered in HTML format in Microsoft Internet Explorer, the chart background appears gray. This is an Internet Explorer issue.

To avoid this problem, select the chart and define the color white as the chart background.

For more information, see the Microsoft Knowledge Base article # 294714 at [http://support.microsoft.com](http://support.microsoft.com)

**Metadata Change in Oracle Essbase Not Reflected in Reports and in the Studios**

When there is a metadata change on the Oracle Essbase server, the change is not immediately reflected in the metadata tree in the studios. In addition, when a report is run, the report does not pick up the republished changes.

To view the new structure, you must restart the IBM Cognos Content Manager server.

**Relationships Not Maintained in a Report With Overlapping Set Levels**

In a report, the relationship between nested or parallel member sets at overlapping levels in the same dimension may not always be maintained.

For example, a named set in the data source that contains members from both a Year and Month member is nested under Year, but is not properly grouped by year.

In another example, an error message such as this appears:

*OP-ERR-0201 Values cannot be computed correctly in the presence of multiple hierarchies ([Product].[B1], [Product].[Product]) that each have a level based on the same attribute (Product).*

This problem occurs in the following scenarios involving non-measure data items X and Y, which overlap in the same dimension:

- X and Y together as ungrouped report details
- Y nested under X
- Y appended as an attribute of a group based on X

When using named sets, or sets that cover more than one level of a hierarchy, do not use sets from the same dimension in more than one place in the same report. They should appear on only one level of one edge.
Summaries in Query Calculations Include Nulls with SAP BW Data Sources

When using an SAP BW data source in IBM Cognos Report Studio, null values in the database are returned in the result set and the count summary function includes the empty cells in the following scenarios:

- A query calculation includes an arithmetic calculation where one or more NULL operands and an aggregation is performed on the calculation.
- The result of a query calculation is a constant, such as current_time and current_date.

The count summary function should normally exclude null values.

To avoid this problem, for the first scenario, ensure that both operands do not return null values. For example, the original expression is [num1]+[num2]. Instead, use the following expression:

\[
\begin{align*}
\text{if } ([\text{num1}] \text{ is null}) \text{ then } (0) \text{ else } ([\text{num1}]) \\
\text{if } ([\text{num2}] \text{ is null}) \text{ then } (0) \text{ else } ([\text{num2}])
\end{align*}
\]

There is no workaround for the second scenario.

Creating Sections on Reports That Access SAP BW Data Sources

SAP BW data sources may have problems with sections in reports under different circumstances:

If a section in a report uses the lowest-level query item in a ragged hierarchy, such as the children of the not assigned node, the following BAPI error may appear:

\[\text{BAPI error occurred in function module BAPI_MDDATASET_SELECT_DATA. Value <valueName> for characteristic <cubeName> unknown}\]

Lowest-level Query Item in a Ragged Hierarchy

The solution is to remove the section from the lowest-level query item.

Several Multicubes with SAP Variables

The solution is to use one SAP multicube when creating sections in reports.

Related concepts:

**“Tips for Working with Ragged or Unbalanced Hierarchies” on page 264**

In ragged or unbalanced hierarchies, some members that are not at the lowest level of the hierarchy may have no descendants at one or more lower levels. Support for these hierarchy gaps in relational data sources is limited. More complete support is provided for OLAP data sources, but some reports may still result in unexpected behavior.

Error Characters (--) Appear in Reports

When you run a report, you see two dash (--) characters in your report instead of values.
These characters may appear if you use an OLAP data sources other than PowerCube and Microsoft SQL Server 2005 Analysis Services (SSAS), and you apply aggregation to calculations and measures that use rollups other than Sum (Total), Maximum, Minimum, First, Last, and Count.

All other types of rollup either fail or return error cells, which typically display as two dash characters (--).

This problem occurs in, but is not limited to, the following:
- footers
- aggregate function
- summary filters and detail filters that use a summary
- detail, summary, and context filters that select more than one member of a hierarchy that is used elsewhere on the report

If you are working with a SSAS 2005 data source, these characters may also appear in summary cells if you use an OR filter in the summary. To avoid this problem, do not use OR filters in summaries.

**Columns, Rows, or Data Disappear With SSAS 2005 Cubes**

Microsoft SQL Server 2005 Analysis Services (SSAS) has a feature called AutoExists that removes tuples that have no facts at the intersection of two hierarchies of the same dimension.

Columns, rows, or data can disappear if you set the default member of a hierarchy to a member that does not exist with every other member in the dimension. To avoid this problem, change the default member that caused the disappearance to a member that exists with all other members in the dimension.

Columns, rows, or data can also disappear if members are specified that result in one or more non-existent tuples. There is currently no workaround for this scenario. For more information, see Microsoft Knowledge Base article #944527 at [http://support.microsoft.com](http://support.microsoft.com).

You may also encounter unexpected results if the default member of a hierarchy is a member that doesn't also exist in all other hierarchies in the dimension, and if you query members from different hierarchies in the same dimension.

For example a crosstab includes the following (using the Adventure Works cube):
- Rows: Generate([Adventure_Works].[Account].[Accounts],set([Balance Sheet],[Units])) nested with children([Adventure_Works].[Department].[Departments]->[YK].[Department].[Departments].&[1]))
- Column: [Adventure_Works].[Account].[Account Number].[Account Number]
- Measure: [Adventure_Works].[Measures].[Amount]

You run the report and notice that the query renders with some blanks cells. You then apply the simple detail filter [Amount]>1 and run the report. Only row labels are displayed and all data and columns are missing.

In the Adventure Works cube, the [Account].[Accounts] attribute has a default member set to [Net Income]. When evaluating the GENERATE set expression, SSAS looks in the entire cube space and looks at all coordinates for the [Account]
dimension. These coordinates include both [Account][Account Type]&[] and [Account],[Accounts],[Net Income]. Because these two coordinates don't exist within the same hierarchy, SSAS returns an empty set.

To avoid this problem the SSAS administrator must set the default member in the cube to a member that exists in all other hierarchies.

Function Unreliable with Sets
If you create an expression that uses the descendants function with sets, you may encounter unpredictable results. Some expected members may be missing or may have blank captions or labels.

This problem occurs if the descendants function uses a set as its first parameter instead of a single member and if the descendants function is nested under another data item from the same hierarchy.

To avoid this problem, replace the first parameter in the descendants function with the function currentmember(H), where H is the hierarchy of the desired set and under which the expression is nested. For example, use descendants(currentmember(H)).

Searching for values might return unexpected results
In the expression editor, when searching for values for a data item, the results you obtain might contain unexpected results if the data item is not a string data type. Because you can edit the expression for a data item, IBM Cognos Business Intelligence cannot determine with certainty what the data type is.

Therefore, IBM Cognos Business Intelligence guesses the data type of the data item by looking at its aggregate and rollup aggregate set.

Related tasks:
"Browse or Search the Values of a Data Item" on page 233

When building expressions in the expression editor, you can browse the data of a data item. This is useful when you do not know how a particular value is stored in the database. For example, you want to filter a report so that it shows data for only New South Wales. The actual value in the database for New South Wales is NSW, and this is what you must insert in the filter expression to obtain the correct results.

Report Differences Between TM1 Executive Viewer and IBM Cognos Business Intelligence with TM1 Data Sources
When using an IBM Cognos TM1 data source, comparable reports created in an IBM Cognos Business Intelligence studio and in TM1 Executive Viewer may contain different cell values. This occurs because the TM1 Executive Viewer product uses an algorithm for selecting default members for non-projected dimensions that differs slightly from traditional OLAP clients.

To avoid this problem, when filtering your reports in the IBM Cognos Business Intelligence studios, use context filters that match the default selections shown in the Executive Viewer user interface. This ensures that the cell values in IBM Cognos Business Intelligence match the values in Executive Viewer.
Order of Metadata Tree Differs for TM1 Data Sources

When using a TM1 data source, the order of members in the metadata tree of the Source tab of an IBM Cognos Business Intelligence studio may differ from the order shown in TM1 Architect.

By default, TM1 Architect renders members of hierarchies using a slightly different algorithm than does IBM Cognos BI. IBM Cognos BI automatically renders member metadata from TM1 data sources in hierarchical order.

From within TM1 Architect, if you want to see how an IBM Cognos BI studio will render a hierarchy, click the Hierarchy Sort button.

MSR-PD-0012 error when importing external data

When you try to import an external data file, you receive an MSR-PD-0012 error.

MSR-PD-0012: Unable to upload the specified external data file. It exceeds the permitted file size of “0(KB)”, as specified by your system administrator.

This error occurs when the size of the file you are trying to import is greater than the value specified for the Maximum external data file size (KB) governor in the Framework Manager model.

To resolve the issue, the modeler must update the governor, save the model, and republish the package.

MSR-PD-0013 error when importing external data

When you try to import an external data file, you receive an MSR-PD-0013 error.

MSR-PD-0013: Unable to upload the specified external data file. It exceeds the permitted maximum number of rows “0”, as specified by your system administrator.

This error occurs when the number of lines in the file you are trying to import is greater than the value specified for the Maximum external data row count governor in the Framework Manager model.

To resolve the issue, the modeler must update the governor, save the model, and republish the package.

Problems Calculating Data

The topics in this section document problems you may encounter when calculating or summarizing data.

Unexpected Summary Values in Nested Sets

If a report contains nested sets, summaries other than the inner set summaries may contain unexpected values. For example, you insert a summary in a crosstab that contains a set with years in the rows.
You then nest a product line set within years.

Notice that the summary value does not change to represent the total of the new values. This occurs because the within set aggregation used with dimensional packages does not take into account sets that are nested below the set that is summarized.

To show the correct summary values, if the inner and outer sets do not belong to the same dimension, you can nest a copy of the inner summary item under the outer summary item, as follows.

Null Results for Calculations Using SAP BW Data Sources

When using a SAP BW data source, the expression you use in your calculation is evaluated as a null value if your expression contains a null item. For example, in the calculation some_expression = result, the result is null if a row or column that the expression references includes a null value.

To avoid obtaining null values as the result of your calculations, suppress null values before you create the calculation.

Incorrect Results with IBM Cognos PowerCubes and Time Measures

If a report uses an IBM Cognos PowerCube data source and a combination of data items, you will encounter incorrect results.
The following combination of data items in a report that uses an IBM Cognos PowerCube data source will give incorrect results.

- a measure with a **Time State Rollup** set to **Average** or **Weighted Average**
- an aggregate (**members from time dimension**) expression
- an intersection with a member in a relative time hierarchy

To avoid incorrect results, do not use this combination in your reports.

### Incorrect Results in Summaries When Using OLAP Data Sources

When using an OLAP data source, summaries that use `for` clauses give incorrect results.

This occurs because `for` clauses require access to the detail rows of the fact table. OLAP data sources do not have detail rows.

For example, this report uses a dimensionally-modeled relational (DMR) data source and contains the following summaries:

- `mx`: maximum ([Revenue] for [Year (ship date)])
- `mx2`: maximum (Aggregate([Revenue]) for [Year (ship date)])

![Figure 58. Example list report that uses a dimensionally-modeled relational data source and revenue for four years](image)

Notice that the `mx` and `mx2` values are different, where `mx2` is based on visible data, but `mx` is not. This result is correct.
The following report uses an OLAP data source and contains the same summaries.

<table>
<thead>
<tr>
<th>Year</th>
<th>Quarter</th>
<th>Revenue</th>
<th>mx</th>
<th>mx2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>Q 1</td>
<td>221,704,706.31</td>
<td>235,750,316.25</td>
<td>235,750,316.25</td>
</tr>
<tr>
<td></td>
<td>Q 2</td>
<td>222,143,384.57</td>
<td>235,750,316.25</td>
<td>235,750,316.25</td>
</tr>
<tr>
<td></td>
<td>Q 3</td>
<td>235,750,316.25</td>
<td>235,750,316.25</td>
<td>235,750,316.25</td>
</tr>
<tr>
<td></td>
<td>Q 4</td>
<td>234,754,397.59</td>
<td>235,750,316.25</td>
<td>235,750,316.25</td>
</tr>
<tr>
<td>2011</td>
<td>Summary</td>
<td>941,352,803.72</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td>Q 1</td>
<td>295,228,480.53</td>
<td>306,706,702.72</td>
<td>306,706,702.72</td>
</tr>
<tr>
<td></td>
<td>Q 2</td>
<td>278,180,796.96</td>
<td>306,706,702.72</td>
<td>306,706,702.72</td>
</tr>
<tr>
<td></td>
<td>Q 3</td>
<td>261,079,666.95</td>
<td>306,706,702.72</td>
<td>306,706,702.72</td>
</tr>
<tr>
<td></td>
<td>Q 4</td>
<td>306,706,702.72</td>
<td>306,706,702.72</td>
<td>306,706,702.72</td>
</tr>
<tr>
<td>2013</td>
<td>Summary</td>
<td>1,159,495,590.16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Q 1</td>
<td>344,124,267.07</td>
<td>391,874,462.51</td>
<td>391,874,462.51</td>
</tr>
<tr>
<td></td>
<td>Q 2</td>
<td>391,874,462.51</td>
<td>391,874,462.51</td>
<td>391,874,462.51</td>
</tr>
<tr>
<td></td>
<td>Q 3</td>
<td>375,118,012.54</td>
<td>391,874,462.51</td>
<td>391,874,462.51</td>
</tr>
<tr>
<td></td>
<td>Q 4</td>
<td>381,774,358.76</td>
<td>391,874,462.51</td>
<td>391,874,462.51</td>
</tr>
<tr>
<td>2015</td>
<td>Summary</td>
<td>1,495,894,106.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2016</td>
<td>Q 1</td>
<td>471,624,387.89</td>
<td>479,209,923.82</td>
<td>479,209,923.82</td>
</tr>
<tr>
<td></td>
<td>Q 2</td>
<td>479,209,923.82</td>
<td>479,209,923.82</td>
<td>479,209,923.82</td>
</tr>
<tr>
<td></td>
<td>Q 3</td>
<td>166,441,982.96</td>
<td>479,209,923.82</td>
<td>479,209,923.82</td>
</tr>
<tr>
<td></td>
<td>Q 4</td>
<td>479,209,923.82</td>
<td>479,209,923.82</td>
<td>479,209,923.82</td>
</tr>
<tr>
<td>2017</td>
<td>Summary</td>
<td>1,147,336,274.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Summary</td>
<td>4,086,775,788.85</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 59. Example list report that uses a dimensional data source and revenue for four years

In the example report, mx and mx2 values are now the same. Both summaries are based on visible data. The mx value is incorrect.

Incorrect results also appear for footer summaries.

To avoid this problem, when using OLAP data sources, ensure that the parameter that precedes the FOR clause is an aggregate function.

Problems Filtering Data

The topics in this section document problems you may encounter when filtering data.

Unexplained Discrepancies in Number Calculations

You might find unexplained discrepancies in number calculations due to round-off errors.

For example:

- You run regression tests and find differences in numbers. They are different only because of the rounding off of decimal places.
- You choose not display zeros in reports, but the zeros are displayed anyway because there are decimal places (0.00000000000000426, for example) that are rounded off to zero in reports.
Round-off problems are not specific to IBM Cognos software. They can occur in any environment where rounding off occurs.

**Binary Round-Off Errors**

Discrepancies in calculations might occur due to binary round-off errors. For example, if the number 1.1 is represented as a binary floating point number and your report format includes a large number of decimal places, the number 1.1 might actually be something like 1.09999999999997.

If your report is formatted to use only one decimal point, decimal round-off takes place, compensating for the binary round-off. So the number appears to be 1.1 when it is really 1.09999999999997. When the number is used in calculations, you might get round-off errors. For example, Microsoft Excel calculations use binary numbers (without rounding off decimal places) but formatting in reports shows rounded off decimal places, which can create small discrepancies.

**Division Round-Off Errors**

Calculations that involve division typically incur round-off errors, regardless of how the numbers are represented. Examples of such calculations are Average and Percent of Base.

**Design Guidelines to Minimize Round-Off Effect**

The best solution is to change the underlying database schema or cube model but that may not always be possible. Another solution is to minimize the round-off effect by following these guidelines when authoring reports and creating models in IBM Cognos Framework Manager and external OLAP cubes:

- Avoid storing data in floating point format whenever possible. This is especially true for currency values, which should be stored as either fixed-point decimals or as integers with a scale value such as 2.
  
    For example, in a cube, the Revenue for Camping Equipment in 2012 is $20,471,328.88. If revenue details are stored as floating point numbers, round-off errors might occur when revenue is calculated.

    The round up errors might have slight differences, depending on the order of calculation. If revenue for Products is calculated first and revenue for Time is calculated second, you might get a different round-off error than if Time is calculated first and Products is calculated second.

    Total revenue might be calculated as the number above. Or there might be slight discrepancies, for example, $20,471,328.8800001 as opposed to $20,471,328.88. The internal number might be slightly different than what is displayed. The number might even be for different runs of the same report, depending on the order that the OLAP engine uses for calculation.

- In reports, avoid division whenever possible. When division is unavoidable, try to do it as late as possible in the calculation process. For example, instead of Total([Revenue]/1000), use Total([Revenue])/1000.

- When doing comparisons, add a margin to allow for round-off. For example, you may want [Profit %] to be a fractional value formatted as a percentage with no decimals. However, the filter [Profit %]<0 (or [Profit %] NOT BETWEEN 0 and 0) rejects zero values and may still return values that appear to be 0% after formatting.

To avoid this, filter in one of these two ways:

- [Profit %] NOT BETWEEN -0.005 and 0.005
\[-(\text{Profit }\%) \leq 0.005) \text{ OR } (\text{Profit }\%) > 0.005]\]

Note that 0.005 is equivalent to 0.5\%, which displays as either 0\% or 1\%, depending on floating point precision losses.

In some cases, you may prefer control round-off errors by rounding values explicitly. For example, instead of \[\text{Profit }\%\], use \text{round}(\text{Profit }\%, 2).

\[\bullet\text{ Recalculate numbers every time instead of reusing calculations that might contain rounded off decimals.}\]

There might be additional considerations for Microsoft Analysis Services 2005/2008, especially when comparing report results from different runs (as happens in Lifecycle Manager). Refer to Microsoft documentation for more information.

**HRESULT= DB_E_CANTCONVERTVALUE Error When Filtering on a _make_timestamp Column**

You cannot filter on a _make_timestamp column, and the following error messages appear:

\[UDA-SQL-0114\text{ The cursor supplied to the operation “sqlOpenResult” is inactive}\]

\[UDA-SQL-0206\text{ The OLEDB driver returned the following value: HRESULT= DB_E_CANTCONVERTVALUE}\]

\[RSV-SRV-0025\text{ Unable to execute this request}\]

The solution is to apply the filter after aggregation and not before.

**Problems Running Reports**

The topics in this section document problems you may encounter when viewing or running reports.

**Report Runs Slowly**

The following is a list of questions that will help you to troubleshoot a slow report.

\[\bullet\text{ Does your IBM Cognos environment conform with the supported environments?}\]

\[\bullet\text{ Has the report always been slow or did it recently become slow?}\]

If it recently became slow, can you identify an event that occurred just before the report began to run slowly? Events could include changes to configuration settings, changes to tuning settings, a recent upgrade where your previous settings have not been applied, an introduction of firewalls or proxies, changes to existing firewalls or proxies, changes to virus scans on temp directories, or temporary table space restrictions on the database. This event could have caused the change in report performance.

\[\bullet\text{ Is the performance slow for all reports or just one report?}\]

If all reports are slow, the issue may be due to your environment or database. If all reports from a specific package are slow, the issue may due to the model design. If just one report is slow, the issue may be due to a specific report element.

\[\bullet\text{ How many queries does your report contain?}\]

The number of queries on the report will proportionally affect the report execution time.

\[\bullet\text{ Does the report run slowly for everyone, or just for one user?}\]
If the report runs slowly for just one user, the issue may be due to something in that user’s environment, such as virus scanning, page file size or location settings, or their location on the network.

- Is the report burst or run often by many people?
  If many people are running the same report at the same time, you may need to scale your environment or consider using dispatcher routing rules to direct all requests for a specific package or group of users to a specific server or server group. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

- Do your queries require local processing?
  The following report elements require local processing: crosstabs and charts, master relationships, unions or joins, multiple fact queries, bursting, and non-vendor specific functions. Local processing requires the IBM Cognos server to compute operations on the result set returned by the database, which can impact the SQL execution time.

- Does your environment use a Custom Authentication Provider?
  Using a Custom Authentication Provider could cause a memory leak if the code is not destroying objects correctly.

- Have you reviewed the logs in the c10_location/logs directory and the audit logs?
  They may help you identify the source of the problem. Monitoring your processes, such as the Java and Business Intelligence bus processes could also identify excessive memory use.

- Is your environment tuned correctly?
  For more information, see the Performance Tuning Settings for IBM Cognos 8 Business Intelligence and the IBM Cognos 8 Business Intelligence Performance Tuning Cheat Sheet documents.

- Have you recently upgraded?
  Ensure that any tuning settings that were applied to your previous installation are applied to the new environment. Ensure that your models have been verified, upgraded, and republished. Verify that the IBM Cognos Framework Manager governor that allows enhanced model portability at runtime is not enabled. Depending on your upgrade method, you may also need to open and save the reports again after upgrading.

The following tips may help you improve report performance.

- Change the order in which items are queried from the database. For more information, see “Working with Relational Queries” on page 236 or “Working with Dimensional Queries” on page 283.

- Create sections without creating master detail relationships. For more information, see “Divide Data into Sections” on page 218.

- Share queries between lists and repeaters.
  You can share an identical query between data containers that use it. To share a query, data containers must be lists, repeaters, or repeater tables and must use the same grouping structure and list of properties. The data containers cannot be part of a master detail relationship. Sharing queries improves performance by minimizing the number of queries executed against the database.
  To share a query, set the Share Result Set property for the data container to Yes.

- Convert queries to SQL. For more information, see “Convert a Query to SQL” on page 247 for relational reporting or “Convert a Query to SQL” on page 297 for dimensional reporting.
- Avoid using functions with limited support, as indicated by the quality of service indicators. For more information, see "Insert a Query Calculation" on page 250 for relational reporting or "Quality of Service Indicators" on page 537 for dimensional reporting.

- Be aware of the limitations of creating expressions with SAP BW data sources. For more information, see "Using Microsoft Excel Functions with SSAS 2005 Data Sources" on page 539.

- Use Select & Search prompts instead of value prompts if your list of prompts is long. For more information, see "Build Your Own Prompt and Prompt Page" on page 315.

- Provide your own prompt values in a prompt to avoid accessing the database. For more information, see "Specify Prompt Values" on page 324.

- Suppress null cells using filters. For more information, see "Suppress Null Cells Using Filters" on page 391.

- Set page breaks without creating master detail relationships. For more information, see "Create a Page Break or Page Set" on page 437.

- Do not modify IBM Cognos Analysis Studio set definitions. For more information, see "Analysis Studio Query Specification" on page 454.

- Do not put filters on non-identifiers.

- Avoid combining large hierarchy levels and sets in a drill-through report in a way that creates large queries. For more information, see "Recommendation - Drilling Down in Very Large Data Sources" on page 310.

- Use database functions when possible. For more information, see Appendix F, "Using the expression editor," on page 555.

- Use fixed width objects in PDF reports by setting their Size & Overflow properties. For more information, see "Set List Properties" on page 63, "Set Crosstab Properties" on page 69, "Customizing Chart Properties" on page 104, or "Set Map Properties" on page 163.

The following are Proven Practices (www.ibm.com) documents that may help you improve your report performance.

- Performance Tuning Settings for IBM Cognos 8 Business Intelligence
- IBM Cognos 8 Business Intelligence Performance Tuning Cheat Sheet
- Writing Efficient OLAP Queries
- Cognos 8 Business Intelligence (BI) on IBM AIX® best practices
- IBM Cognos ReportNet and Java Heap

The IBM Cognos Business Intelligence Installation and Configuration Guide also includes a section on performance maintenance.

**Summaries in a report do not correspond to the visible members**

If a crosstab or chart created in IBM Cognos Report Studio using a dimensional data source has a context-dependent set function such as filter or topCount on an edge, summaries do not correspond to the visible members.

This problem occurs because a summary that has the Use Set Aggregation set to Yes, which produces an expression that contains the within set clause, uses a set that is dependent on the members that it intersects with on the opposite edge. For example, the following crosstab has the top three products returned as columns. The expression used to generate the columns is
topCount ([Product],3,[Return quantity])

where [Product] is the level.

<table>
<thead>
<tr>
<th>Return quantity</th>
<th>Bug Shield Lotion</th>
<th>Bug Shield Extreme</th>
<th>Sun Shelter 30</th>
<th>Total</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Americas</td>
<td>25,219</td>
<td>19,870</td>
<td>13,814</td>
<td>62,392</td>
<td>17,303</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>22,822</td>
<td>19,171</td>
<td>6,389</td>
<td>54,758</td>
<td>12,765</td>
</tr>
<tr>
<td>Northern Europe</td>
<td>8,325</td>
<td>14,634</td>
<td>4,065</td>
<td>32,936</td>
<td>8,325</td>
</tr>
<tr>
<td>Central Europe</td>
<td>17,627</td>
<td>13,854</td>
<td>14,089</td>
<td>45,570</td>
<td>13,854</td>
</tr>
<tr>
<td>Southern Europe</td>
<td>7,196</td>
<td>4,726</td>
<td>5,401</td>
<td>20,220</td>
<td>5,790</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>81,189</strong></td>
<td><strong>72,255</strong></td>
<td><strong>43,758</strong></td>
<td><strong>215,876</strong></td>
<td><strong>58,037</strong></td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
<td><strong>7,196</strong></td>
<td><strong>4,726</strong></td>
<td><strong>4,065</strong></td>
<td><strong>20,220</strong></td>
<td><strong>5,790</strong></td>
</tr>
</tbody>
</table>

*Figure 60. A crosstab showing that the summary values for all rows do not correspond to the visible members.*

The summary values for **Total** and **Minimum** for all rows except **Central Europe** do not correspond to the member values in the crosstab. This means that the top three products returned in all regions except for Central Europe are not Bug Shield Lotion, Bug Shield Extreme, and Sun Shelter 30. Note that the summary values for **Total** and **Minimum** for all columns do correspond to the visible member values. That is because those summary values represent the total and minimum quantities returned for those three products in each region.

You can see what the top three products returned in each region are by dragging the columns to the right of the rows.
To obtain summary values that reflect the visible members, modify the expression of the data item containing the context-dependent set function so that it includes a tuple that is locked to the default member of every hierarchy that appears on the opposite edge. For this example, modify the expression to the following:

\[
\text{topCount ([Product],3,tuple([Return quantity], defaultMember([Retailer site])))}
\]

where [Product] is the level and [Retailer site] is the hierarchy.

When you run the report, all summary values reflect the visible members in the crosstab.

Figure 61. A single-edge crosstab showing the Total Returned Products and Minimum Returned Products for each region
Cannot Find the Database in the Content Store (Error QE-DEF-0288)

You cannot retrieve data from the selected database when running a report from IBM Cognos Query Studio, IBM Cognos Connection, or Report Studio.

The following error message appears:

QE-DEF-0288 Unable to find the database...

If this error does not occur when you are logged on as an administrator, then to solve the problem, ensure that the user has permissions to the signon embedded. If this error always occurs, the data source has not been created. Create the data source with the name mentioned in the error message.

Parse Errors When Opening or Running an Upgraded Report

Earlier versions of ReportNet and IBM Cognos Business Intelligence included the cast_Date function for reports that run on an Oracle database. This function does not exist for Oracle in IBM Cognos 8.1.2 MR1 and later versions. If a report that uses an Oracle database includes the cast_Date function, parse errors will be received when you try to open or run the report.

Overflow Error Occurs When a Value in a Crosstab Is More Than 19 Characters

In a crosstab report, values support a maximum of 19 characters, including the decimal point. If a value exceeds 19 digits, an overflow error occurs. By default, the decimal precision is set to 7 digits, which restricts the number of integers to 11 digits.

To use more than 11 integers, you must edit the qfs_config.xml file in the c10_location\configuration directory. For more information, see the section about reducing decimal precision in the IBM Cognos Business Intelligence Administration and Security Guide.

The ORA-00907 Error Appears When Running a Report

When using an Oracle 9.2 data source, under certain circumstances, multiple or nested join operations may fail and produce the following error.

Figure 62. A crosstab showing the return quantity of products in different regions of the world
ORA-00907: missing right parenthesis

A query that uses both a left outer join and an ON clause condition returns zero values instead of null values.

**A Report or Analysis Does Not Run Because of Missing Items**

You attempt to run a report or analysis and a message indicates that one or more items are missing or changed. Each missing item is listed by its MUN (member unique name). The MUN includes the complete path within the hierarchy for the item. When you place your cursor on an item in the **Source** tab, the MUN for that item is displayed in a tooltip. This situation may occur if members have been removed from or changed in the data source. It may also occur when you attempt to run a report that uses items to which you do not have access. For example, an administrator may create an analysis that includes items that you do not have the correct permission to access.

The solution is to find a suitable replacement in the **Source** tab, and drag it to the **New Item** column. The report or analysis will then run.

**Cannot View Burst Report**

When you burst a report, each burst output is sent to the associated list of recipients.

If a list of recipients contains invalid entries, the following occurs:

- The burst output is not saved to IBM Cognos Content Manager.
  Consequently, you cannot view the burst output in IBM Cognos Connection.
- If you choose to send the output by email, only valid recipients will receive an email. Although the output is sent as an attachment if you select the **Attach the report** check box, no link is generated if you select the **Include a link to the report** check box.
- The following error message appears in the run history for the report, where parameter 1 is the burst key, parameter 2 is the list of recipients, and parameter 3 contains the error messages returned by Content Manager:
  
  An error occurred while saving the output for the burst instance <param type="string" index="1"/> with the recipients (<param type="string" index="2"/>). Here are the details: <param type="string" index="3"/>

  **Note:** The list of recipients includes both the valid and invalid recipients.

For example, a report is set up to burst on Country or Region, and the recipients are managers. Running the report produces the following countries and regions and recipients:

- Canada: John, Mary
- US: Peter, Frank
- France: Danielle, Maryse

Frank is an invalid recipient. The burst outputs for Canada and France are saved to Content Manager, but not the U.S. output. If you choose to send an e-mail to each recipient and you selected the **Include a link to the report** check box, the e-mail to Peter will not contain a link to the output for US. The error message that is generated will contain Peter and Frank as values for parameter 2 with no indication as to which is invalid.
Procedure
1. View the error message in the run history for the report.
2. From the list of recipients, determine which recipients are invalid.
   You may need to consult with your administrator to find out which recipients are invalid.
3. Correct or remove the invalid recipients.
   Correcting or removing invalid recipients will depend on how the list of recipients was defined, such as through a calculated field or a burst table.
4. Run the report again.

A report upgraded from ReportNet does not retain its original look

When you upgrade a report to IBM Cognos Business Intelligence, a new style sheet is applied that changes the look of the report.

To preserve the formatting that was used in the original report, you can select a different style sheet. This retains the original look of the report and specifies that any new items added to the report, such as list columns or crosstab levels, have the original formatting applied to them.

Procedure
2. Click Report styles and select 1.x styles.

Measure Format Disappears in SSAS 2005

Microsoft SQL Server 2005 Analysis Services (SSAS) does not propagate formatting through calculations. IBM Cognos compensates for this whenever possible, but cannot guarantee to do so in all cases. As a result, if you are working with a Microsoft SSAS cube, any calculation (other than a non-count summary) that is based on or intersects with a formatted measure, such as a currency, may lose the measure format. This may also happen if you use a detail filter or context filter (slicer).

For example, a crosstab includes members on one edge and a measure with formatting, such as a currency symbol and decimal places, applied on the other edge. When you run the report, you see the formatting for each cell. However, if you add a detail filter, such as measure > 1 and run the report, all the formatting disappears.

Additionally, the fine details of the MDX generated by IBM Cognos Business Intelligence can change from release to release. As the SSAS behavior depends on the MDX generated, the loss of formatting in reports might not occur in a future release.

To avoid this problem, specify explicit formatting for the affected row, column, or cell.

Drill-through Links are Not Active in the Safari Browser

When viewing a PDF report in the Macintosh Safari browser, you cannot open hyperlinks. This is because the Macintosh Safari browser does not have the necessary Adobe Acrobat plug-in.
To avoid this problem, use the HTML format when creating drill-through reports that may be viewed in Safari.

**Data Does Not Appear in a Target Report or the Wrong Data Appears**

If no data appears when you drill through to a target report or if the wrong data appears, the problem might be data source conformance. The business keys might be different or might be mismatched.

For example, the business key for Camping Equipment might be 100 in the data source for the source report and 1 in the data source for the target report, in which case no data appears in the target report. Another example might be that the business key for Camping Equipment is 100 in the data source for the source report but, in the data source for the target report, 100 is the business key for Golf Equipment, in which case the wrong data appears in the target report.

To solve the problem, ensure that business keys have the same value in both data sources. If there are cases where data does not appear to match, contact your database administrator or data modeler.

For more information about data source conformance, search for "conformed dimensions" and "business keys" in the *IBM Cognos Transformer User Guide*.

You might also want to see "Unexpected or Empty Results When Drilling Through" on page 503.

**Related concepts**:

"Conformed Dimensions" on page 420
If you work with more than one dimensional data source, you may notice that some dimensions are structured the same, and some are not.

"Business Keys" on page 421
When drill-through access is defined from a member to a relational value, the business key of the member is passed by default.

**A Running Total in Grouped Reports Gives Unexpected Results**

You have a running total calculation in a grouped report that returns unexpected values.

Because tabulation of the running total calculation depends on the order in which the grouping is executed, you must ensure that the grouped totals are tabulated before applying the running total.

To ensure that the grouping is executed in correct order, define a running total calculation as a freestanding calculation outside the query subject in IBM Cognos Framework Manager, and ensure that the Regular Aggregate property is set to Automatic.

This may also be an issue with other running, moving, and ranking aggregations.

**PCA-ERR-0057 Recursive Evaluation Error**

You run a report and encounter the following error.
PCA-ERR-0057 Recursive evaluation has exceeded limit. Calculated member trace:
COG_QQP_USR_Aggregate(Retailer Type): COG_QQP_INT_m2: COG_QQP_INT_m1:
COG_QQP_USR_Aggregate(Retailer Type): COG_QQP_INT_m2: COG_QQP_INT_m1:
COG_QQP_USR_Aggregate(Retailer Type): COG_QQP_INT_m2: COG_QQP_INT_m1:
COG_QQP_USR_Aggregate(Retailer Type): COG_QQP_INT_m2: COG_QQP_INT_m1

You may encounter this error when two or more data items form a recursive evaluation. For example, in the above error, the calculation of Aggregate(Retailer Type) is dependent on a column expression while at the same time the column expression is dependent on Aggregate(Retailer Type). Therefore, the cyclic relationship cannot be resolved.

To avoid this problem, ensure that calculations do not have cyclic relationships.

**Arithmetic Overflow Error When Running a Report in PDF Format**

If you use a Microsoft SQL Server 2005 data source and your report includes aggregations, you may encounter the following error when you run your report in PDF format:

RQP-DEF-0177 An error occurred while performing operation 'sqlOpenResult' status=’-28’. UDA-SQL-0114 The cursor supplied to the operation “sqlOpenResult” is inactive. UDA-SQL-0564 [Microsoft OLE DB Provider for SQL Server] Arithmetic overflow error converting expression to data type int. (SQLSTATE=22003, SQLERRORCODE=8115)

This error occurs because the action is performed in the database, and the database data type is too small.

This error did not occur in IBM Cognos Business Intelligence version 8.3 or earlier because aggregation was processed locally, by the Business Intelligence server. In version 8.4 or later, aggregation is processed at the database level.

To avoid this problem, increase the size of the database data type.

RQP-DEF-0177 An error occurred while performing operation 'sqlPrepareWithOptions' status=’-69’ UDA-SQL-0043 Error

You cannot run a report in IBM Cognos Report Studio or IBM Cognos Query Studio, and the following error messages appear.

RQP-DEF-0177 An error occurred while performing operation 'sqlPrepareWithOptions' status=’-69’ UDA-SQL-0043 The underlying database detected an error during processing the SQL request.[NCR][ODBC Teradata Driver][Teradata Database] Partial string matching requires character operands

These error messages do not indicate an IBM Cognos Application Firewall problem.

There is a problem with your data source not converting numeric data items. Ask your administrator to consult the topic Enable Conversion of Numeric Search Keys to Strings in Queries in the IBM Cognos Business Intelligence Administration and Security Guide.
Unable to View Active Reports in Mozilla Firefox

When you try to view an active report in the Mozilla Firefox browser, you may see a blank page or a message such as the one below appears.

Your report is ready and will download to your Web browser in a few moments.

However, the report never appears. This problem occurs in the following scenarios:
• You try to open the saved output of an active report in IBM Cognos Connection.
• You try to run an active report from within IBM Cognos Report Studio.

The issue is with the UnMHT add-on, and was reported on the UnMHT Warehouse Web site.

A solution was implemented for each scenario above to work around the issue when using the Mozilla Firefox browser.
• When you try to view an active report in IBM Cognos Connection, instead of opening directly, the download dialog box appears, prompting you to open or save the report.
• In Report Studio, the default action is Download Active Report when clicking the run report button.

Note that if you click the drop-down arrow and choose Run Active Report, the report will not render.

Problems with viewing saved Active Report output in Mozilla Firefox 8 or later versions

When you try to view saved Active Report output in Mozilla Firefox 8 or later, you receive errors. The problem also occurs when you try to insert a saved active report in a workspace in IBM Cognos Workspace.

No problems occur when you use an earlier version of Firefox. In Firefox 8 and above, you see the following errors:

Error: ReferenceError: HTMLIsIndexElement is not defined

Error: TypeError: _IS1 is undefined

The problem occurs because HTMLIsIndexElement was deprecated as of Firefox 8. To resolve the problem, rerun the active report and create a new saved output version using IBM Cognos Business Intelligence 10.2.1 or later.

Tip: If you must rerun many active reports, you can create a job in IBM Cognos Connection to run multiple reports. For more information, see the IBM Cognos Connection User Guide.

Problems with large SAP BW queries

When you are working with an SAP BW data source, if your report includes a data set greater than 1,000,000 cells, you might encounter an XQE error.

To avoid this error, set the Processing property for the query to Limited Local. This option allows the report to process locally.

**Master Detail or Burst Reports with Charts or Crosstabs May Result in Denial of Service**

When running a master detail or burst report that includes a chart or crosstab, disk space exhaustion may cause the report or other requests to fail. A large set of burst keys or master rows may produce one or more charts per detail, resulting in many master detail executions. This may cause the temp folder to accumulate many gigabytes of temporary files containing data required for successful chart rendering.

To avoid this issue, we recommend that you test large master detail or burst reports that include charts or crosstabs to determine the potential peak disk requirements for the report.

**PCA-ERR-0087 error when you run a large report**

You run a large report and you receive a PCA-ERR-0087 error that indicates that the report exceeds the maximum number of tuples allowed.

PCA-ERR-0087 The "crossJoinSet" operator is not applicable. The limit on the number of tuples per edge has been exceeded (<value>).

This error appears when there is an item in a query that has no relationship to the rest of the data or does not make sense in the query.

To resolve the problem, review the query for unrelated items. If an item is found, redesign the query or apply a filter on the item to reduce the number of results returned.

If you still receive the error, ask your administrator to modify the maximum number of tuples setting in IBM Cognos Business Intelligence. The setting is an XML attribute named maxTuplesPerEdge. For more information, see the IBM Cognos Business Intelligence Administration and Security Guide.

**Differences in the appearance of charts that are run in different formats or on different operating systems**

IBM Cognos Report Studio charts can appear differently when reports are run in different formats or on different operating systems. For example, donut and pie charts can appear smaller in reports that are run on the Linux on System z® operating system compared to the Windows operating system.

The differences in appearance occur only with Report Studio default charts. Report Studio legacy charts are not affected. When default charts are rendered in HTML or PDF, the following font-related problems might occur.

- The font that is used in HTML output is different than the font used in PDF output.
- Some chart items appear misaligned. For example, a pie chart might show smaller pies in PDF output than in HTML output.
- The font that is used in the chart is different from the expected font.
- The font that is used in the chart is different when you run the chart in different operating systems.
There are two possible causes to the font-related problems.

- The Java Runtime Environment (JRE) used by IBM Cognos Business Intelligence did not find the font that is specified in the chart, and substituted the font with a different font.

- An unknown default font is specified.

To resolve the problem, ask your administrator to configure JRE to find the installed fonts on the Cognos BI server. There are two ways that you can configure JRE to find the installed fonts:

- Copy the installed fonts to the `${jre}/lib/fonts` folder.
  
  For example, to use the Cognos BI default font Andale WT, copy `Andalewt.ttf` from the `c10_location/bin/fonts` folder to `${jre}/lib/fonts`.

- Configure the JRE font search path to point to the location where the fonts are installed.

  On AIX, set the shell environment variable `JAVA_FONTS=<list of directories>`.

  **Tip:** Setting this variable is equivalent to setting the properties `java.awt.fonts` and `sun.java2d.fontpath`.

The default fonts configuration in your Cognos BI server default style sheet can also cause font-related problems, particularly when the operating system is UNIX. On UNIX systems, fonts must be purchased and installed. To ensure that a specific font is used, perform one of the following tasks:

- Ask your administrator to update the Cognos BI server default style sheet to reflect the fonts that are installed on the server.

- Update the chart fonts that are used in your report to use that font, and do not rely on any default fonts.

  **Tip:** For information about changing the default font for reports, see the technote How to change the default font for reports in Cognos 8 BI in the IBM Cognos Customer Center (http://www.ibm.com/software/data/cognos/customercenter/).

### Out-of-memory errors with reports that are run in interactive HTML format

Out-of-memory errors occur when you run a report in interactive HTML format. The errors do not occur when you run the report in other formats, such as PDF or saved HTML.

Out-of-memory errors occur in reports that contain many objects that require memory intensive processing, such as crosstabs and charts that are linked together with master-detail relationships. Running reports in interactive HTML can consume more memory than running reports in other formats. When a report is run in interactive HTML, data sets are kept in memory for all pages in the report. For report formats like PDF, data sets are released from memory after the report is rendered.

Under certain conditions, you can exceed the memory limitations of a 32-bit configuration of the Cognos BI server report server component when you interact with an interactive HTML report. For example, scrolling through a report page by page or drilling up or drilling down can consume more memory and processing capacity in interactive HTML than in other output formats.
Charts in PDF Output Show Unexpected Results

Charts, when viewed in PDF output, have different levels of interaction support, depending on the version of Adobe Acrobat Reader and the style of chart element.

Adobe Reader version 5 does not support tooltips. Drill up and down and Go to links have limited support, due to technical limitations. Only rectangular areas in charts, such as bars, columns, horizontal labels on the axis, or legend labels can be enabled for drill or Go to interaction. Non-rectangular areas, such as pie slices in pie charts, cannot be enabled for drill or Go to interactions.

Adobe Reader version 6 and 7 supports tooltips, drill up and down, and Go to links for all chart types. When chart elements overlap or are separated by only a few pixels, the interactive region may be smaller than the area shown.

Result set of a multi-fact query contains blanks

The result set of a multi-fact query contains blanks in some columns, even though there is a conformed dimension that should join the records from both fact tables.

When IBM Cognos Business Intelligence processes a multi-fact query, it breaks it up into two single fact queries (each with items from a single star in the star schema), then stitches the results of both of those queries together to create a single result set. You can see the two queries and stitching in the native SQL generated for the multi-fact query in Report Studio.

Stitching these queries together is not the same as joining tables in the model. Stitching the two sides together requires a stitch key that exists in each single fact query as a unique value for the row. The rows from the two single fact queries are then matched 1:0, based on the stitch key values. The stitching must be 1:0. Otherwise, you can get double counting of measure data, or exclude rows that should be included. The stitching also means that it is possible for a row from one of the single fact queries to not have a corresponding row from the other single fact query, leading to blanks in the result set.

When the fact tables contain non-conformed dimensions between them, there is not a suitable stitch key among the query items selected. As a result, IBM Cognos BI creates one using RSUM, and joins the results of the two queries. For more information, see the topic Multiple-fact, Multiple-grain Query on Non-Conformed Dimensions in Guidelines for Modeling Metadata Guide.

When there is a conformed dimension, it is still possible to not have a suitable stitch key, depending on what items are used in the query and other model design factors. In such cases, IBM Cognos BI uses the conformed dimensions and the row numbers from the results of each query to create a stitch key. If one single fact query returns more rows than the other, there are blanks in the result.

To resolve the problem, it might be possible to avoid splitting the query by changing the model. For more information, see Resolving Queries That Should Not Have Been Split in the Framework Manager User Guide. If you cannot avoid splitting the query and the report needs these two result sets to be joined differently, create two separate, single-fact queries in Report Studio, create a join between them, and define the join links and cardinality as needed.

Cognos Statistics object is not displayed in a report

An IBM Cognos Statistics object is not displayed in a report.
Each removed statistical object is replaced with an image in the report:

![Image that replaces statistical objects in upgraded reports](image)

A warning also is displayed in the Upgrade Information window for each statistical object that is replaced with the image.

Beginning with IBM Cognos Business Intelligence version 10.2.1, IBM Cognos Statistics is no longer available.

To ensure that reports that were created in previous releases and that contain statistical objects run, statistical objects are removed when the reports are upgraded.

**Tip:** Queries, and their data items, that are associated to statistical objects are not removed from upgraded reports.

You can use IBM SPSS Statistics to perform statistical reporting and analysis.

---

**Problems When Drilling Through**

The topics in this section document problems you may encounter when drilling through reports.

**Cannot Drill Through Between PowerCubes Because MUNs Do Not Match**

We recommend that business keys be unique throughout the dimension for PowerCubes. These keys are used as the source value for levels in a hierarchy of a dimension. If the values are not unique throughout the dimension, the corresponding Category Code values may be generated with tildes.

For example, if a category for the Product Line level has a source value of 101 and a category in the Product Type level has a source value of 101, the Category Code value for the Product Type level is automatically generated with a unique value such as 101–245. The Category Code values are used in the Member Unique Name (MUN) for each member, for example, [Sales and Marketing].[Products].[Products].[Product type]->:[PC].[@MEMBER].[101–245].

Because these values are generated automatically, they cannot be guaranteed from one cube build to the next or in a build for another cube with the same dimension structure using the same source values. Therefore, drilling from one PowerCube to another on what appears to be the same member might not work since the MUNs might not match.

If the MUNs do not match, consult the cube modellers to see if the business keys can be made unique throughout the dimension. If this is not likely, or might take some time to resolve, you can use calculations to pass the source value from one PowerCube to another for drill-through.
For more information about drill-through access, see the Drill-Through Access chapter in the IBM Cognos Business Intelligence Administration and Security Guide.

Procedure
1. In the target report, create a filter with the following syntax: `filter([Hierarchy or Level], roleValue('_businessKey', [Hierarchy or Level]) = ?Parameter?)`
   For example: `filter([Sales Cube].[Products].[Products].[Product type], roleValue('_businessKey',[Sales Cube].[Products].[Products].[Product type]) = ?Prod Type?)`

2. In the source report, create a Query Calculation which is used to pass the business key (source value) to the target report by mapping it to the target parameter in the drill-through definition. Use the following syntax: `roleValue('_businessKey', [Hierarchy or Level])`
   For example:
   ```
   roleValue('_businessKey',
   [sales_and_marketing].[Products].[Products].[Product type])
   ```

Unexpected or Empty Results When Drilling Through

When you drill from a source report to a target report, there might be no data returned. This might be the correct result if there is no data that corresponds to the drill-through selections or if you do not have permission to view the data.

In other cases, if no data or the wrong data appears, the source item might not be mapped to the target correctly or the values in the data sources might not be conformed (the values do not match in both data sources).

If you have the necessary permissions, you can debug drill-through definitions by using the drill-through assistant from the Go To page (right-click the selection in the source report and select Go To). You can view the passed source values and the mapping to the target report parameters. You can use this tool for both authored and package drill-through definitions.

You might be able to correct the problem by modifying the parameter mapping in the drill-through definition. For example, when you drill from a cube to a relational data source, sometimes no data is returned or the wrong data is returned because the business key values in the two data sources do not match. You can change the drill-through definition to pass the caption of the IBM Cognos PowerCube member instead of the business key, but you must also change the target report to filter on the corresponding string value and not the business key value.

However, it is best to ensure the data sources are conformed. In this example, the business keys in the cube should match the business keys in the relational source. Filtering on a key is more efficient than filtering on a larger string that may or may not be indexed in the database.

For more information on data source conformance, search for "conformed dimensions" and "business keys" in the IBM Cognos Business Intelligence Transformer User Guide.

Procedure
1. Ensure that the target report filters on a string value that matches the caption being passed from the PowerCube.
2. Edit the drill-through definition as follows:
   - If the drill-through definition was created in IBM Cognos Report Studio, open the report, and go to the drill-through definition associated with the drill-through source object. On the parameter mapping page, select Member Caption in the Property to pass column.
   - If the drill-through definition was created in the source package, go to IBM Cognos Connection, Drill-Through Definitions, and open the package drill-through definition. On the Target tab of the drill-through definition, select Member Caption in the Property to pass column for the appropriate parameter.

**Results**

When you drill through, instead of the business key, the caption is passed to the target.

You might also want to see “Data Does Not Appear in a Target Report or the Wrong Data Appears” on page 496. For more information about drill-through access, see the Drill-Through Access chapter in the IBM Cognos Business Intelligence Administration and Security Guide.

**Related concepts:**
- “Conformed Dimensions” on page 420
- “Business Keys” on page 421

**Cannot Drill Through From a Relational Source to a Cube**

By default, you cannot drill through from a relational data source to a cube. This is because a cube expects a Member Unique Name (MUN) as a parameter value and relational sources do not use MUNs.

Members have properties which include a business key and a caption. If either of these match data items within the relational source, drilling through can be performed as long as the cube target report is authored in Report Studio.

If the source data source has a query item, for example display name, that corresponds to a member property in the target cube, for example caption, you must create the parameter on the caption in the target report.

To pass the data item to the cube target, do the following:
- In the cube target report, create a parameter that accepts the caption of the member. This parameter should be created in a Query Calculation object from the Toolbox tab with the following syntax. Type the following
  
  ```
  filter([Hierarchy or Level],caption([Hierarchy of Level]) = ?Parameter?)
  ```
  
  For example:
  ```
  filter([sales_and_marketing].[Products].[Products].[Product line],
  caption([sales_and_marketing].[Products].[Products].[Product line])
  = ?Product Line?)
  ```

  For more information about drill-through access, see the Drill-Through Access chapter in the IBM Cognos Business Intelligence Administration and Security Guide.
Calculations Do Not Appear in the Target Report

If you drill through to PowerPlay Studio from a report in Report Studio, Analysis Studio, or PowerPlay Studio, calculations on the edges in the target report might not appear.

For example, you have a target report with the calculation Personal Accessories+100 as a column in a crosstab report. When you drill through from a source report to the target report, if Personal Accessories is filtered out of the target report (Personal Accessories is not one of the items that is returned on the column edge), then the Personal Accessories+100 calculation does not appear. Personal Accessories has been filtered out of the target report and is not available to fulfill the calculation.

To see the calculations in the target report, ensure the items used in the calculations are returned in the result set (not filtered out). For more information and examples, see Understanding Drill-Through Behavior in IBM Cognos 8 at developerWorks (http://www.ibm.com/developerworks/).

Nested Crosstab Only Filters on Some Items

If you perform a parameter-based drill-through from a source report to a Report Studio target report with two or more dimensions nested on a row or column, you may encounter unexpected results depending on the filters applied to the target report.

For example, a target Report Studio report has the following two filters:

- [sales_and_marketing_mdc].[Order method].[Order method].[Order method type]=?Order Method Type?
- [sales_and_marketing_mdc].[Retailers].[Retailers].[Region]=?Region?

Order method type and Region both have filters, but Product line does not. A drill-through definition mapped to the appropriate parameters, in this case Order method type and Region, is created.

When the source report is run and the intersection of Outdoor protection, Northern Europe, and Telephone is selected to drill through to the target report, the order method type and region display as expected, but all product lines are returned. This is because there are filters on Order method type and Region but not Product line.

For more information and examples, see Understanding Drill-Through Behavior in IBM Cognos 8 at developerWorks (http://www.ibm.com/developerworks/).

Data is Not Filtered in the Target Report After Drill-Through

You drill through to a target report, but no filtering occurs in the target report. For example, you drill through on a crosstab intersection of Camping Equipment and 2010 and expect to see only data for Camping Equipment for 2010 in the target report. Instead you see all products for all years. This occurs because the target report has no filters for the parameters that were passed.

To solve the problem, ensure that the target report has the correct filters. In the above example, the correct filters in the target report are Product line and Year. Alternatively, you can enable Dynamic Drill-Through in a package-based drill-through definition.
Drill-through links in active reports do not work

When viewing an active report that contains drill-through links in Microsoft Internet Explorer 8 and later, the links do not work.

Clicking on a drill-through link produces an error like the following:

The search path "<drill_through_target_search_path>" is invalid. An object may contain invalid syntax, or an unsupported character, or the user account in the namespace may not have sufficient privileges. Check the object to ensure that the target destination location does not contain special characters.

Details

CM-REQ-4069 The property "na" is unknown. Replace it with a valid property.

When clicking a link in an active report, you are attempting to move from a local domain (the active report MHT file on your computer) to the IBM Cognos Business Intelligence server's domain. Internet Explorer views this as a potential risk. To resolve the problem, make the following changes to the security settings in Internet Explorer.

Procedure

1. Click Tools, Internet Options.
2. Click the Security tab.
3. Click Trusted sites and then click Sites.
4. In the Add this website to the zone box, type the IBM Cognos BI server's domain.
5. Click Add and then Close.
6. Click Custom level.
7. In the Scripting section, under Enable XSS filter, click Disable and then click OK twice.
Appendix C. Sample Reports and Packages

Sample reports are included with IBM Cognos Business Intelligence. When installed, you can find them in the Public Folders tab in IBM Cognos Connection.

The Sample Outdoors Company

The Sample Outdoors Company samples illustrate product features and technical and business best practices.

You can also use them for experimenting with and sharing report design techniques and for troubleshooting. As you use the samples, you can connect to features in the product.

For examples related to different kinds of businesses, see the product blueprints on the IBM Cognos Information Centers (http://pic.dhe.ibm.com/infocenter/cogic/v1r0m0/index.jsp).

The Sample Outdoors Company, or GO Sales, or any variation of the Sample Outdoors name, is the name of a fictitious business operation whose sample data is used to develop sample applications for IBM and IBM customers. Its fictitious records include sample data for sales transactions, product distribution, finance, and human resources. Any resemblance to actual names, addresses, contact numbers, or transaction values, is coincidental. Unauthorized duplication is prohibited.

Samples outline

The samples consist of the following:
• Two databases that contain all corporate data, and the related sample models for query and analysis
• Sample cubes and the related models
• A metrics data source including associated metrics and a strategy map for the consolidated company, and a model for Metric extracts.
• Reports, queries, query templates, and workspaces

To run interactive reports, scripts are required. To see all the reports included in the samples packages, copy the files from the samples content installation into deployment folder and then import the deployments into the IBM Cognos Business Intelligence product.

Security

Samples are available to all users.

The Sample Outdoors Group of Companies

To make designing examples faster, especially financial examples, some general information about The Sample Outdoors Company is useful.

To look for samples that use particular product features, see the individual sample descriptions in this section.
Revenue for The Sample Outdoors Company comes from corporate stores and from franchise operations. The revenues are consolidated from the wholly-owned subsidiaries. There are six distinct organizations, each with its own departments and sales branches. Five of these are regionally-based companies.

The sixth company, GO Accessories:
- Has its own collection of products, differentiated from the other GO companies by brand, name, price, color and size.
- Sells from a single branch to all regions and retailers.
- Functions both as an operating company based in Geneva, and as a part owner of the three GO subsidiaries in Europe.

The diagram illustrates the consolidated corporate structure, including the percentage changes in ownership for GO Central Europe, and shows the reporting currency and GL prefix for each subsidiary.

![Diagram of consolidated corporate structure]

Each corporation has the same departmental structure and the same GL structure, shown in the table. Divisions may not report in the same currencies. For example, the Americas subsidiary reports in US dollars, but the Corporate division local currency is Canadian dollars, and the Operations division local currency is pesos.
Table 4. Departmental structure

<table>
<thead>
<tr>
<th>Division (GL)</th>
<th>Department (GL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate (1700)</td>
<td>Sales (1720)</td>
</tr>
<tr>
<td></td>
<td>Marketing (1750)</td>
</tr>
<tr>
<td></td>
<td>IS&amp;T (1760)</td>
</tr>
<tr>
<td></td>
<td>Human Resources (1730)</td>
</tr>
<tr>
<td></td>
<td>Finance (1740)</td>
</tr>
<tr>
<td></td>
<td>Procurement (1710)</td>
</tr>
<tr>
<td>Operations (1800)</td>
<td>Production and Distribution (1820)</td>
</tr>
<tr>
<td></td>
<td>Customer Service (1820)</td>
</tr>
</tbody>
</table>

Each corporation has a complete chart of accounts. Most of the accounts, such as those under non-personnel expenses, are at the department level, and contain only summary amounts. For example, although each marketing department has expenses, the cost is unspecified at the transaction level where marketing promotions occur.

Employees

The Sample Outdoors data contains a full list of employees in all divisions, departments, and locations.

Data is available for reports about bonuses (Global Bonus report) and sales commissions (Sales Commissions for Central Europe report), training (Employee Training by Year report), and performance reviews and employee satisfaction surveys (Employee Satisfaction 2012). If you use Metric Studio, sample metrics for human resources are also available.

In the GO Data Warehouse (analysis) package, groups of measures and the related dimensions are organized into folders. The employees are organized in hierarchies for region and manager, to make different kinds of aggregation easy to report on. Aggregation has been defined for the Employee Position Summary measures, so that Position count and Planned position count aggregate correctly at each level of time: monthly, quarterly, or yearly. For example, see the Planned Headcount report.

The employees are also listed in a sample LDIF file which could be used for any LDAP IBM product authentication including Tivoli®. This authentication directory is necessary for IBM Cognos Planning samples. No other samples depend on security profiles.

Sales and marketing

Data about sales and marketing is available for all of the companies in the Sample Outdoors group.

GO Accessories has richer details to support analysis examples. For example, see the Revenue vs % Profit Margin by Product Brand analysis, based on the Sales and Marketing cube. Marketing and sales campaigns are tied to the Sample Outdoors regional companies.
Overall, the GO companies have experienced solid growth across most product lines (Sales Growth Year Over Year), in all regions (Revenue by GO Subsidiary 2011), because of factors like an increase in repeat business and new or improved products, such as the high margin sunglasses product line. In the product lines sold by the five regional companies (all but GO Accessories) promotions have had mixed success (Promotion Success by Campaign, Bundle and Quarter). If you use Metric Studio, this can also be seen in the sample metrics.

**Customer surveys**

The data also contains information from customer surveys. For example, the product line that includes bug spray, sun screen, and so on has not been successful (Product Satisfaction - Outdoor Protection 2011) and a source of retailer dissatisfaction may be the level of customer service rather than the returns (Customer Returns and Satisfaction). If you use Metric Studio, this information can also be monitored in metrics.

**Sales outlets**

Revenue from the corporate outlets is available at the transaction level. Revenue from the franchise outlets is available at the consolidated level only (Sales and Marketing cube). Metrics about retailers show that the number of new retail outlets has dropped over the time period covered by this data.

GO Accessories sells worldwide, and sells only accessories. Transaction data for GO Accessories is the primary source for analysis of product by brand, color and size. The other five subsidiaries in the group of companies are regional and sell all product lines for retailers in their region. For example, the report Top 10 Retailers in 2011 uses sparklines and list data to review revenues at the retailer level.

**Sample Outdoors database, models, and packages**

The Sample Outdoors Framework Manager models illustrate modeling techniques and support the samples.

The models are based on the GO data warehouse and the GO sales transactional database and are the basis for the sample reports and queries. Each model contains two packages for publishing analysis (dimensional) and query views of the data.

You must have access to Framework Manager, the modeling tool in IBM Cognos Business Intelligence, to look at the sample models. You may also need to set up the sample databases and connections.

**GO Data Warehouse**

The GO Data Warehouse model, `great_outdoors_data_warehouse.cpf`, is based on the database GOSALESDW. It contains data about human resources, sales and marketing, and finance, grouped into business areas. In the Database view, the three business areas are grouped into separate namespaces. The Database view contains a fourth namespace (GO Data) for the common information.

The Database view is very similar to the structure of the underlying database. All tables (database query subjects) are unchanged. This enables IBM Cognos BI to retrieve metadata directly from the package in most cases, instead of using a metadata call to the database. The following changes and additions have been made in the Database view:
• Joins have been added as necessary.
• To allow for aggregation at different levels of granularity, some model query subjects have been created. For example, see the relationships between Time and Sales or Sales fact.
• To allow single joins to be made between the lookup tables and each level in a dimension, lookup tables have been copied. For example, see the Products look up tables.

The Business view contains only model query subjects, with no joins. The following changes and additions have been made in the Business view:
• Calculations were added to the model query subjects. For example, the time dimension contains language calculations.
• Where the database has multiple hierarchies, new dimensions have been created to organize each hierarchy. For example, the employee hierarchies are organized into several categories, such as manager and region.

**The GO Sales transactional database**

The GO Sales model, great_outdoors_sales.cpf, is based on the GOSALES database, which is structured as a transactional database. It contains principally sales data.

The Database view is very similar to the underlying database structure. The following changes and additions have been made in the Database view:
• To make it possible to join the fact tables to the time dimension, model query subjects and multipart joins have been used.
• Other joins have been added as necessary.

The Business view contains only model query subjects, with no joins. The following changes and additions have been made in the Business view:
• Calculations were added to the model query subjects.
• Model query subjects that were created in the Database view to enable joins on the time dimension have been linked as reference shortcuts.
• Where the database has multiple hierarchies, new dimensions have been created to organize each hierarchy.
• Sales Staff is a subset of the slowly changing Employee dimension. There is no unique Employee key in GO Sales, so a filter retrieves the current record only. This model does not use historical data.

**The samples PowerCubes**

The following cubes are delivered with the Sample Outdoors samples in English, French, German, Japanese and Chinese:
• sales_and_marketing.mdc
• employee_expenses.mdc
• go_accessories.mdc
• go_americas.mdc
• go_asia_pacific.mdc
• great_outdoors_sales_en.mdc
• great_outdoors_7.mdc
The samples packages

The Sample Outdoors samples include six packages. A brief description of each available package is provided.

Go Data Warehouse (analysis) is a dimensionally modeled view of the GOSALES database. This package can be used in all studios, including IBM Cognos Analysis Studio. Using this package you can drill up and down.

Go Sales (analysis) is a dimensionally modeled view of the GOSALES database. This package can be used in all studios, including Analysis Studio. Using this package you can drill up and down.

Go Data Warehouse (query) is a non-dimensional view of the GOSALES database. This package can be used in all studios except Analysis Studio, and is useful for reporting when there is no need for drilling up and down.

Go Sales (query) is a non-dimension view of the GOSALES database. This package can be used in all studios except Analysis Studio, and is useful for reporting when there is no need for drilling up and down.

Sales and Marketing (cube) is an OLAP package, based on the sales_and_marketing.mdc cube.

Great Outdoor Sales (cube) is an OLAP package, based on the great_outdoors_sales_en.mdc cube.

Note: The OLAP packages, Great Outdoor Sales (cube) and Sales and Marketing (cube), are not multilingual. The IBM_Cognos_PowerCube.zip archive contains five versions of each package; one in English, French, German, Japanese and Chinese.

Samples in the Sample Outdoors Sales (cube) package

The following report is found in the Sample Outdoors Sales (cube) package.

Consecutive Periods Comparison

This report shows the revenue and gross profit for Camping Equipment and Personal Accessories, as well as quantity for Camping Equipment, in two consecutive periods selected by the user.

Samples in the Sales and Marketing (Cube) Package

The following reports are some of the reports found in the Sales and Marketing (Cube) package.

Actual vs. Planned Revenue

This report shows the actual revenue versus planned revenue by order method and year. This report is also a target for the measure based scope drill-through from other reports in the same package.

This report uses the following features:

- lists
- filters
- multiple prompts
• grouping  
• sorting  
• summarizing  

**Historical Revenue**  
This prompted report shows a 13-month rolling forecast of monthly and year-to-date revenue.  

This report uses the following features:  
• filters  
• cascading prompts  
• combination charts  
• axis titles  

**Revenue by Date Range**  
This report shows revenue for a date range that is specified on a prompt page.  

This report uses the following features:  
• lists  
• crosstabs  
• context filters  
• custom headers and footers  
• multiple prompts  
• calculations  

**Revenue by Product Brand (2011)**  
This report shows the revenue and gross profit by product filtered by the product brand. There is always product turnover, so the report conditionally highlights products that are discontinued.  

This report uses the following features:  
• lists  
• filters  
• prompts  
• combination charts  
• bar charts  
• HTML items  
• grouping  
• sorting  
• axis titles  

**Same Month Prior Year**  
This report shows sales volume by product line in one or more months. The report is filtered by a prompt for month. The report generates totals for the selected months and for the same months in the prior year.  

This report uses the following features:  
• crosstabs
Selected Retailer Country or Region
This report uses the revenue from a selected country or region as a baseline value for a set of countries or regions. A chart shows the difference in revenue for each country and region as it compares to the base country and region. The report is filtered by a prompt for country and region.

This report uses the following features:
- crosstabs
- bar charts
- tables to control where objects appear

Top Retailers by Country or Region
This report shows the top 10 retailers by country or region. It is used as source for drill-through to the Total Revenue by Country or Region report.

This report uses the following features:
- crosstabs
- prompts
- filters
- line charts
- prompt pages
- query calculations
- calculations
- singletons
- bar charts
- custom headers and footers

Samples in the GO Data Warehouse (analysis) Package
The following reports are some of the reports found in the GO Data Warehouse (analysis) package.

Sample reports that were created in Report Studio are located in the Active Report folder and the Report Studio Report Samples folder.

Budget vs. Actual
This report shows three years of data by retailer and retailer site for the camping equipment product line. Each year includes budget and actual data.

This report uses the following features:
- summarizing
- crosstabs
- context filters

Core products results
This active report shows revenue data for the core products Camping Equipment and Golf Equipment.
This list report uses two drop-down list controls to filter data by the following criteria:

- core product
- country or region

### Customer Returns and Satisfaction

This report shows the customer satisfaction survey results for Asia Pacific in 2013. It highlights the customers who are the least satisfied. It also provides information about customers with the highest number of product returns.

This report uses the following features:

- combination charts
- customizing the color and size of a chart
- lists
- formatting a list
- conditional highlighting
- filters
- custom headers and footers
- colors
- lineage
- text items
- grouping
- baselines
- summarizing
- calculations
- drilling through

### Employee Satisfaction 2012

This report shows employee satisfaction survey results by department, compared to targets and industry standards. It also shows employee rankings and terminations.

This report uses the following features:

- crosstabs
- conditional highlighting
- combination charts
- lineage
- text items
- calculations

### Employee Training by Year

This report shows employee training data for the selected year and quarter(s). A bar chart shows training costs by region and a crosstab shows data for the selected quarter(s).

This report uses the following features:

- context filters
- cascading prompts
• bar charts
• customizing the color of a chart
• crosstabs
• calculations

**Eyewear Revenue by Brand and Size**
This report shows a summary of eyewear revenue by brand and compares two prompted retailer sites. The report is filtered by prompts for region, retailer type, and year.

This report uses the following features:
• prompts
• bar charts
• lists
• conditional styles
• calculations
• text items
• custom headers and footers
• combination charts
• axis titles
• crosstabs
• grouping

**Global Bonus Report**
This list report shows employees who received more than $2,500 bonus in a year by region. It is grouped by country or region. It also shows how much the sales target was exceeded for each region.

This report uses the following features:
• lists
• page sets (page breaks by country or region with different sorting and grouping)
• multiple prompts and parameters
• calculations
• filters
• conditional highlighting
• hidden objects
• lineage

**GO Balance Sheet as at Dec 31 2012**
This is the Balance sheet report for Americas where current year data is compared to the previous year data. Analysts can see negative trends under Variance where negative percentages are highlighted.

This report uses the following features:
• conditional highlighting
• padding
• crosstabs
• text items
• context filters

Sample Outdoors Company Balance Sheet as at Dec 31 2012

This report shows a simple balance sheet with assets, liabilities, and equity for 2012 with a 2011 comparative. It uses IBM Cognos Workspace Advanced. The IBM Cognos Business Intelligence Getting Started guide provides a step-by-step example of how to create this report.

This report uses the following feature:
• crosstabs

Manager Profile

This report shows information about managers, including salary, bonuses, and all compensations grouped by year.

This report uses the following features:
• column charts
• lists
• grouping
• summarizing
• custom chart palette
• prompts

New order methods

This active report shows revenue by order method, with focus on the new order methods.

This list report has the following features:
• check box control that allows you to filter data by new order method
• sorting by year or new order method

Planned Headcount

This chart report shows headcount variance compared to the plan for each organization for 2013.

This report uses the following features:
• progressive column charts
• templates
• hidden objects
• custom headers and footers
• lists
• baselines

Positions to Fill

This report shows a list of department names, positions, longest days to fill the positions, and ranking. The report uses a prompt for the year and is a drill-through target for the Recruitment report.

This report uses the following features:
Promotion Plan Revenue
This report shows the planned revenue for all the promotions of a selected campaign. It is a drill-through target for the Top 10 Promotions by Retailer report that is based on the Sales and Marketing (cube) package.

This report uses the following features:
- prompts
- bar charts
- crosstabs
- axis titles

Promotion Success
This report shows the financial results of the company’s promotions. It shows how much of the company’s total revenue is attributable to each promotional campaign.

This report uses the following features:
- prompt pages
- HTML items
- summarizing
- axis titles
- bar charts
- lists
- grouping

Quantity Sold vs. Shipped and Inventory
This report compares the quantity of goods sold and shipped with the opening and closing inventory levels.

This report uses the following features:
- filters
- combination charts
- defined y-axes
- custom headers and footers

Recruitment Report
This report shows a variety of recruitment techniques for certain positions or organizations.

This report uses the following features:
- drilling through
- crosstabs
- prompt pages
- colors
Return Quantity by Order Method
This report shows quantity sold, number of returns, and percentage of returns (with those greater than 5% highlighted) by return reason for each product in the Outdoor Protection product line.

This report uses the following features:
• filters
• lists
• conditional highlighting
• grouping

Returned Items
This report shows the number of returned items by return reason and retailer type. A column chart shows returned items by product line and region for the selected date range.

This report uses the following features:
• date and time prompts
• crosstabs
• bar charts
• drilling down
• text items
• sorting

Returns by Damage, Failed Orders and Complaints in 2012
This report shows quality measures based on product returns.

This report uses the following features:
• pie charts
• crosstabs
• indented text
• singletons
• calculations
• drilling through
• text items
• custom headers and footers

Returns by Failed Orders in 2012
This report shows quality measures based on product returns and focuses on failed orders.

This report uses the following features:
• pie charts
• crosstabs
• indented text
• singletons
• drilling through
• calculations, including the tuple function

**Returns by Order Method**
This report shows product returns and reasons filtered on the order method. The Getting Started guide provides a step-by-step example of how to create this report.

This report uses the following features:
• bar charts
• prompts
• crosstabs
• filters
• custom headers and footers

**Returns by Order Method - Prompted Chart**
This prompted chart report shows product returns and reasons filtered on a prompted order method.

This report uses the following features:
• bar charts
• prompts
• crosstabs
• filters
• custom headers and footers

**Revenue by GO Subsidiary 2011**
This prompted chart report shows 2011 quarterly revenues for each GO subsidiary.

This report uses the following features:
• templates
• colors
• prompts
• hyperlinks
• customizing charts
• singletons
• bar charts
• drilling through
• layout calculations
• pie charts
• calculations
• combination charts
• text items
• blocks
• sorting
Sales Commissions for Central Europe

This report shows an annual summary of sales commissions, revenues, and gross profit for each branch in Central Europe. It also compares actual commission expenses with planned commission expenses.

This report uses the following features:
- prompts
- calculations
- bar charts
- lists
- conditional highlighting
- drilling through
- custom headers and footers
- axis titles

Sales Growth Year Over Year

This report shows annual sales growth in both percentage and dollar amounts.

This report uses the following features:
- bar charts
- lists
- filters
- sorting
- baselines
- axis titles

Sales target by region

This active report shows sales target by region, including the percentage differences between planned and actual revenue.

Succession Report

This report shows the succession data by department and status for percent ready in a column chart. It also contains a detailed crosstab for the managers associated with the possible successors.

This report uses the following features:
- drilling through to the Manager Profile report
- filters
- lists
- grouping

Top 10 Retailers for 2011

This report shows the top 10 retailers for 2011 by revenue and sales target.

This report uses the following features:
- bar charts
- lists
- filters
• multiple queries
• combination charts
• line charts
• notes
• axis titles
• text items
• custom headers and footers

## Samples in the GO Data Warehouse (query) Package

The following reports are some of the reports found in the GO Data Warehouse (query) package.

Sample reports that were created in Report Studio are located in the Active Report folder and the Report Studio Report Samples folder.

### Advertising-cost vs revenue

This active report shows the advertising cost vs revenue by year. Tab controls are used for grouping similar report items.

### Burst Sales Performance Report

This list report shows how to burst a product sales report to a sales manager for Northern Europe sales staff. To successfully burst this report, IBM Cognos Business Intelligence must be configured to use an email server.

This report uses the following features:
• lists
• bursting
• conditional highlighting
• filters
• calculations
• summarizing
• blocks
• custom headers and footers
• sorting
• grouping

### Employee Expenses (report)

This report is used as a data source for the Employee Expenses Power Cube.

This report uses the following feature:
• lists

### Health Insurance

This report is used as a data source for the Employee Expenses Power Cube.

This report uses the following features:
• lists
• filters
Pension Plan
This report is used as a data source for the Employee Expenses Power Cube.

This report uses the following features:
• lists
• filters

Regular Salary
This report is used as a data source for the Employee Expenses Power Cube.

This report uses the following features:
• lists
• filters

TOC Report
This report takes advantage of the bookmark object to allow a user to navigate through this report easily. This report should be run in PDF or saved HTML format. The report contents show a product order table and an expected volume fact table.

This report uses the following features:
• lists
• bookmarks
• background color
• multiple pages
• grouping

Total Revenue by Country or Region
This report summarizes revenue for Retailer Country or Region and Product Line. It is also a drill-through target for the Top Retailers by Country or Region and Revenue by Order Method reports.

This report uses the following features:
• crosstabs
• combination charts
• summarizing
• tables to control where objects appear

Samples in the GO Sales (analysis) Package
The following reports are some of the reports found in the GO Sales (analysis) package.

2011 Quarterly Sales Forecast
This report shows the sales forecast by product line and region for each quarter in 2011.

This report uses the following features:
• lists
• summarizing
• grouping
• sorting

**2011 Sales Summary**

This report summarizes revenue and gross profit for 2011 and shows the top sales representatives by revenue and quantity sold.

This report uses the following features:
• lists
• filters
• combination charts
• axis titles
• custom headers and footers
• conditions

**Samples in the GO Sales (query) Package**

The following reports are some of the reports found in the GO Sales (query) package.

Sample reports that were created in Report Studio are located in the Active Report folder and the Report Studio Report Samples folder.

**Active Report Techniques**

This report demonstrates common active report and dashboard features and functionality.

**Briefing Book**

This report shows a Briefing Book style of report.

This report uses the following features:
• multiple pages
• crosstabs
• multiple queries
• filters
• pie charts
• singletons
• tables of contents
• bookmarks
• PDF options
• horizontal pagination
• sorting
• custom headers and footers
• text items

**Film strip**

This active report shows detailed sales facts in different charts. Deck controls are used for navigation.
**Horizontal Pagination**

This report shows crosstabs rendered across several horizontal pages. The first crosstab shows the fit-to-page behavior while the second crosstab shows the horizontal pagination.

This report uses the following features:
- multiple pages
- horizontal pagination
- crosstabs
- custom headers and footers

**Matrix-chart and graph**

This active report shows a summary of sales facts for each province or state in the Americas. Clicking a category in the map displays the data in a list.

**Order Invoices - Donald Chow, Sales Person**

This report generates invoices for all the sales by Donald Chow.

This report uses the following features:
- lists
- adding list row cells
- calculations
- formatting tables
- calculations
- filters
- grouping
- tables to control where objects appear

**No Data**

Each page of this report presents a different option for dealing with a No Data condition. It also generates invoices of sales for the Order Invoices - Donald Chow, Sales Person report in the GO Sales (query) package.

This report uses the following features:
- crosstabs
- custom headers and footers
- no data
- lists

**PDF Page Properties**

The two pages of this report appear with different Page Orientation (portrait and landscape) when the report is run in PDF format.

This report uses the following features:
- crosstabs
- lists
- page orientation
- PDF options
• custom headers and footers

Product details
This active report shows attributes as color, size, and description for products.

Sales analysis
This active report shows interactions with charts. Clicking a pie series in a chart filters the product line selected.

Sales Dashboard
This active report focuses on sales details by region and product brand. Describes the top performers and the best performance by region.

Singletons on Page Body
This report uses singleton results to display information with no data relationship in the same layout context.

This report uses the following features:
• singletons
• tables
• custom headers and footers

Table of Contents
This report shows two Tables of Contents: one for the main pages and another for the appendices.

This report uses the following features:
• crosstabs
• pie charts
• bookmarks
• tables
• tables of contents
• custom headers and footers
• hyperlinks

Interactive Samples
The following reports are some of the reports found in the Interactive Samples folder.

Bursted Sales Performance Report
This list report shows how to burst a product sales report to a sales manager for Northern Europe sales staff. To successfully burst this report, IBM Cognos Business Intelligence must be configured to use an email server.

This report uses the following features:
• lists
• bursting
• conditional highlighting
Percentage Calculation (by year)
This prompted report shows a percentage calculation based on a particular year.

This report uses the following features:
• lists
• pie charts

Recruitment Report
This report shows a variety of recruitment techniques for certain positions or organizations.

This report uses the following features:
• drilling through
• crosstabs
• prompt pages
• colors
• floating object adjustment
• custom headers and footers

Revenue by GO Subsidiary 2011
This prompted chart report shows 2011 quarterly revenues for each GO subsidiary.

This report uses the following features:
• templates
• colors
• prompts
• hyperlinks
• customizing charts
• singletons
• bar charts
• drilling through
• layout calculations
• pie charts
• calculations
• combination charts
• text items
• blocks
• sorting
Rolling and Moving Averages
This report shows the rolling and moving average count for the return quantity. A prompt uses a macro to provide static choices within a time dimension.

This report uses the following features:
- prompts
- calculations
- crosstabs
- combination charts

Top 10 Retailers for 2011
This report shows the top 10 retailers for 2011 by revenue and sales target.

This report uses the following features:
- bar charts
- lists
- filters
- multiple queries
- combination charts
- line charts
- notes
- axis titles
- text items
- custom headers and footers

Prompt API samples
The following reports are some of the reports found in the Samples_Prompt_API folder.

Related reference:
Appendix H, “Prompt API for IBM Cognos BI,” on page 975
The JavaScript Prompt API provides report authors with a method of customizing prompt interaction in the reports they author.

Clear prompt selections
This report demonstrates using the prompt API to clear selections from all prompts.

Date prompt presets
This report uses the prompt API to provide the user with a set of prompt selection presets based on today's date.

Display all prompt values ignoring user selections
This report uses the prompt API to display all the values in the prompt, whether they are selected or not.

Display user selected prompt values
This report uses the prompt API to display the prompt values selected by the user.
Filter country by letter
This report uses a custom prompt control to provide parameters to filter the report.

Limit date prompt selection by database value
This report demonstrates limiting the selection of a date to a value less than or equal to a latest date value in a query item.

Limit numeric prompt selection by database value
This report demonstrates limiting the selection of a value less than or equal to a largest value in a query item.

Limit time between two dates
This report demonstrates preventing the user from selecting a date range greater than 10 days.

Limit user selection to two items
This report shows how to validate prompt values to stop the user from selecting more than two items.

Pass parameter via hidden prompt
This report demonstrates setting the value of a hidden prompt.

Personal default prompt selections - set selections
This report allows the user to save a set of default prompt selections for use in subsequent reports. The prompt selections are saved to browser cookies for reuse in other reports.

Personal default prompt selections - use selections
This report shows how to use the personal default prompt selections.

Validate prompt values when button pushed
This sample shows how to validate prompt values when a prompt button is clicked.

Validate type-in postal code values
The report uses the prompt API to validate user input as the user types.

Validate type-in product line code
The report uses the prompt API to validate user input as the user types.
Appendix D. Limitations when using dimensional data sources

There are limitations when authoring or running reports against dimensional data sources.

Running a Report Against a Dimensional Data Source

You can cancel a report that is running against Microsoft SQL Server Analysis Services only during the initial portion of its execution. After this time, the report runs to completion.

The same behavior applies to SAP BW data sources.

In IBM Cognos Framework Manager, you can also control the number of levels within a hierarchy from which members, or values, are extracted from the hierarchy to populate a tree prompt.

For SAP BW, you can reduce the number of hierarchy levels to limit the number of nodes by setting the SAP BW variable property `trimHierarchyLevels` to 1. This removes the lowest level from the hierarchy prior to creating the list of nodes.

Units of Measure Notation

When running a report against an SAP BW data source, units of measure are included in the same column as the data values, separated by one space. For example, Celsius and Fahrenheit notations are appended to the end of the value.

If you see an asterisk character (*), one of the following was detected:

• an unknown currency
• a value with an unknown or questionable unit of measure, such as a mixed currency calculation or rollup
  Mixed currency values occur when you calculate values with different currencies.

This behavior occurs when you are using an IBM Cognos cube as a data source.

This behavior also occurs for SAP BW data sources.

Unsupported SAP Variable Properties

The following SAP variable properties are not supported:

• Exclusionary ranges appear as an inclusionary prompt.
• Mandatory not initial appears as a mandatory prompt.

When using Business Explorer (BEx) to define variables in your SAP data source, avoid using exclusionary ranges and the mandatory not initial property.
Limitations When Using Set Expressions in List Reports

In list reports, we recommend that you avoid using set expressions. When in a list, set expressions, such as TopCount, may produce fewer rows than in the corresponding crosstab.

For example, the following list report includes Year in the first column, followed by a column containing an expression that returns the top three months by revenue. Only three rows appear in the report and they correspond to the top three months across all years. If you group by Year, the report still shows only three months. However, the corresponding crosstab report, you see three months for each year.

Figure 65. a list showing the top three revenue months in 2013 and a crosstab showing the top three revenue months for each year

In crosstab reports in IBM Cognos Business Intelligence, set evaluation is always done in the context of what the set is nested under in the report. However, in list reports set evaluation is done independently of the grouping. For example, if the first column contains Country or Region instead of Year, you see the top three months for each country or region (across all years) in both cases.

In the case of different dimensions, you can force context independence by replacing [Revenue] in the topCount expression with tuple ([Revenue], X), where X is the default member of the hierarchy that contains Month.

However, for nested levels in the same hierarchy, there is no such workaround at this time.

Limitations When Using Clauses in Summary Functions

A summary function that uses a for clause may yield unexpected results. These may include error messages, warnings, incorrect numbers, and more or fewer than expected rows, columns, or chart points and lines.
To avoid these problems, ensure that the parameters that follow the \texttt{for} clause adhere to the following constraints:

- Parameters must be simple data item references.
- All data items in the parameter list must appear on every list, crosstab, or chart that uses that summary.
- For any edge used in the \texttt{for} clause, data items listed in the \texttt{for} clause must start with the first data item on that edge.
- Data items must be listed in the order in which they appear on each edge of the report with no gaps.
- In crosstabs and charts, there must be no sibling data items that are considered details. Summaries are normally not considered details.
- Section headers must not be included in the parameter list.

If following these constraints does not resolve the problems and your report uses dimensional data with no detail or summary filters, consider using the \texttt{within set} clause instead of the \texttt{for} clause.

**Limited Support for Relational Functions When Used with OLAP Data Sources**

When working with an OLAP data source, we recommend that you not use relational functions, such as substring and concatenation functions, in a report that also contains a measure with the \texttt{Aggregate Function} property set to \texttt{Calculated} or \texttt{Automatic} in the model. If you do so, you may encounter unexpected results. For example, some summaries are calculated using the \texttt{Minimum} function instead of the aggregate function derived from the individual query items.

In the expression editor, an exclamation mark (!) that precedes a function indicates that the function is not naturally supported for that data source. IBM Cognos Business Intelligence uses a local approximation for that function. Because an approximation is used, performance can be degraded and the results may not be what you expect.

For example, you create an IBM Cognos Query Studio report that contains the data items Product line and Retailer site count. The footer summary is set to \texttt{Calculated}. You then insert a calculated column that returns the first three characters of the Product line item, which uses the relational concatenation function. The footer summary now shows the lowest gross margin value.

<table>
<thead>
<tr>
<th>Product line</th>
<th>First Three Characters</th>
<th>Retailer Site Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Equipment</td>
<td>Cam</td>
<td>632</td>
</tr>
<tr>
<td>Mountaineering Equip</td>
<td>Mou</td>
<td>265</td>
</tr>
<tr>
<td>Personal Accessories</td>
<td>Per</td>
<td>820</td>
</tr>
<tr>
<td>Outdoor Protection</td>
<td>Out</td>
<td>639</td>
</tr>
<tr>
<td>Golf Equipment</td>
<td>Gol</td>
<td>267</td>
</tr>
<tr>
<td>Summary</td>
<td></td>
<td>265</td>
</tr>
</tbody>
</table>
Limitations When Summarizing Measures in DMR Data Sources

There are limitations when summarizing dimensionally-modeled relational (DMR) measures and semi-additive measures in crosstabs using the aggregation function count distinct, median, standard-deviation, or variance. The following limitations can produce empty or error cells when the report is run:

- The aggregation function must apply to all members of a level or all children of a member.
- To use OLAP functions in detail filters that are applied to a dimension that is not in the report, or is at a level below what is being reported, only the functions children, level, members, roleValue, and rootMembers will work.
- You cannot define detail filters that reference one or more measures and are set to After auto aggregation.
- You cannot define context filters that have more than one member from a dimension that does not appear in the report.
- Context filters that have more than one member from a dimension that appears in the report produce errors in all cells that are ancestors of the slicer members.
- If a crosstab has a row that is a set of members from one dimension (dimension A) and another row that is a set of members from another dimension (dimension B), and a context filter containing members from dimension A is defined, error cells are produced in the row that contains members from dimension B.
- If a context filter contains members from a dimension, and a crosstab has a row that is a set of members from a higher level than the slicer members, error cells are produced for that row.
- Error cells are produced when drilling down on a crosstab that has two nested levels.

If you do not consider these limitations in a calculation, the report may return inaccurate results.

If there is a non-measure calculation that returns a constant or contains a summary function, and the calculation has a lower solve order than the measure being aggregated, error cells are returned for the aggregated measure.

In list reports, error cells are produced as a result of these limitations if the list uses an OLAP function other than children, filter, level, members, roleValue, and rootMembers.

Limitations When Filtering Dimensional Data Sources

Avoid combining context filters (slicers) with dimensional constructs that involve members from hierarchies that are used elsewhere in the report. This combination gives results that are often not what you might expect and that may change in a future release.

Dimensional data sources provide implicit rollup at all levels of each dimensional hierarchy. Context filters and slicers with more than one member invalidate any pre-computed rollup of members at hierarchy levels above the level at which the filter applies.
The following types of filtering are safe for dimensional reporting:

- Context filters (slicers) with one or more explicit, non-calculated members per hierarchy, if those hierarchies are not used elsewhere in the report.
- Expressions that use the `filter` function (not in a slicer), comparing measure, tuple, or attribute values.

### Nesting Incomplete Sets

When you nest sets (especially sets of explicit members, and sets that result from the functions such as `filter` and `intersect`), there may be members at one nesting level that have no corresponding members that can appear nested below them. In such cases, filtering results may not be what you expect: the higher-lever members may or may not appear in the result.

While this is most commonly seen within a single hierarchy, it can also happen across different hierarchies and dimensions.

To avoid this problem, ensure that the lower-level sets are constructed to ensure that they are never empty for any of the members in the higher-level set. For example, you could filter only the top level set and nest only the complete set of descendants at the desired levels below.

### Error Characters (--) When Filtering

When you use slicers or context filters with a calculated fact or measure, you may see two dashes (--) as values for some or all of the cells in the report that represent aggregates that are calculated in the database (aggregation is set to `Automatic`). This means that the data source is unable to compute these values.

You can avoid this error by using an explicit rollup rule or aggregation function such as `Total`. However, do this only if you are familiar with the data and absolutely certain that this is the appropriate answer for that report.

### Time Hierarchies

Error characters also appear in summaries if your report includes a time hierarchy and you use two or more members from a different time hierarchy as a context filter. To avoid the error, use only one member as the context filter.

### Filters in Queries and Sub-queries

For dimensional data sources, filters in queries and sub-queries are considered equivalent. The same applies to slicers.

### Limitations When Specifying the Scope of Summary Filters with Dimensional Data Sources

When you specify the scope for a summary filter, consider the following limitations. The scope of a summary filter

- must refer to a grouped data item for list reports. Otherwise, you encounter the following error:

  `OP-ERR-0212 The summary filter scope (Scope1) found in the query (Query1) is not valid. It must refer to a grouped query item.`

- can refer to multiple data items, but it can refer to only one data item per edge of the crosstab. Otherwise, you encounter the following error:
**Limitations When Filtering Data Using an SAP BW Data Source**

If you are working with an SAP BW data source, you must consider additional exceptions when applying filters.

If you apply a filter and a sort to an item that corresponds to the leaf-level of a recursive hierarchy, siblings may appear in the report even though the aggregated values are correct. Siblings are characteristic values with the same parent as the filtered member.

Each level in an SAP BW hierarchy has an item with the same name as the level and a role of _businessKey. Such items are known as level identifiers. The level identifier must be an exact value for the operators =, <, and > to work. For example, for the filter `[Office] > 'Chicago'` to work, the value 'Chicago' must exist in the data source. If you do not know the exact values, you can apply the filter to one of the attribute items associated with the level, such as `[OfficeLongName] > 'C'`. Filters on non-identifiers are possible, but they are slower because SAP BW data sources are optimized for queries based on level identifiers.

When filtering time-related data, only the level identifier items of the time-related characteristics in SAP BW, such as 0CALDAY and 0CALMONTH, should be used for performing anything other than equality filters. All other (attribute) items in these hierarchies are formatted string representations of the characteristic values with which they are associated. These formatted values sort alphanumerically and not chronologically.

**Using Prompt Expressions in Filters**

If you create a filter expression using report item attributes, such as the following, no data is returned.

```sql
[ report item attribute ] = ?prompt?
```

To resolve the problem, do one of the following:

- Fully qualify the item by using items from the source tree instead of data items from the report.
- Use prompt controls in the report instead of hand coded prompts.
Considerations when Creating Calculations

Consider the following information and limitations when creating dimensional calculations.

**Calculation Solve Order**

When calculations in the rows and columns of a report intersect, calculations are performed in the following order: addition or subtraction, multiplication or division, aggregation (rollup), and then the remaining arithmetic functions.

The remaining functions are as follows:
- absolute, round, average, minimum, maximum, medium, count
- percentage, % difference (growth), or % of total
- rank, quartile, quantile, percentile

If both calculations have the same precedence, for example, if they are both business functions, then the row calculation takes precedence.

You can override the order of precedence by changing the solve order property. For more information, see "Resolve Multiple Calculations for Crosstabs and Charts" on page 540.

**Length of Expressions**

When creating layout calculations that use complex expressions, limit the length of the expression to less than 1,000 tokens. A token is a keyword, identifier, constant, or special character symbol that the server identifies when it parses the SQL. For example, the expression 1+1+1+1+1+1+1+1+1 contains approximately 28 tokens, whereas the expression 1+1+1...+1 (400 times) contains more than 1000 tokens.

*Tip:* The Sales Commissions for Central Europe sample report in the GO Data Warehouse (analysis) package includes expressions. For more information about The Sample Outdoors Company samples, see Appendix C, "Sample Reports and Packages," on page 507.

**Quality of Service Indicators**

Not all data sources support functions the same way. The data modeler can set a quality of service indicator on functions to give a visual clue about the behavior of the functions. Report authors can use the quality of service indicators to determine which functions to use in a report. The quality of service indicators are:
- not available (X)  
The function is not available for any data source in the package.
- limited availability (!!)  
The function is not available for some data sources in the package.
- limited support (!)  
The function is available for all data sources in the package but is not naturally supported for that data source. IBM Cognos Business Intelligence uses a local approximation for that function. Because an approximation is used, performance can be poor and the results may not be what you expect.
- unconstrained (check mark)  
The function is available for all data sources.
Using Quotation Marks in Literal Strings

When inserting literal strings in an expression, you must enclose the string in single quotation marks. If the string contains a quotation mark, it must be escaped. For example, if you want to insert the string ab’c, you must type ‘ab’c’.

Limitations of Calculations

You should use only the expressions and functions available in IBM Cognos Report Studio, and follow their syntax.

Minimal checking exists for calculations. If your calculation uses an invalid expression, your report results may contain unexpected values.

In addition, you should define member summaries as follows:

\[ \text{summary\_function (currentMeasure within set set\_reference)} \]

where \( set\_reference \) is a level or set inserted from the Source tab.

Unless otherwise required, \( \text{summary\_function} \) should be the aggregate function. If you use an explicit summary function, you may encounter problems with measures (such as profit margin, distinct count, and so on) that have complex rollup values, and/or scenario or account dimension members that do not roll up.

Know your data, and confirm with the owner of the cube where overriding the automatic aggregation is safe.

Because of the above limitations, summaries of calculations may not provide reliable values. For convenience, you may have to build reports where row summaries and calculated member columns intersect. In such reports, these intersections may contain unexpected values. In contrast, row calculations intersecting with column aggregates using the aggregate function are safe because the calculation is performed on the reliably summarized values.

Units of Measure

When creating calculations in IBM Cognos Report Studio and IBM Cognos Query Studio, you may encounter problems with the units of measure. For example, the calculation Cost*Cost returns the unit of measure * instead of a currency unit of measure. To avoid this problem, change the format of the corresponding column to obtain the desired unit of measure.

Limitation When Using Dimensional Functions with Running and Moving Summaries

IBM Cognos Business Intelligence does not currently support queries that contain both dimensional functions and running summaries or moving summaries. For example, when you drill down in a report, the query uses the dimensional function children, which is not compatible with running and moving summaries. Running and moving summaries are supported for only the overall level of a dimension.

Creating Expressions Using SAP BW Data Sources

You must consider the following when creating expressions using an SAP BW data source, or you may not get the results you expect.
• The case and if/then/else constructs are not supported in filters.
• The query item identifier of the leaf-level of the 0CALDAY characteristic and its presentation hierarchies is of type date. When the values for the query item identifier are presented in IBM Cognos Report Studio, they are formatted as dates. These formatted values should not be used in filter expressions. The correct date constant format for use in expressions is YYYY-MM-DD.
• You can apply a comparison expression with an operator other than equals to a query item that represents a level identifier. However, level identifiers are more efficient for identifying specific values. Range comparisons must be performed on the IBM Cognos application server, which slows down the performance of the report.

Using Microsoft Excel Functions with SSAS 2005 Data Sources
If you are working with a Microsoft SQL Server 2005 Analysis Services (SSAS) data source and you want to use Microsoft Excel VBA functions such as ROUNDDOWN in MDX queries, the following Microsoft Office features must be installed on the SSAS server:
• Microsoft Office Excel
• Microsoft Visual Basic for Applications

To see a list of Excel VBA functions, see the Microsoft Web site http://www.microsoft.com.

If these Microsoft Office features are not installed on the SSAS 2005 server, and you use Excel VBA functions, you encounter an error such as the following: YK-ERR-0008 The data provider returned an error message: "The [Excel].[ROUNDDOWN] function does not exist."

This error occurs because the OLAP server cannot process the function since Excel function libraries are missing.

For more information, see the Microsoft Knowledge Base article # 932214 at http://support.microsoft.com.

Concatenating Strings
When IBM Cognos Business Intelligence concatenates strings locally and if any of the involved strings contain null values, the result of the concatenation is an empty cell or a null value. This occurs because IBM Cognos BI requires that an expression that involves a null value returns a null value. Many databases ignore null strings when they perform concatenations. For example, if you concatenate strings A, B, and C, and if string B is a null value, the database may concatenate only strings A and C.

Intersecting Calculations in Crosstabs and Charts
An intersection point in a crosstab or chart can contain a value that is derived from multiple calculations.

If the query expressions for the row and column that intersect in a crosstab both include calculations, the intersecting value is the result of performing both calculations. The second calculation is performed on the result of the first calculation. If you change the order in which the calculations are performed, the resulting value at the intersection point changes.
The solve order is a positive numeric value that you assign to data items. Negative values are not supported. The data item with the lowest value is calculated first, followed by the next higher value, and so on. The data item with the highest solve order is computed last. If you do not specify a solve order, the default value of 0 is used. In crosstabs, if more than one data item has the same solve order, column items are calculated first and row items are calculated second.

For example, a crosstab contains the columns Revenue, Sales target, and a column that calculates the percentage by which actual revenue exceeds target revenue. The crosstab contains a row for each Year and a summary row that calculates the overall totals for each of the three columns. The cell where the percentage calculation and the overall total calculation intersect contains only one value. By default, IBM Cognos Business Intelligence calculates the column percentage first and the summary row last. The value that appears at the intersection is therefore a sum of the percentages, which is not a meaningful result. The value is more meaningful if the overall percentage is calculated after the actual and sales target revenues are summed. Specifying a solve order for the percentage calculation that is higher than the overall total calculation gives the following result:

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
<th>Sales target</th>
<th>Percentage by which Actual Exceeds Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>$9,143,528,032.72</td>
<td>$8,128,885,300.00</td>
<td>12%</td>
</tr>
<tr>
<td>2011</td>
<td>$1,159,159,590.16</td>
<td>$1,036,923,300.00</td>
<td>12%</td>
</tr>
<tr>
<td>2012</td>
<td>$1,495,891,100.90</td>
<td>$1,332,553,100.00</td>
<td>12%</td>
</tr>
<tr>
<td>2013</td>
<td>$1,117,336,274.07</td>
<td>$1,023,006,640.00</td>
<td>9%</td>
</tr>
<tr>
<td>Total</td>
<td>$4,686,775,768.85</td>
<td>$4,205,368,540.00</td>
<td>11%</td>
</tr>
</tbody>
</table>

Figure 66. Crosstab showing a calculation that has a solve order specified

It is not necessary to specify the solve order for the overall total calculation. Because the default solve order is 0, setting the solve order for the percentage calculation to a value higher than 0 is sufficient.

**Nested Calculations in Crosstabs**

If a crosstab contains nested calculations, the calculations are performed in the following order by default:

1. outermost calculation on the columns
2. innermost calculation on the columns
3. outermost calculation on the rows
4. innermost calculation on the rows

If there are solve order ties, column items are calculated before row items and outer nesting items are calculated before inner nesting items.

**Resolve Multiple Calculations for Crosstabs and Charts**

If an intersection in a crosstab or chart contains a value that is derived from multiple calculations, you can specify the order in which to perform the calculations. The solve order must be a positive numeric value. The calculations are performed from lowest to highest solve order value.

Setting the Rollup Aggregate Function property to Calculated for a data item assigns a solve order that is higher than any other data item in a query unless a
solve order is explicitly specified. If there are multiple data items that have the Rollup Aggregate Function property set to Calculated, the normal rules for solve order apply.

**Procedure**

1. Click the data item for which to specify the solve order.
   A data item can have only one solve order value. Consequently, if a data item is in a query that is used in more than one context and you want the data item to have a different solve order in each context, you must use a separate data item.

   **Tip:** In crosstabs, to specify the solve order for the default measure, click the crosstab and, in the Properties pane, click the select ancestor button and click Crosstab. Or you can click the container selector (three orange dots) of the crosstab to select it.

2. In the Properties pane, set the Solve Order property to an integer value.
   **Tip:** To specify the solve order for the default measure, click Default Measure Solve Order.

**Resolve Query Calculations that Intersect with Calculated Measures Defined in Microsoft Analysis Services**

Microsoft SQL Server Analysis Services (SSAS) data sources may include data source calculations. Because IBM Cognos Business Intelligence cannot predict the presence of a solve order on such calculations, setting the Solve Order property may not resolve the solve order correctly. You can suppress potentially meaningless results when querying dimensional data sources that include data source calculations. In the following example, the columns YTD Change and YTD Growth are data source calculated members and the summary values for these two columns were suppressed.

<table>
<thead>
<tr>
<th>Revenue</th>
<th>Prior YTD</th>
<th>YTD</th>
<th>YTD Change</th>
<th>YTD Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Equipment</td>
<td>290,355,466.50</td>
<td>352,910,329.57</td>
<td>62,554,863.47</td>
<td>21.54%</td>
</tr>
<tr>
<td>Personal Accessories</td>
<td>329,883,402.11</td>
<td>443,693,449.65</td>
<td>113,810,047.74</td>
<td>34.50%</td>
</tr>
<tr>
<td>Outdoor Protection</td>
<td>5,945,944.77</td>
<td>4,471,625.26</td>
<td>-1,471,919.51</td>
<td>-24.77%</td>
</tr>
<tr>
<td>Golf Equipment</td>
<td>138,993,550.79</td>
<td>174,740,819.29</td>
<td>35,747,258.50</td>
<td>25.72%</td>
</tr>
<tr>
<td>Mountaineering Equipment</td>
<td>93,654,629.48</td>
<td>141,520,649.70</td>
<td>47,866,020.22</td>
<td>51.11%</td>
</tr>
<tr>
<td>Total(Product line)</td>
<td>858,830,063.65</td>
<td>1,117,336,274.07</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

*Figure 67. Crosstab showing revenue by product line with a summary row*

When a query-defined calculated member and a data source calculated member intersect, the value is the result of performing both calculations. The second calculation is performed on the result of the first calculation. Whether the final value is meaningful depends on the order in which the calculations are performed. Because IBM Cognos BI cannot determine the presence of these members ahead of time, it cannot automatically determine the correct solve order.

In most cases, SSAS 2005 gives the expected results when no solve order is defined in the cube. Therefore, when using SSAS 2005, you do not need to set either the Solve Order property in the query or the Calculation Intersection property.
Note: Dimensionally-modeled relational (DMR) data sources query one or more underlying relational databases, and therefore cannot contain data source calculations. They are not considered dimensional data sources in the context of this topic.

Procedure

1. Pause the pointer over the query explorer button and click the query that contains the data item for which to suppress values.
2. In the Data Items pane, click the data item.
3. In the Properties pane, set the Calculation Intersection property to Show "Not applicable".

Results

When a query-defined calculated member and a data source calculated member intersect, the resulting value is suppressed. If the intersecting members do not contain data source calculations, the value is not suppressed.

Null (Missing) Values in Calculations and Summaries

The data source that you use determines how null (or missing) values are handled in calculations, summaries, and other expressions.

Null values represent missing data values. For example, a new eyewear product, called Abby, is introduced in 2012. Therefore there are no sales in 2011. If you create a report with the years 2011 and 2012, the values for sales of Abby for the year 2011 are null (missing).

By default, null values appear as blank cells in a crosstab or list report. You can change the default to specify the display of other characters, such as the word null, by changing the data format of the Missing Value Characters property for the cells or report. For more information, see "Set the Default Data Formats" on page 387.

Arithmetic Operations

Null values in arithmetic operations yield different results depending on the type of data source.

In Microsoft SQL Server Analysis Services (SSAS) and Cubing Services data sources, a null value in arithmetic operations is treated as follows:
- In operations such as addition (+), a null value yields the same result as zero (9 + NULL = 9)
- In operations such as multiplication (*), a null value yields a null result (9 * NULL = NULL)

You can accommodate this behavior by using more complex expressions such as the following:
- if ([M] is not NULL, then ([M]) else (0)
- if ([M] is not NULL, then (expression involving M) else NULL)

In relational and other OLAP data sources the result of an operation that includes a null value is always null. For example, 9 + NULL = NULL.
Equality Comparisons

Equality comparisons, such as equal to (=) and not equal to (<>), compare two values and return a Boolean value of true or false.

For all data sources, equality comparisons are treated as tests for missing data. Inequality comparisons are treated as tests for non-missing data.

For example, NULL=NULL is true and NULL=0 is false.

Ordered Comparisons

In ordered comparisons, such as rank and quantile functions, null values are handled as follows.

For relational data sources, the result of the comparison is always false. For example, NULL < 1 is false.

For Microsoft SSAS, Cubing Services, and other OLAP data sources, the null value is treated as zero. For example, NULL < 1 is true.

Boolean Functions

In Boolean operations, such as or, null values are handled as follows.

For relational and dimensionally-modeled relational data sources, the result of a Boolean operation is always null. Boolean functions are used in conditions and the null result is treated as a false condition. For example, NULL OR TRUE = NULL (FALSE)

For Microsoft SSAS, Cubing Services, and other OLAP data sources, the null value is treated as if it is false. For example, NULL OR TRUE = TRUE.

Summary Functions

All summary functions, such as aggregate, average, and maximum, ignore null values. For all functions except count, if all values in the expression are null, the result is null. For count, if all values in the expression are null, the result is zero.

For example, if the value for Quantity for 2013 is 10 and the values for 2011 and 2012 are null, then the average is as follows:

- average ([Quantity] within set set([2012], [2013])) = 10
- average ([Quantity] within set set([2012], [2011])) = NULL

The count is as follows:
- count ([Quantity] within set set([2012], [2013])) = 1
- count ([Quantity] within set set([2012], [2011])) = 0

String Operations

For all data sources, the result of string operations, such as concatenation (||), that include a null value is always null.

For example, 'A' || NULL = NULL and NULL || NULL = NULL.
Dimensional Coercion Rules

IBM Cognos Business Intelligence uses dimensional coercion rules to automatically convert dimensional types to better match other types. These conversions help you build simpler expressions, making them easier to understand. In addition to the implicit rules that IBM Cognos BI provides, you can make explicit data type conversions using various functions, such as `children`.

IBM Cognos BI supports the following types of coercion:
- coercion of an object of one dimensional type into another, such as a level into a member set
- coercion of a dimensional object into a numeric, date, time, or string value, such as a numeric measure into a numeric value

Coercion rules are applied to expressions and data items. In expressions, an operator or function may require operands to be of a particular dimensional type. When an operand is not of the required type, one or more coercion rules may be applied to coerce the operand to the appropriate type. Coercion rules may also be applied to data items to coerce the data item to a collection of members or values.

Function Operands

The following describes how coercion rules are applied to function operands:
- If the operand is of the required type, no coercion is required.
- If the function operand is supposed to be a numeric value, it is coerced into one.
  A coercion rule exists for each type of dimensional object coercion.
- If the function operand is supposed to be a dimensional object and a coercion exists to coerce the operand to the required type, the coercion is applied.
  - If no coercion exists, an error message with the code QE-DEF-0478 appears to indicate that an unsupported conversion of a dimensional object from the source to target type occurred.

Comparison and Other Symmetric Operators

Binary operators that accept operands of more than one type, such as equals (=) and `in`, need both operands to be the same dimensional type.

No coercion is possible between value domains (numeric, date, time, and string) or between members and values. Consequently, if either operand is a value type, both must be in the same value domain and converted explicitly using a function. Otherwise, the query fails.

Members and member set operands are valid only with the operators equals (=), not equals (`<>`), `in`, and `not in`, where the right side of the expression is a member, member set, or prompt. Only the following are valid:
- `[member / member set] = [member]`
- `[member / member set] <> [member]`
- `[member / member set] = ?p?`
- `[member / member set] <> ?p?`
- `[member / member set] in ([member], ...)[member / member set] not in ([member], ...)
- `[member / member set] in ([member set])[member / member set] not in ([member set])[member / member set] in ?p?[member / member set] not in ?p?`
Exceptions

For the left operand, member sets are supported in detail and summary filter expressions, but not in expressions that use the `filter` function. Members are not supported in detail and summary filters, but they may be used in expressions that use the `filter` function.

The `in_range` operator is not supported for members and member sets.

Normal coercion rules apply to these operands.

NULL operands are considered values, rather than members.

Examples

The following examples illustrate how coercion is applied to levels in expressions with operators.

\[
[\text{Sales}].[\text{Products}].[].[\text{Product Line}] = [\text{Sales}].[\text{Products}].[].[\text{Product Line}] ->[\text{Outdoor Equipment}]
\]

The left operand is coerced to the following member set:

\[
\text{members } ([\text{Sales}].[\text{Products}].[],[\text{Product Line}])
\]

The following expressions are invalid:

- \([\text{Sales}].[\text{Products}].[].[\text{Product Line}] = \text{NULL}\)
- \([\text{Sales}].[\text{Products}].[].[\text{Product Line}] + 1\)
- \([\text{Sales}].[\text{Products}].[].[\text{Product Line}] = \text{‘Canada’}\)
- \([\text{Sales}].[\text{Products}].[].[\text{Product Line}] > 2000\)

The following examples illustrate how coercion is applied to members in expressions with operators. In each of the examples below, the left operand is coerced to a value of the current cell at this member:

\[
(\text{tuple } ([\text{Sales}].[\text{Products}].[],[\text{Product Line}]->[\text{Outdoor Equipment}]))
\]
- \([\text{Sales}].[\text{Products}].[].[\text{Product Line}]->[\text{Outdoor Equipment}] = \text{NULL}\)
- \([\text{Sales}].[\text{Products}].[].[\text{Product Line}]->[\text{Outdoor Equipment}] + 1\)
- \([\text{Sales}].[\text{Products}].[].[\text{Product Line}]->[\text{Outdoor Equipment}] > 2000\)
- \([\text{Sales}].[\text{Products}].[].[\text{Product Line}]->[\text{Outdoor Equipment}] + \text{‘Canada’}\)

Coercion Rules for Dimensional Objects

The following table describes coercion rules that are applied when coercing an object of one dimensional type into another.

**Note:** Measures and Member Unique Names (MUNs) are considered members.

<table>
<thead>
<tr>
<th>Source type</th>
<th>Target type</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure dimension</td>
<td>Hierarchy</td>
<td>A measure dimension can be used without any conversion as a hierarchy.</td>
</tr>
</tbody>
</table>

Appendix D. Limitations when using dimensional data sources 545
Coercion Rules for Numeric and String Values

The following table describes the coercion rules for expressions or prompts that return a numeric or string value. These rules apply to functions and operators that require an operand of a particular data type, as well as to binary operators that require operands of the same type.

<table>
<thead>
<tr>
<th>Source type</th>
<th>Target type</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure</td>
<td>Number, Date/Time, or String</td>
<td>The result is the value of the measure at the current cube intersection.</td>
</tr>
<tr>
<td>Measure</td>
<td>Member set</td>
<td>The member set contains the members of the measure dimension.</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>Member set</td>
<td>A set containing the members of the hierarchy.</td>
</tr>
<tr>
<td>Level</td>
<td>Member set</td>
<td>A set containing the members of the level.</td>
</tr>
<tr>
<td>Measure</td>
<td>Member</td>
<td>A measure can be used without any conversion as a member.</td>
</tr>
<tr>
<td>Measure</td>
<td>Tuple</td>
<td>This coercion is done in two steps. First, measures are coerced to members, and then members are coerced to tuples.</td>
</tr>
<tr>
<td>Measure</td>
<td>Member set</td>
<td>This coercion is done in two steps. First, measures are coerced to members, and then members are coerced to member sets.</td>
</tr>
<tr>
<td>Member</td>
<td>Tuple</td>
<td>The tuple consists of the member.</td>
</tr>
<tr>
<td>Member</td>
<td>Member set</td>
<td>The set consists of the member.</td>
</tr>
</tbody>
</table>

The equivalent expression is $\text{member}$.

<table>
<thead>
<tr>
<th>Source type</th>
<th>Target type</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure</td>
<td>Number, Date/Time, or String</td>
<td>The result is the value of the measure at the current cube intersection.</td>
</tr>
<tr>
<td>Source type</td>
<td>Target type</td>
<td>Result</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>Member</td>
<td>Number</td>
<td>The result is the value of the cell at the cube intersection specified by the member. This coercion is done in two steps. First, members are coerced to tuples, and then tuples are coerced to the target type.</td>
</tr>
<tr>
<td>Tuple</td>
<td>Number</td>
<td>The value of the cell at the cube intersection specified by the tuple. The equivalent expression is value (operand or parameter).</td>
</tr>
</tbody>
</table>

**Coercion Paths**
The following diagram shows how coercion rules are connected. The expression that coerces a source node type to a target node type is built by following the path that connects them. A dotted line indicates a coercion that does not require an explicit conversion function. This occurs whenever the source type is a specialization of the target type. For example, every measure is a member so you can use measures without conversion wherever a member is required.

![Coercion Diagram](image)

**Recommendation - Use Member Unique Name (MUN) Aliases**
If you are working with a dimensional data source, use MUN aliases to simplify building reports and expressions. In addition, you can tie the MUN back to the member in the package.

When working with dimensional data sources, IBM Cognos Business Intelligence uses MUNs to uniquely identify members. MUNs are very complex. For example, the MUN for the member 2012 might appear as
When you build expressions that involve members, the use of MUNs makes these expressions difficult to read and maintain.

**Tip:** To view the MUN of a member, in the **Source** tab, right-click the member and click **Properties**.

IBM Cognos Report Studio has an option that automatically generates an alias for MUNs (**Tools > Options > Report tab > Alias member unique names**). The MUN alias is a data item that has the MUN as its definition. For the previous example, you would see 2012 as the MUN alias in the **Expression** property for the data item. If the MUN alias is not enabled, you would see the complex MUN as shown above in the **Expression** property.

**Important:** Do not modify or rename MUN aliases, because they will lose their connection to their respective members in the package.

### Running a Report That Contains Members

If you are using members in a report and your modeler has updated the data source in a way that changes member unique names, the following problems may arise when you run the report.

- If the report queries an OLAP data source, you receive an error message explaining that some specific members cannot be found.
- If the report queries a dimensionally modeled relational (DMR) data source, data items whose member unique names have changed do not appear in the report. Calculations that refer to members that have changed no longer contain values from the members.

To resolve these problems, you must update the member unique names in the report. Open the query that contains the members that you need to update in **Query Explorer**. Delete the member from the pane in which it appears and reinsert it from the **Source** tab. For example, if you inserted the member as a detail filter, delete it from the **Detail Filters** pane and reinsert it.

### Limitations When Formatting Empty Cells in SAP BW Data Sources

When working with SAP BW data sources, if the SAP BW server administrator configured custom formatting for empty cells on the SAP BW server, this custom format does not appear in IBM Cognos Business Intelligence reports. Ask your administrator to configure the formatting of empty cells in IBM Cognos BI.

**Related tasks:**

- [“Suppress Empty Cells” on page 391](#)

Sparse data may result in crosstabs showing empty cells. For example, a crosstab that matches employees with products results in many rows of empty values for the revenue measure if the employee does not sell those products.
Appendix E. Limitations When Producing Reports in Microsoft Excel Format

There are limitations when producing reports in Microsoft Excel format.

Unable to Load Images from the IBM Cognos Business Intelligence Content Store in a Report

If a report contains an image whose URL points to the IBM Cognos Business Intelligence content store, the Microsoft Excel spreadsheet software generates an access violation error and shuts down.

This problem is a known issue in the Microsoft knowledge base, and Microsoft is currently investigating the problem. This problem occurs only in Excel 2002.

Blank Worksheet Appears

If the Microsoft Excel spreadsheet software cannot download a worksheet within a timeout period, Excel may instead open a blank worksheet.

Warning Message Appears When Excel Opens an IBM Cognos Business Intelligence Report

Each time the Microsoft Excel spreadsheet software opens an IBM Cognos Business Intelligence report, a warning message appears.

The warning message is as follows:

Some of the files in this Web page are not in the expected location. Do you want to download them anyway? If you are sure the Web page is from a trusted source, click Yes.

The Excel workbook in HTML/XML format requires the presence of the file filelist.xml. IBM Cognos BI does not allow the creation of local files on the client side. In addition, a local file that contains URLs introduces a security issue. Consequently, this message will appear whenever you open an IBM Cognos BI report in Excel. If you see this error message, click Yes to open the report.

Spreadsheet Content Not Saved for Reports Saved in XLS Format

If you open a report that was saved in XLS format or run a report in XLS format, and security settings in your Web browser are set so that you are prompted to open or save the report, do not click Save. If you save the report, the spreadsheet content will not be saved. This is because Microsoft Excel reports in Microsoft Office 2000 HTML format use relative paths to the spreadsheets. The relative URL paths are no longer available when you open a saved XLS report.

Instead, click Open first and then choose to save the report.
Unable to Load Excel Report in Netscape 7.01

This version of IBM Cognos Business Intelligence does not support loading Microsoft Excel reports in Netscape 7.01.

Unable to Nest Labels in Charts

Currently, it is not possible to specify nested labels for the category axis via XML.

Data Series Are Truncated

The Microsoft Excel spreadsheet software may group data series or categories differently when compared to a chart produced by IBM Cognos Business Intelligence.

A 1 KB buffer limit in Excel limits the maximum number of data series per chart to 120. Data series over 120 are truncated.

Colors Different From Those in HTML or PDF

If an IBM Cognos Business Intelligence report contains a chart that uses custom colors, the Microsoft Excel spreadsheet software may not be able to add the custom color to the Excel color palette. Excel will attempt to match the custom color to one of its available standard colors. As a result, colors in Excel may vary from those seen in HTML or PDF.

We recommend that you use standard colors in charts.

Repeating Pie Charts

If you have a report that has repeating pie charts and you define a chart title, the Microsoft Excel spreadsheet software will show each pie with a title that is a concatenation of the chart title and the data series. For example, if the chart title is Quantity Sold by Order Method and Product Line and the data series is Order method, the title of each pie in Excel will be Quantity Sold by Order Method and Product Line, order method.

Unable to Skip Discrete Axis Labels

In IBM Cognos Business Intelligence charts, you can control the skipping of discrete axis labels. This feature is not supported in Microsoft Excel charts.

Unsupported IBM Cognos BI Formatting

About 30% of the formatting functions available in IBM Cognos Business Intelligence are not supported in the Microsoft Excel spreadsheet software.

In particular, Excel does not allow changing locale-dependent formatting attributes, such as the following:

- Decimal Separator
- Exponential Symbol
- Group Separator
- Monetary Decimal Separator
- AM String
• Day Name
• Day Short Name
• Decimal Delimiter Symbol
• Month Name
• Month Short Name
• PM String

In addition, Excel does not support the following:
• Format Width
• International Currency Symbol
• List Separator
• Percent Symbol (Excel does not support percent symbols for charts)
• Multiplier
• Overline Text Format
• PerMill Symbol
• Plus Sign
• Scale (Excel has a different scaling formula than IBM Cognos BI)
• Calendar (Excel does not allow changing the calendar)
• Era Name
• First Day Of Week
• Show Era

**Cells Contain Series of #**

Cells in the Microsoft Excel spreadsheet software have a limit of 255 characters. If your report contains text strings that are longer than 255 characters, they will be formatted as text and appear as ######.

To resolve this problem, use fewer characters.

**Table and Column Widths**

The Microsoft Excel spreadsheet software does not support using percentages to determine the width of tables.

If the report contains only one table, the value of the width attribute for the Table element in the report specification determines the width of the table in the Excel worksheet. If the report contains more than one table, Excel determines the width of all the tables in the worksheet. If the tables are nested, the width specified for the outer table is used and, if necessary, the width is adjusted to accommodate data in the nested tables. The columns and rows around the table are merged to preserve the appearance of the nested table. When you save the workbook, only a single table is saved per worksheet.

**Secure Socket Layer (SSL) Is Not Supported in Some Excel Formats and Versions**

SSL is supported for only the Microsoft Excel 2002 format in Microsoft Excel 2002 and Microsoft Excel 2003.
Number Formats Become Currency Formats in Japanese Excel

A report uses the Number data format and you save it as Microsoft Excel output. When you open the report in the Japanese version of Microsoft Excel, the data format is listed as Currency rather than Number. This occurs because Japanese Excel interprets the standard Number data format slightly differently than other versions of Excel.

The value appears correctly in Number format. For example, if you specified five digits as your number format, five digits still appear. In Excel, click the Custom number format to see the exact format string being used.

Reports Show Data in Wrong Columns

A report contains a large amount of data that is presented using a large number of nested report objects, such as tables and blocks. When the report is produced in Microsoft Excel format, some of the data appears in the wrong columns. This occurs because Excel has a 64K limit on how many nested cell objects can appear in a single spreadsheet.

To solve this problem, you can redesign the report to present the data using non-nested structures.

Unable to Access Reports on Remote Servers

You cannot access a report in Microsoft Excel format on a remote server.

To resolve this problem, you must change the hostname portion of the gateway URI from localhost to either the IP address of the computer or the computer name. You do this using IBM Cognos Configuration.

Drill-through Reports Are Not Supported in Excel

IBM Cognos Business Intelligence does not support drill-through for reports in Microsoft Excel format.

Map Reports Are Not Supported in Excel

IBM Cognos Business Intelligence does not support map reports in Microsoft Excel format.

Unsupported Excel Formatting

IBM Cognos Business Intelligence does not support some formatting.

The following formatting functions that are available in the Microsoft Excel spreadsheet software are not supported by IBM Cognos BI:

- background images in table cells
- Excel-specific headers and footers
- text flow and justification
- floating text objects
- white space, normal, and wrap text formatting
- maximum characters
Some layouts do not show exactly in HTML and PDF due to Excel limitations.

**Hyperlink Buttons Are Not Supported in Excel**

The Microsoft Excel spreadsheet software does not support hyperlink buttons.

**Unable to View Reports in Excel Format Sent as Email Attachments**

IBM Cognos Business Intelligence can send Microsoft Excel reports in HTML and XML format by email. However, you cannot open them directly from the email message.

Save the Excel email attachments to your computer and view them from there.

**Unsupported Chart Properties in Excel**

The following IBM Cognos Business Intelligence chart properties are not supported in the Microsoft Excel spreadsheet software:

- tool tips
- conditional text
- depth
- visual angle
- show values
- marker text location
- show baseline
- new note
- new marker
- truncation text and allow n-degrees rotation category labels
- border
- margin
- box type
- font and font alignment
- footer
- subtitle
- regression line
- baseline
- Include Zero For Auto Scale

In addition, IBM Cognos BI makes sure that Excel reuses the same color palette that is defined in IBM Cognos BI. However, Excel can only use the first 16 colors from the IBM Cognos BI palette. If the number of categories in a chart exceeds 16, the rest of the colors are taken from the default Excel palette.

**Unsupported Chart Types in Excel**

About 30% of the chart types available in IBM Cognos Business Intelligence are not matched in the Microsoft Excel spreadsheet software.

The following chart types appear differently or are not supported. Charts that are not supported appear as a default column chart in Excel.

- Bipolar charts
- Bubble charts
  Excel does not support regression lines.

- Combination charts
  In Excel, combination charts appear as two-dimensional. If a three-dimensional or two-dimensional combination chart includes only an area, column, or line chart, only the one chart appears.

- Gauge charts

- Donut charts
  Excel may fill in the donut hole to accommodate extra measures. Excel shows donut three-dimensional charts as donut charts.

- Maps
  In Excel, no chart appears.

- Matrix charts

- Metrics range charts
  In Excel, they appear as combination charts with two line charts for the tolerance bars.

- Pareto charts
  In Excel, the cumulative line is not displayed.

- Pie charts
  Excel shows only one type of pie chart and may show it at a different angle.

- Progressive column charts

- Polar charts
  In Excel, a scatter chart appears instead.

- Quadrant charts

- Radar charts
  Excel does not support stacked area radar charts. Excel names area radar charts as filled radar charts. Excel shows radar charts as radar with markers.

- Scatter charts
  In Excel, three-dimensional scatter charts appear as two-dimensional scatter charts. The z-axis is dropped.

- 100% stacked bar and column charts
  In Excel, the percentage of each value in the bar or column is not displayed. Instead, the underlying values are displayed.

---

**Cell Height and Width Are Incorrect**

The width and height of cells that contain data with curly brackets {} or parentheses () may appear incorrectly.

This is because the Microsoft Excel spreadsheet software uses different word wrapping algorithms than IBM Cognos Business Intelligence.
Appendix F. Using the expression editor

An expression is any combination of operators, constants, functions, and other components that evaluates to a single value. You build expressions to create calculation and filter definitions. A calculation is an expression that you use to create a new value from existing values contained within a data item. A filter is an expression that you use to retrieve a specific subset of records.

You build expressions using the components that are defined below.

Operators

Operators specify what happens to the values on either side of the operator. Operators are similar to functions, in that they manipulate data items and return a result.

(  
Identifies the beginning of an expression.

Syntax
( expression )

)  
Identifies the end of an expression.

Syntax
( expression )

*  
Multiplies two numeric values.

Syntax
value1 * value2

,  
Separates expression components.

Syntax
expression ( parameter1, parameter2 )

/  
Divides two numeric values.

Syntax
value1 / value2
||
Concatenates, or joins, strings.

**Syntax**

```
string1 || string2
```

+  
Adds two numeric values.

**Syntax**

```
value1 + value2
```

-  
Subtracts two numeric values or negates a numeric value.

**Syntax**

```
value1 - value2
```

\(<\)

Compares the values that are represented by "value1" against "value2" and retrieves the values that are less than "value2".

**Syntax**

```
value1 < value2
```

\(<=\)

Compares the values that are represented by "value1" against "value2" and retrieves the values that are less than or equal to "value2".

**Syntax**

```
value1 <= value2
```

\(<>\)

Compares the values that are represented by "value1" against "value2" and retrieves the values that are not equal to "value2".

**Syntax**

```
value1 <> value2
```

=  
Compares the values that are represented by "value1" against "value2" and retrieves the values that are equal to "value2".

**Syntax**

```
value1 = value2
```

>  
Compares the values that are represented by "value1" against "value2" and retrieves the values that are greater than "value2".
Syntax
value1 > value2

->
Separates the components in a literal member expression.

Syntax
[namespace].[dimension].[hierarchy].[level]->[L1]

>=
Compares the values that are represented by "value1" against "value2" and retrieves the values that are greater than or equal to "value2".

Syntax
value1 >= value2

and
Returns "true" if the conditions on both sides of the expression are true.

Syntax
argument1 and argument2

auto
Works with summary expressions to define the scope to be adjusted based on the grouping columns in the query. The scope is context-dependent.

Syntax
aggregate_function { expression AUTO }

between
Determines if a value falls in a given range.

Syntax
expression between value1 and value2

Example
[Revenue] between 200 and 300

Result
Returns the number of results with revenues between 200 and 300.

Result data
<table>
<thead>
<tr>
<th>Revenue</th>
<th>Between</th>
</tr>
</thead>
<tbody>
<tr>
<td>$332.06</td>
<td>false</td>
</tr>
<tr>
<td>$230.55</td>
<td>true</td>
</tr>
<tr>
<td>$107.94</td>
<td>false</td>
</tr>
</tbody>
</table>
case
Works with when, then, else, and end. Case identifies the beginning of a specific situation, in which when, then, and else actions are defined.

Syntax
```
case expression { when expression then expression } [ else expression ] end
```

contains
Determines if "string1" contains "string2".

Syntax
```
string1 contains string2
```

currentMeasure
Keyword that can be used as the first argument of member summary functions. This function appears in the Total Revenue by Country sample report in the GO Data Warehouse (query) package.

Syntax
```
aggregate_function ( currentMeasure within set expression )
```

default
Works with the lookup construct.

Syntax
```
lookup (....) in (....) default (....)
```

distinct
A keyword used in an aggregate expression to include only distinct occurrences of values. See also the function unique.

Syntax
```
distinct dataItem
```

Example
```
count ( distinct [OrderDetailQuantity] )
```

Result
```
1704
```

else
Works with the if or case constructs. If the if condition or the case expression are not true, then the else expression is used. This function appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package.

Syntax
```
if ( condition ) then .... else ( expression ), or case .... else ( expression ) end
```
end

Indicates the end of a case or when construct.

Syntax
    case .... end

ends with

Determines if "string1" ends with "string2".

Syntax
    string1 ends with string2

escape

Determines if "string1" matches the pattern of "string2", with the character "char" optionally used to escape characters in the pattern string.

Syntax
    string1 LIKE string2 [ ESCAPE char ]

Example 1
[PRODUCT_LINE] like 'G%'

Result
All product lines that start with 'G'.

Example 2
[PRODUCT_LINE] like '%Ga%' escape 'a'

Result
All the product lines that end with 'G%'.

for

Works with summary expressions to define the scope of the aggregation in the query.

Syntax
    aggregate_function ( expression for expression { , expression } )

for all

Works with summary expressions to define the scope to be all the specified grouping columns in the query. See also the for clause.

Syntax
    aggregate_function ( expression for ALL expression { , expression } )

for any

Works with summary expressions to define the scope to be adjusted based on a subset of the grouping columns in the query. Equivalent to the for clause.

Syntax
aggregate_function (expression for ANY expression { , expression })

**for report**
Works with summary expressions to set the scope to be the whole query. See also the for clause. This function appears in the Customer Returns and Satisfaction sample report in the GO Data Warehouse (analysis) package.

**Syntax**
aggregate_function (expression for report)

**if**
Works with the then and else constructs. If defines a condition; when the if condition is true, the then expression is used. When the if condition is not true, the else expression is used. This function appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package.

**Syntax**
if (condition) then (expression) else (expression)

**in**
Determines if "expression1" exists in a given list of expressions.

**Syntax**
expression1 in (expression_list)

**in_range**
Determines if "expression1" exists in a given list of constant values or ranges.

**Syntax**
expression1 in_range {constant : constant [ , constant : constant]}

**Example 1**
[code] in_range { 5 }

**Result**
This is equivalent to [code] = 5.

**Example 2**
[code] in_range { 5: }

**Result**
This is equivalent to [code] >= 5.

**Example 3**
[code] in_range { :5 }

**Result**
This is equivalent to [code] <= 5.
Example 4
[code] in_range { 5:10 }

Result
This is equivalent to ( [code] >= 5 and [code] <= 10 ).

Example 5
[code] in_range { :5,10,20: }

Result
This is equivalent to ( [code] <= 5 or [code] = 10 or [code] >= 20 ).

is missing
Determines if "value" is undefined in the data.

Syntax
value is missing

is null
Determines if "value" is undefined in the data.

Syntax
value is null

is not missing
Determines if "value" is defined in the data.

Syntax
value is not missing

is not null
Determines if "value" is defined in the data.

Syntax
value is not null

like
Determines if "string1" matches the pattern of "string2", with the character "char" optionally used to escape characters in the pattern string.

Syntax
string1 LIKE string2 [ ESCAPE char ]

Example 1
[PRODUCT_LINE] like 'G%'

Result
All product lines that start with 'G'.
Example 2

[PRODUCT_LINE] like 'Ga%' escape 'a'

Result

All the product lines that end with 'G%'.

lookup

Finds and replaces data with a value you specify. It is preferable to use the case construct.

Syntax

lookup ( name ) in ( value1 --> value2 ) default ( expression )

Example

lookup ( [Country]) in ( 'Canada'--> ( [List Price] * 0.60), 'Australia'--> ( [List Price] * 0.80 ) ) default ( [List Price] )

not

Returns TRUE if "argument" is false or returns FALSE if "argument" is true.

Syntax

NOT argument

or

Returns TRUE if either of "argument1" or "argument2" are true.

Syntax

argument1 or argument2

prefilter

Performs a summary calculation before applying the summary filter.

Syntax

summary_function ([expression] prefilter)

Example

total ( [Quantity] for report prefilter )
summaryFilter: total( [Quantity] for [ProductNo] ) > 50000

Result

Sums the quantities in a report before the summary filter is applied.

Result data

<table>
<thead>
<tr>
<th>ProductNo</th>
<th>Total_forProductNo</th>
<th>Total_forReport</th>
<th>Total_forReport_Prefilter</th>
</tr>
</thead>
<tbody>
<tr>
<td>88</td>
<td>54928</td>
<td>298140</td>
<td>2215354</td>
</tr>
<tr>
<td>89</td>
<td>51126</td>
<td>298140</td>
<td>2215354</td>
</tr>
<tr>
<td>90</td>
<td>69996</td>
<td>298140</td>
<td>2215354</td>
</tr>
<tr>
<td>94</td>
<td>69004</td>
<td>298140</td>
<td>2215354</td>
</tr>
<tr>
<td>95</td>
<td>53086</td>
<td>298140</td>
<td>2215354</td>
</tr>
</tbody>
</table>
rows
Counts the number of rows output by the query. Use with Count ().

Syntax
count ( ROWS )

starts with
Determines if "string1" starts with "string2".

Syntax
string1 starts with string2

then
Works with the if or case constructs. When the if condition or the when expression are true, the then expression is used. This function appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package.

Syntax
if ( condition ) then ..., or case expression when expression
then .... end

when
Works with the case construct. You can define conditions to occur when the WHEN expression is true.

Syntax
case [expression] when ... end

Summaries
This list contains predefined functions that return either a single summary value for a group of related values or a different summary value for each instance of a group of related values.

Statistical functions
This list contains predefined summary functions of statistical nature.

corr
Returns the coefficient of correlation of a set of number pairs. This is computed as follows: COVAR_POP(numeric_expression1, numeric_expression2) / (STDDEV_POP(numeric_expression1) * STDDEV_POP(numeric_expression2))

Syntax
corr ( numeric_expression1 , numeric_expression2 [ auto ] )
corr ( numeric_expression1 , numeric_expression2 for [ all|any ] expression ( , expression ) )
corr ( numeric_expression1 , numeric_expression2 for report )
Example

corr( Cost, Margin for report)

Result

The coefficient of correlation between Cost and Margin.

Result data

<table>
<thead>
<tr>
<th>Cost</th>
<th>Margin</th>
<th>corr (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.33</td>
<td>0.0872648</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>0.0872648</td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>0.0872648</td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>0.0872648</td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>0.0872648</td>
</tr>
</tbody>
</table>

**covariance-pop**

Returns the population covariance of a set of number pairs.

Syntax

covariance-pop ( numeric_expression1, numeric_expression2 )
covariance-pop ( numeric_expression1, numeric_expression2 for [all|any] expression [^, expression] )
covariance-pop ( numeric_expression1, numeric_expression2 for report )

Example

covariance-pop ( Cost, Margin for report)

Result

The population covariance between Cost and Margin.

Result data

<table>
<thead>
<tr>
<th>Cost</th>
<th>Margin</th>
<th>covariance-pop (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.33</td>
<td>0.032384</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>0.032384</td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>0.032384</td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>0.032384</td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>0.032384</td>
</tr>
</tbody>
</table>

**covariance-samp**

Returns the sample covariance of a set of number pairs.

Syntax

covariance-samp ( numeric_expression1, numeric_expression2 )
covariance-samp ( numeric_expression1, numeric_expression2 for [all|any] expression [^, expression] )
covariance-samp ( numeric_expression1, numeric_expression2 for report )

Example

covariance-samp ( Cost, Margin for report)
Result

The sample covariance between Cost and Margin.

Result data

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>covariance-samp (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Cost</td>
<td>Margin</td>
</tr>
<tr>
<td>4</td>
<td>0.33</td>
<td>0.04048</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>0.04048</td>
<td></td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>0.04048</td>
<td></td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>0.04048</td>
<td></td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>0.04048</td>
<td></td>
</tr>
</tbody>
</table>

regression-average-x

Returns the average of the independent variable (numeric_expression2) of the regression line.

Syntax

regression-average-x ( numeric_expression1 , numeric_expression2 )
regression-average-x ( numeric_expression1 , numeric_expression2 for [ all|any ] expression { , expression } )
regression-average-x ( numeric_expression1 , numeric_expression2 for report )

Example

regression-average-x ( Cost , Margin for report )

Result

The average of Margin within the regression line for Cost and Margin.

Result data

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>regression-average-x (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Cost</td>
<td>Margin</td>
</tr>
<tr>
<td>4</td>
<td>0.33</td>
<td>0.284</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>0.284</td>
<td></td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>0.284</td>
<td></td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>0.284</td>
<td></td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>0.284</td>
<td></td>
</tr>
</tbody>
</table>

regression-average-y

Returns the average of the dependent variable (numeric_expression1) of the regression line.

Syntax

regression-average-y ( numeric_expression1 , numeric_expression2 )
regression-average-y ( numeric_expression1 , numeric_expression2 for [ all|any ] expression { , expression } )
regression-average-y ( numeric_expression1 , numeric_expression2 for report )

Example

regression-average-y ( Cost , Margin for report )
Result

The average of Cost within the regression line for Cost and Margin.

Result data

<table>
<thead>
<tr>
<th>Cost</th>
<th>Margin</th>
<th>regression-average-y (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.33</td>
<td>13.824</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>13.824</td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>13.824</td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>13.824</td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>13.824</td>
</tr>
</tbody>
</table>

regression-count

Returns the number of non-null numbers used to fit the regression line.

Syntax

regression-count ( numeric_expression1 , numeric_expression2 )
regression-count ( numeric_expression1 , numeric_expression2 for [ all|any ] expression { , expression } )
regression-count ( numeric_expression1 , numeric_expression2 for report )

Example

regression-count ( Cost , Margin for report )

Result

The number of non-null numbers used to fit the regression line for Cost and Margin.

Result data

<table>
<thead>
<tr>
<th>Cost</th>
<th>Margin</th>
<th>regression-count (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.33</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>5</td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>5</td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>5</td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>5</td>
</tr>
</tbody>
</table>

regression-intercept

Returns the y-intercept of the regression line. This is computed as follows:

\[ \text{AVG}(\text{numeric_expression1}) - \text{REGR_SLOPE}(\text{numeric_expression1}, \text{numeric_expression2}) \times \text{AVG}(\text{numeric_expression2}) \]

Syntax

regression-intercept ( numeric_expression1 , numeric_expression2 )
regression-intercept ( numeric_expression1 , numeric_expression2 for [ all|any ] expression { , expression } )
regression-intercept ( numeric_expression1 , numeric_expression2 for report )

Example

regression-intercept ( Cost , Margin for report )
Result

The y-intercept of the regression line for Cost and Margin.

Result data

<table>
<thead>
<tr>
<th>Cost</th>
<th>Margin</th>
<th>regression-intercept (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.33</td>
<td>5.18015038</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>5.18015038</td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>5.18015038</td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>5.18015038</td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>5.18015038</td>
</tr>
</tbody>
</table>

**regression-r2**

Returns the coefficient of determination (also known as "R-squared" or "goodness of fit") of the regression line. This value is computed based on the following conditions: IF VAR_POP(numeric_expression2) = 0 THEN NULL IF VAR_POP(numeric_expression1) = 0 AND VAR_POP(numeric_expression2) <> 0 THEN 1 IF VAR_POP(numeric_expression1) > 0 and VAR_POP(numeric_expression2) <> 0 THEN POWER(CORR (numeric_expression1, numeric_expression2))

Syntax

regression-r2 ( numeric_expression1 , numeric_expression2 )
regression-r2 ( numeric_expression1 , numeric_expression2 for [ all|any ] expression { , expression } )
regression-r2 ( numeric_expression1 , numeric_expression2 for report )

Example

regression-r2 ( Cost , Margin for report )

Result

The coefficient of determination of the regression line for Cost and Margin.

Result data

<table>
<thead>
<tr>
<th>Cost</th>
<th>Margin</th>
<th>regression-r2 (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.33</td>
<td>0.00761514</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>0.00761514</td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>0.00761514</td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>0.00761514</td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>0.00761514</td>
</tr>
</tbody>
</table>

**regression-slope**

Returns the slope of the regression line. This is computed as follows:

COVAR_POP(numeric_expression1,numeric_expression2) / VAR_POP(numeric_expression2)

Syntax
regression-slope ( numeric_expression1 , numeric_expression2 )

Example

regression-slope ( Cost , Margin )

Result

The slope of the regression line for Cost and Margin.

Result data

<table>
<thead>
<tr>
<th>Cost</th>
<th>Margin</th>
<th>regression-slope (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.33</td>
<td>30.43609023</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>30.43609023</td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>30.43609023</td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>30.43609023</td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>30.43609023</td>
</tr>
</tbody>
</table>

regression-sxx

Returns the following computation after eliminating NULL pairs:
REGR_COUNT(numeric_expression1, numeric_expression2) * VAR_POP(numeric_expression2)

Syntax

regression-sxx ( numeric_expression1 , numeric_expression2 )

Example

regression-sxx ( Cost , Margin for report )

Result

The sxx computation of the regression line for Cost and Margin.

Result data

<table>
<thead>
<tr>
<th>Cost</th>
<th>Margin</th>
<th>regression-sxx (Cost, Margin for report)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.33</td>
<td>0.00532</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>0.00532</td>
</tr>
<tr>
<td>9.22</td>
<td>0.23</td>
<td>0.00532</td>
</tr>
<tr>
<td>15.93</td>
<td>0.28</td>
<td>0.00532</td>
</tr>
<tr>
<td>34.97</td>
<td>0.3</td>
<td>0.00532</td>
</tr>
</tbody>
</table>

regression-sxy

Returns the following computation after eliminating NULL pairs:
REGR_COUNT(numeric_expression1, numeric_expression2) * COVAR_POP(numeric_expression1, numeric_expression2)
Syntax

\texttt{regression-sxy (numeric_expression1, numeric_expression2 )}
\texttt{regression-sxy (numeric_expression1, numeric_expression2 for [all|any] expression {, expression} )}
\texttt{regression-sxy (numeric_expression1, numeric_expression2 for report )}

Example

\texttt{regression-sxy (Cost, Margin for report)}

Result

The sxy computation of the regression line for Cost and Margin.

Result data

\begin{tabular}{lll}
\textbf{Cost} & \textbf{Margin} & \texttt{regression-sxy (Cost, Margin for report)} \\
4 & 0.33 & 0.16192 \\
5 & 0.28 & 0.16192 \\
9.22 & 0.23 & 0.16192 \\
15.93 & 0.28 & 0.16192 \\
34.97 & 0.3 & 0.16192 \\
\end{tabular}

\textbf{regression-syy}

Returns the following computation after eliminating NULL pairs:
\texttt{REGR\_COUNT(numeric_expression1, numeric_expression2) * VAR\_POP(numeric_expression1)}

Syntax

\texttt{regression-syy (numeric_expression1, numeric_expression2 )}
\texttt{regression-syy (numeric_expression1, numeric_expression2 for [all|any] expression {, expression} )}
\texttt{regression-syy (numeric_expression1, numeric_expression2 for report )}

Example

\texttt{regression-syy (Cost, Margin for report)}

Result

The syy computation of the regression line for Cost and Margin.

Result data

\begin{tabular}{lll}
\textbf{Cost} & \textbf{Margin} & \texttt{regression-syy (Cost, Margin for report)} \\
4 & 0.33 & 647.15932 \\
5 & 0.28 & 647.15932 \\
9.22 & 0.23 & 647.15932 \\
15.93 & 0.28 & 647.15932 \\
34.97 & 0.3 & 647.15932 \\
\end{tabular}
**standard-deviation**
Returns the standard deviation of selected data items.

**Syntax**
```plaintext
standard-deviation ( expression [ auto ] )
standard-deviation ( expression for [ all|any ] expression { , expression } )
standard-deviation ( expression for report )
```

**Example**
```plaintext
standard-deviation ( ProductCost )
```

**Result**
Returns a value indicating the deviation between product costs and the average product cost.

**standard-deviation-pop**
Computes the population standard deviation and returns the square root of the population variance.

**Syntax**
```plaintext
standard-deviation-pop ( expression [ auto ] )
standard-deviation-pop ( expression for [ all|any ] expression { , expression } )
standard-deviation-pop ( expression for report )
```

**Example**
```plaintext
standard-deviation-pop ( ProductCost )
```

**Result**
Returns a value of the square root of the population variance.

**standard-deviation-samp**
Computes the sample standard deviation and returns the square root of the population variance.

**Syntax**
```plaintext
standard-deviation-samp ( expression [ auto ] )
standard-deviation-samp ( expression for [ all|any ] expression { , expression } )
standard-deviation-samp ( expression for report )
```

**Example**
```plaintext
standard-deviation-samp ( ProductCost )
```

**Result**
Returns a value of the square root of the sample variance.

**variance**
Returns the variance of selected data items.

**Syntax**
variance ( expression [ auto ] )
variance ( expression for [ all|any ] expression { , expression } )
variance ( expression for report )

Example
variance ( Product Cost )

Result
Returns a value indicating how widely product costs vary from the average product cost.

variance-pop
Returns the population variance of a set of numbers after discarding the nulls in this set.

Syntax
variance-pop ( expression [ auto ] )
variance-pop ( expression for [ all|any ] expression { , expression } )
variance-pop ( expression for report )

Example
variance-pop ( Qty )

Result
For each row, returns the population variance of a set of numbers after discarding the nulls in this set.

variance-samp
Returns the sample variance of a set of numbers after discarding the nulls in this set.

Syntax
variance-samp ( expression [ auto ] )
variance-samp ( expression for [ all|any ] expression { , expression } )
variance-samp ( expression for report )

Example
variance-samp ( Qty )

Result
For each row, returns the sample variance of a set of numbers after discarding the nulls in this set.

aggregate
Returns a calculated value using the appropriate aggregation function, based on the aggregation type of the expression. This function appears in the Budget vs. Actual sample report in the GO Data Warehouse (analysis) package.

Syntax
aggregate ( expression [ auto ] )
aggregate ( expression for [ all|any ] expression { , expression } )
aggregate ( expression for report )

average
Returns the average value of selected data items. Distinct is an alternative expression that is compatible with earlier versions of the product.

Syntax
average ( [ distinct ] expression [ auto ] )
average ( [ distinct ] expression for [ all|any ] expression { ,
expression } )
average ( [ distinct ] expression for report )

Example
average ( Sales )

Result
Returns the average of all Sales values.

count
Returns the number of selected data items excluding null values. Distinct is an alternative expression that is compatible with earlier versions of the product.

Syntax
count ( [ distinct ] expression [ auto ] )
count ( [ distinct ] expression for [ all|any ] expression { ,
expression } )
count ( [ distinct ] expression for report )

Example
count ( Sales )

Result
Returns the total number of entries under Sales.

maximum
Returns the maximum value of selected data items. Distinct is an alternative expression that is compatible with earlier versions of the product.

Syntax
maximum ( [ distinct ] expression [ auto ] )
maximum ( [ distinct ] expression for [ all|any ] expression { ,
expression } )
maximum ( [ distinct ] expression for report )

Example
maximum ( Sales )

Result
Returns the maximum value out of all Sales values.
median

Returns the median value of selected data items.

**Syntax**

```
median ( expression [ auto ] )
machine ( expression for [ all|any ] expression { , expression } )
machine ( expression for report )
```

minimum

Returns the minimum value of selected data items. Distinct is an alternative expression that is compatible with earlier versions of the product.

**Syntax**

```
minimum ( [ distinct ] expression [ auto ] )
machine ( [ distinct ] expression for [ all|any ] expression { , expression } )
machine ( [ distinct ] expression for report )
```

**Example**

```
minimum ( Sales )
```

**Result**

Returns the minimum value out of all Sales values.

moving-average

Returns a moving average by row for a specified set of values of over a specified number of rows. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

**Syntax**

```
moving-average ( numeric_expression , numeric_expression [ at expression { , expression } ] [ <for-option> ] [ prefilter ] )
moving-average ( numeric_expression , numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto
```

**Example**

```
moving-average ( Qty, 3 )
```

**Result**

For each row, returns the quantity and a moving average of the current row and the preceding two rows.

**Result data**

<table>
<thead>
<tr>
<th>Qty</th>
<th>Moving-Average (Qty, 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>700</td>
<td>450</td>
</tr>
<tr>
<td>400</td>
<td>433.3333</td>
</tr>
<tr>
<td>200</td>
<td>433.3333</td>
</tr>
<tr>
<td>200</td>
<td>266.6667</td>
</tr>
<tr>
<td>200</td>
<td>300.0000</td>
</tr>
<tr>
<td>500</td>
<td></td>
</tr>
</tbody>
</table>
moving-total

Returns a moving total by row for a specified set of values over a specified number of rows. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

Syntax

moving-total ( numeric_expression , numeric_expression [ at expression { , expression } ] [ <for-option> ] [ prefilter ] )
moving-total ( numeric_expression , numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto

Example

moving-total ( Qty , 3 )

Result

For each row, returns the quantity and a moving total of the current row and the preceding two rows.

Result data

<table>
<thead>
<tr>
<th>Qty</th>
<th>Moving-Total (Qty, 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>700</td>
<td>900</td>
</tr>
<tr>
<td>400</td>
<td>1300</td>
</tr>
<tr>
<td>200</td>
<td>1300</td>
</tr>
<tr>
<td>200</td>
<td>800</td>
</tr>
<tr>
<td>500</td>
<td>900</td>
</tr>
</tbody>
</table>

percentage

Returns the percent of the total value for selected data items. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources. This function appears in the Percentage Calculation (by year) interactive sample report.

Syntax

percentage ( numeric_expression [ at expression { , expression } ] [ <for-option> ] [ prefilter ] )
percentage ( numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto

Example

percentage ( Sales 98 )

Result

Returns the percentage of the total sales for 1998 that is attributed to each sales representative.

Result data

<table>
<thead>
<tr>
<th>Sales Rep</th>
<th>Sales 98</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbons</td>
<td>60646</td>
<td>7.11%</td>
</tr>
</tbody>
</table>
percentile

Returns a value, on a scale of one hundred, that indicates the percent of a distribution that is equal to or below the selected data items. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

Syntax

```
percentile ( numeric_expression [ at expression { , expression } ]
[ <for-option> ] [ prefilter ] )
percentile ( numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto
```

Example

```
percentile ( Sales 98 )
```

Result

For each row, returns the percentage of rows that are equal to or less than the quantity value of that row.

Result data

<table>
<thead>
<tr>
<th>Qty</th>
<th>Percentile (Qty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>1</td>
</tr>
<tr>
<td>700</td>
<td>0.875</td>
</tr>
<tr>
<td>600</td>
<td>0.75</td>
</tr>
<tr>
<td>500</td>
<td>0.625</td>
</tr>
<tr>
<td>400</td>
<td>0.5</td>
</tr>
<tr>
<td>400</td>
<td>0.5</td>
</tr>
<tr>
<td>200</td>
<td>0.25</td>
</tr>
<tr>
<td>200</td>
<td>0.25</td>
</tr>
</tbody>
</table>

quantile

Returns the rank of a value within a range that you specify. It returns integers to represent any range of ranks, such as 1 (highest) to 100 (lowest). The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

Syntax

```
quantile ( numeric_expression , numeric_expression [ at expression { , expression } ]
[ <for-option> ] [ prefilter ] )
quantile ( numeric_expression , numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto
```

Example

```
quantile ( Qty, 4 )
```

Result
Returns the quantity, the rank of the quantity value, and the quantity values broken down into 4 quantile groups (quartiles).

**Result data**

<table>
<thead>
<tr>
<th>Qty</th>
<th>Rank (Qty)</th>
<th>Quantile (Qty, 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>700</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>600</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>500</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>400</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>400</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>200</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>200</td>
<td>7</td>
<td>4</td>
</tr>
</tbody>
</table>

**quartile**

Returns the rank of a value, represented as integers from 1 (highest) to 4 (lowest), relative to a group of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

**Syntax**

quartile ( numeric_expression [ at expression { , expression } ] [ <for-option> ] [ prefilter ] )
quartile ( numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression } | for report | auto

**Example**

quartile ( Qty )

**Result**

Returns the quantity and the quartile of the quantity value represented as integers from 1 (highest) to 4 (lowest).

**Result data**

<table>
<thead>
<tr>
<th>Qty</th>
<th>Quartile (Qty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>450</td>
<td>1</td>
</tr>
<tr>
<td>400</td>
<td>1</td>
</tr>
<tr>
<td>350</td>
<td>2</td>
</tr>
<tr>
<td>300</td>
<td>2</td>
</tr>
<tr>
<td>250</td>
<td>3</td>
</tr>
<tr>
<td>200</td>
<td>3</td>
</tr>
<tr>
<td>150</td>
<td>4</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
</tr>
</tbody>
</table>

**rank**

Returns the rank value of selected data items. The sort order is optional; descending order (DESC) is assumed by default. If two or more rows tie, then there is a gap in the sequence of ranked values (also known as Olympic ranking). The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources. Distinct is an alternative expression that is compatible with earlier versions of the
product. Null values are ranked last. This function appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package.

Syntax

rank ( expression [ ASC|DESC ] { , expression [ ASC|DESC ] } [ at expression { , expression } ] [ <for-option> ] [ prefilter ] )

rank ( [ distinct ] expression [ ASC|DESC ] { , expression [ ASC|DESC ] } [ <for-option> ] [ prefilter ] )

<for-option> ::= for expression { , expression }|for report|auto

Example

rank ( Sales 98 )

Result

For each row, returns the rank value of sales for 1998 that is attributed to each sales representative. Some numbers are skipped when a tie between rows occurs.

Result data

<table>
<thead>
<tr>
<th>Sales Rep</th>
<th>Sales 98</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbons</td>
<td>60000</td>
<td>1</td>
</tr>
<tr>
<td>Flertjan</td>
<td>50000</td>
<td>2</td>
</tr>
<tr>
<td>Cornel</td>
<td>50000</td>
<td>2</td>
</tr>
<tr>
<td>Smith</td>
<td>48000</td>
<td>4</td>
</tr>
</tbody>
</table>

Running-average

Returns the running average by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

Syntax

running-average ( numeric_expression [ at expression { , expression } ] [ <for-option> ] [ prefilter ] )

running-average ( numeric_expression [ <for-option> ] [ prefilter ] )

<for-option> ::= for expression { , expression }|for report|auto

Example

running-average ( Qty )

Result

For each row, returns the quantity and a running average of the current and the previous rows.

Result data

<table>
<thead>
<tr>
<th>Name</th>
<th>Qty</th>
<th>Avg</th>
<th>Running-Average for name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>7</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Smith</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Smith</td>
<td>6</td>
<td>5</td>
<td>5.33</td>
</tr>
<tr>
<td>Smith</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Wong</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Wong</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
running-count

Returns the running count by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

Syntax

```
running-count ( numeric_expression [ at expression | expression ] [ <for-option> ] [ prefilter ] )
```

**Example**

```
running-count ( Qty )
```

**Result**

For each row, returns the quantity and a running count of the position of the current row.

**Result data**

<table>
<thead>
<tr>
<th>Name</th>
<th>Qty</th>
<th>Count</th>
<th>Running-Count for name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>7</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Smith</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Smith</td>
<td>6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Smith</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Wong</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Wong</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

running-difference

Returns a running difference by row, calculated as the difference between the value for the current row and the preceding row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

Syntax

```
running-difference ( numeric_expression [ at expression | expression ] [ <for-option> ] [ prefilter ] )
```

**Example**

```
running-difference ( Qty )
```

**Result**

For each row, returns the quantity and a running difference between the value for the current row and the preceding row.

**Result data**
<table>
<thead>
<tr>
<th>Name</th>
<th>Qty</th>
<th>Running-Difference for name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>7</td>
<td>NULL</td>
</tr>
<tr>
<td>Smith</td>
<td>3</td>
<td>-4</td>
</tr>
<tr>
<td>Smith</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>Smith</td>
<td>4</td>
<td>-2</td>
</tr>
<tr>
<td>Wong</td>
<td>3</td>
<td>-1</td>
</tr>
<tr>
<td>Wong</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

**running-maximum**

Returns the running maximum by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

**Syntax**

```
running-maximum ( numeric_expression [ at expression { , expression } ] [ <for-option> ] [ prefilter ] )
```

**Example**

```
running-maximum ( Qty )
```

**Result**

For each row, returns the quantity and a running maximum of the current and previous rows.

**Result data**

<table>
<thead>
<tr>
<th>Name</th>
<th>Qty</th>
<th>Max</th>
<th>Running-Maximum (Qty) for name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>2</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>Smith</td>
<td>3</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Smith</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Smith</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Wong</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Wong</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

**running-minimum**

Returns the running minimum by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

**Syntax**

```
running-minimum ( numeric_expression [ at expression { , expression } ] [ <for-option> ] [ prefilter ] )
```

**Example**

```
running-minimum ( Qty )
```
Result

For each row, returns the quantity and a running minimum of the current and previous rows.

Result data

<table>
<thead>
<tr>
<th>Name</th>
<th>Qty</th>
<th>Min</th>
<th>Running-Minimum (Qty) for name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>7</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Smith</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Smith</td>
<td>6</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Smith</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Wong</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Wong</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

running-total

Returns a running total by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

Syntax

running-total ( numeric_expression [ at expression { , expression } ] [ <for-option> ] [ prefilter ] )
running-total ( numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto

Example

running-total ( Qty )

Result

For each row, returns the quantity and a running total of the current and previous rows.

Result data

<table>
<thead>
<tr>
<th>Name</th>
<th>Qty</th>
<th>Total</th>
<th>Running-Total (Qty) for name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>2</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>Smith</td>
<td>3</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>Smith</td>
<td>6</td>
<td>18</td>
<td>11</td>
</tr>
<tr>
<td>Smith</td>
<td>7</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Wong</td>
<td>3</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>Wong</td>
<td>5</td>
<td>12</td>
<td>8</td>
</tr>
</tbody>
</table>

total

Returns the total value of selected data items. Distinct is an alternative expression that is compatible with earlier versions of the product. This function appears in the Budget vs. Actual sample report in the GO Data Warehouse (analysis) package.

Syntax
total ([ distinct ] expression [ auto ])
total ([ distinct ] expression for [ all|any ] expression { ,
expression })
total ([ distinct ] expression for report)

Example
total ( Sales)

Result
Returns the total value of all Sales values.

Member Summaries
This list contains predefined functions that return either a single summary value for a set of members or a different summary value for each member of a set of members.

aggregate
Returns a calculated value using the appropriate aggregation function based on the aggregation type of the expression.

Syntax
aggregate (< currentMeasure|numeric_expression > within set
set_expression)
aggregate (< currentMeasure|numeric_expression > within <
detail|aggregate > expression)

average
Returns the average value of the selected data items.

Syntax
average (< currentMeasure|numeric_expression > within set
set_expression)
average (< currentMeasure|numeric_expression > within <
detail|aggregate > expression)

Example
average ( Sales)

Result
Returns the average of all Sales values.

count
Returns the number of selected data items excluding null values.

Syntax
count (< currentMeasure|numeric_expression > within set
set_expression)
count (< currentMeasure|numeric_expression > within <
detail|aggregate > expression)

Example
count ( Sales)
**Result**

Returns the total number of entries under Sales.

**maximum**

Returns the maximum value of selected data items.

**Syntax**

\[
\text{maximum} \left( \text{currentMeasure|numeric_expression} \right. \text{within set set_expression} \\
\text{maximum} \left( \text{currentMeasure|numeric_expression} \right. \text{within } \text{detail|aggregate} \text{ expression} \\
\text{maximum} \left( \text{currentMeasure|numeric_expression} \right. \text{within detail|aggregate} \\
\text{Example}
\]

maximum ( Sales )

**Result**

Returns the maximum value out of all Sales values.

**median**

Returns the median value of selected data items.

**Syntax**

\[
\text{median} \left( \text{currentMeasure|numeric_expression} \right. \text{within set set_expression} \\
\text{median} \left( \text{currentMeasure|numeric_expression} \right. \text{within } \text{detail|aggregate} \text{ expression} \\
\text{median} \left( \text{currentMeasure|numeric_expression} \right. \text{within detail|aggregate} \\
\text{Example}
\]

median ( Sales )

**minimum**

Returns the minimum value of selected data items.

**Syntax**

\[
\text{minimum} \left( \text{currentMeasure|numeric_expression} \right. \text{within set set_expression} \\
\text{minimum} \left( \text{currentMeasure|numeric_expression} \right. \text{within } \text{detail|aggregate} \text{ expression} \\
\text{minimum} \left( \text{currentMeasure|numeric_expression} \right. \text{within detail|aggregate} \\
\text{Example}
\]

minimum ( Sales )

**Result**

Returns the minimum value out of all Sales values.

**percentage**

Returns the percent of the total value for the selected data items.

**Syntax**

\[
\text{percentage} \left( \text{numeric_expression} \left[ \text{tuple member_expression} \left[ , \text{member_expression} \right] \right] \text{within set set_expression} \\
\text{percentage} \left( \text{numeric_expression} \left[ \text{tuple member_expression} \left[ , \text{member_expression} \right] \right] \text{within detail|aggregate} \text{ expression} \\
\text{percentage} \left( \text{numeric_expression} \left[ \text{tuple member_expression} \left[ , \text{member_expression} \right] \right] \text{within detail|aggregate} \\
\text{Example
percentage ( [gosales].[sales measures].[quantity] tuple [gosales]. [Staff].[].[department] -> [West] within set children ( [gosales]. [Staff].[].[Staff] ) )

**percentile**

Returns a value, on a scale from 0 to 100, that indicates the percent of a distribution that is equal to or below the selected data items.

**Syntax**

percentile ( numeric_expression [ tuple member_expression { ,
member_expression } ] within set set_expression )

**quantile**

Returns the rank of a value for the specified range. It returns integers to represent any range of ranks, such as 1 (highest) to 100 (lowest).

**Syntax**

quantile ( numeric_expression , numeric_expression [ tuple
member_expression { , member_expression } ] within set
set_expression )

**quartile**

Returns the rank of a value, represented as integers from 1 (highest) to 4 (lowest), relative to a group of values.

**Syntax**

quartile ( numeric_expression [ tuple member_expression { ,
member_expression } ] within set set_expression )

**rank**

Returns the rank value of the selected data items. The type of ranking returned (Olympic, dense, or serial) is data source dependent. The sort order is optional; DESC is assumed by default.

**Syntax**

rank ( numeric_expression [ ASC|DESC ] [ tuple member_expression { ,
member_expression } ] within set set_expression )

**Example**

rank ( [gosales].[sales measures].[quantity] tuple [gosales].[Staff]. [].[department] -> [West] within set children ( [gosales].[Staff].[]. [Staff] ) )

**standard-deviation**

Returns the standard deviation of the selected data items.

**Syntax**

standard-deviation ( < currentMeasure|numeric_expression > within set
set_expression )
standard-deviation ( < currentMeasure|numeric_expression > within <
detail|aggregate > expression )
**standard-deviation-pop**

Returns the standard deviation population of the selected data items.

**Syntax**

```
standard-deviation-pop ( < currentMeasure|numeric_expression > within set set_expression )
standard-deviation-pop ( < currentMeasure|numeric_expression > within < detail|aggregate > expression )
```

**total**

Returns the total value of the selected data items.

**Syntax**

```
total ( < currentMeasure|numeric_expression > within set set_expression )
total ( < currentMeasure|numeric_expression > within < detail|aggregate > expression )
```

**variance**

Returns the variance of the selected data items.

**Syntax**

```
variance ( < currentMeasure|numeric_expression > within set set_expression )
variance ( < currentMeasure|numeric_expression > within < detail|aggregate > expression )
```

**variance-pop**

Returns the variance population of the selected data items.

**Syntax**

```
variance-pop ( < currentMeasure|numeric_expression > within set set_expression )
variance-pop ( < currentMeasure|numeric_expression > within < detail|aggregate > expression )
```

---

**Constants**

A constant is a fixed value that you can use in an expression.

**date**

Inserts the current system date.

**date-time**

Inserts the current system date and time.

**time with time zone**

Inserts a zero time with time zone.

**timestamp with time zone**

Inserts an example of a timestamp with time zone.
interval
Inserts a zero interval: 000 00:00:00.000.

interval year
Inserts a zero year interval: 0 year.

interval month
Inserts a zero month interval: 0 month.

interval year to month
Inserts a zero year to month interval: 0000-00 year to month.

interval day
Inserts a zero day interval: 0 day.

interval hour
Inserts a zero hour interval: 0 hour.

interval minute
Inserts a zero minute interval: 0 minute.

interval second
Inserts a zero second interval: 0 second.

interval day to hour
Inserts a zero day to hour interval: 0 00 day to hour.

interval day to minute
Inserts a zero day to minute interval: 0 00:00 day to minute.

interval day to second
Inserts a zero day to second interval: 0 00:00:00.000000000 day to second.

interval hour to minute
Inserts a zero hour to minute interval: 00:00 hour to minute.

interval hour to second
Inserts a zero hour to second interval: 00:00:00.000000000 hour to second.

interval minute to second
Inserts a zero minute to second interval: 00:00.000000000 minute to second.

null
Inserts "null" if the expression conditions are not met.

number
Inserts the number 0, which can be replaced with a new numeric value.
string

Inserts an empty string as two single quotation marks between which you can type
a string.

time

Inserts the current system time.

Constructs

This list contains constructs and templates that can be used to create an expression.
Templates combine multiple functions into a group. For example, the search case
template includes the case, when, else, and end functions.

if then else

This construct is the template for an if...then...else statement. This construct
appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse
(analysis) package.

Syntax

IF ([Country] = 'Canada') THEN ([List Price] * 0.60) ELSE ([List
Price])

in_range

This is the template for an in_range expression.

Syntax

[code] IN_RANGE { :30 , 40, 50, 999: }

Example 1

[code] IN_RANGE { 5 }

Result

This is equivalent to [code] = 5.

Example 2

[code] IN_RANGE { 5: }

Result

This is equivalent to [code] >= 5.

Example 3

[code] IN_RANGE { :5 }

Result

This is equivalent to [code] <= 5.

Example 4

[code] IN_RANGE { 5:10 }

Result
This is equivalent to \(( \text{[code]} \geq 5 \text{ and } \text{[code]} \leq 10 )\).

**Example 5**
\[
\text{[code]} \text{ IN_RANGE } \{ 5,10,20 : \}
\]

**Result**
This is equivalent to \(( \text{[code]} \leq 5 \text{ or } \text{[code]} = 10 \text{ or } \text{[code]} \geq 20 )\).

**search case**
This construct is the template for a search case, including the CASE, WHEN, ELSE, and END functions.

**Syntax**
CASE WHEN \([\text{Country}] = 'Canada' \) THEN \((\text{[List Price]} \times 0.60)\) WHEN \([\text{CountryCode}] > 100 \) THEN \(\text{[List Price]} \times 0.80\) ELSE \(\text{[List Price]}\) END

**simple case**
This construct is the template for a simple case, including the CASE, WHEN, ELSE, and END functions.

**Syntax**
CASE \([\text{Country}] \) WHEN 'Canada' THEN \((\text{[List Price]} \times 0.60)\) WHEN 'Australia' THEN \(\text{[List Price]} \times 0.80\) ELSE \(\text{[List Price]}\) END

---

**Business Date/Time Functions**
This list contains business functions for performing date and time calculations.

**_add_days**
Returns the date or datetime, depending on the format of "date_expression", that results from adding "integer_expression" days to "date_expression".

**Syntax**
_\_add_days ( date_expression, integer_expression )

**Example 1**
_\_add_days ( 2002-04-30 , 1 )

**Result**
2002-05-01

**Example 2**
_\_add_days ( 2002-04-30 12:10:10.000 , 1 )

**Result**
2002-05-01 12:10:10.000

**Example 3**

---

Appendix F: Using the expression editor 587
_add_days ( 2002-04-30 00:00:00.000, 1/24 )
Note that the second argument is not a whole number. This is supported by some database technologies and increments the time portion.

Result
2002-04-30 01:00:00.000

_add_months
Returns the date or datetime, depending on the format of "date_expression", that results from the addition of "integer_expression" months to "date_expression".

Syntax
_add_months ( date_expression, integer_expression )

Example 1
_add_months ( 2002-04-30 , 1 )

Result
2002-05-30

Example 2
_add_months ( 2002-04-30 12:10:10.000, 1 )

Result
2002-05-30 12:10:10.000

_add_years
Returns the date or datetime, depending on the format of "date_expression", that results from the addition of "integer_expression" years to "date_expression".

Syntax
_add_years ( date_expression, integer_expression )

Example 1
_add_years ( 2002-04-30 , 1 )

Result
2003-04-30

Example 2
_add_years ( 2002-04-30 12:10:10.000 , 1 )

Result
2003-04-30 12:10:10.000
_age
Returns a number that is obtained from subtracting "date_expression" from today's
date. The returned value has the form YYYYMMDD, where YYYY represents the
number of years, MM represents the number of months, and DD represents the
number of days.

Syntax
_age ( date_expression )

Example
_age ( 1990-04-30 ) (if today's date is 2003-02-05)

Result
120906, meaning 12 years, 9 months, and 6 days.

_day_of_week
Returns the day of week (1 to 7), where 1 is the first day of the week as indicated
by the second parameter (1 to 7, 1 being Monday and 7 being Sunday). Note that
in ISO 8601 standard, a week begins with Monday being day 1.

Syntax
_day_of_week ( date_expression, integer )

Example
_day_of_week ( 2003-01-01, 1 )

Result
3

_day_of_year
Returns the day of year (1 to 366) in "date_expression". Also known as Julian day.

Syntax
_day_of_year ( date_expression )

Example
_day_of_year ( 2003-03-01 )

Result
61

_days_between
Returns a positive or negative number representing the number of days between
"date_expression1" and "date_expression2". If "date_expression1" <
"date_expression2", then the result will be a negative number.

Syntax
_days_between ( date_expression1, date_expression2 )

Example
_days_between (2002-04-30, 2002-06-21)

Result

-52

_days_to_end_of_month

Returns a number representing the number of days remaining in the month represented by "date_expression".

Syntax

_days_to_end_of_month (date_expression)

Example

_days_to_end_of_month (2002-04-20 14:30:22.123)

Result

10

_first_of_month

Returns a date or datetime, depending on the argument, by converting "date_expression" to a date with the same year and month but with the day set to 1.

Syntax

_first_of_month (date_expression)

Example 1

_first_of_month (2002-04-20)

Result

2002-04-01

Example 2

_first_of_month (2002-04-20 12:10:10.000)

Result

2002-04-01 12:10:10.000

_last_of_month

Returns a date or datetime, depending on the argument, that is the last day of the month represented by "date_expression".

Syntax

_last_of_month (date_expression)

Example 1

_last_of_month (2002-01-14)

Result

590
Example 2

_last_of_month ( 2002-01-14 12:10:10.000 )

Result

2002-01-31 12:10:10.000

_make_timestamp

Returns a timestamp constructed from "integer_expression1" (the year), "integer_expression2" (the month), and "integer_expression3" (the day). The time portion defaults to 00:00:00.000.

Syntax

_make_timestamp ( integer_expression1, integer_expression2, integer_expression3 )

Example

_make_timestamp ( 2002 , 01 , 14 )

Result

2002-01-14 00:00:00.000

_months_between

Returns a positive or negative integer number representing the number of months between "date_expression1" and "date_expression2". If "date_expression1" is earlier than "date_expression2", then a negative number is returned.

Syntax

_months_between ( date_expression1, date_expression2 )

Example

_months_between ( 2002-04-03 , 2002-01-30 )

Result

2

_week_of_year

Returns the number of the week of the year of "date_expression" according to the ISO 8601 standard. Week 1 of the year is the first week of the year to contain a Thursday, which is equivalent to the first week containing January 4th. A week starts on Monday (day 1) and ends on Sunday (day 7).

Syntax

_week_of_year ( date_expression )

Example

_week_of_year ( 2003-01-01 )

Result
years_between
Returns a positive or negative integer number representing the number of years between "date_expression1" and "date_expression2". If "date_expression1" < "date_expression2" then a negative value is returned.

Syntax
years_between ( date_expression1, date_expression2 )

Example
years_between ( 2003-01-30, 2001-04-03 )

Result
1

ymdint_between
Returns a number representing the difference between "date_expression1" and "date_expression2". The returned value has the form YYYYMMDD, where YYYY represents the number of years, MM represents the number of months, and DD represents the number of days.

Syntax
ymdint_between ( date_expression1, date_expression2 )

Example
ymdint_between ( 1990-04-30, 2003-02-05 )

Result
120906, meaning 12 years, 9 months and 6 days.

Block Functions
This list contains functions used to access members of a set, usually in the context of Analysis Studio.

firstFromSet
Returns the first members found in the set up to "numeric_expression_maximum" + "numeric_expression_overflow". If "numeric_expression_maximum" + "numeric_expression_overflow" is exceeded, then only the maximum number of members are returned. For a set that has only a few members more than the specified numeric_expression_maximum, the numeric_expression_overflow allows the small set of extra members to be included. If the set has more members than the overflow allows, then only the numeric_expression_maximum members will be returned.

Syntax
firstFromSet ( set_expression, numeric_expression_maximum, numeric_expression_overflow )

Example 1
Result

Returns the five members in the Product line set. The first two members are returned within the maximum and the following three members are returned as the overflow.

Result data

Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories

Example 2

_resultFromSet ( [great_outdoors_company].[Products].[Products].[Product line], 2, 8)

Result

Camping Equipment, Golf Equipment

_remainderSet

Returns the set containing "member_expression" when the size of "set_expression" is greater than "numeric_expression"; i.e., a new member will be generated if the number of members in "set_expression" is larger than the specified "numeric_expression".

Syntax

_remainderSet ( member_expression, set_expression, numeric_expression )

Example

_remainderSet ( member ( aggregate ( currentMeasure within set [great_outdoors_company].[Products].[Products].[Product line] ), 'Product Aggregate' ), [great_outdoors_company].[Products].[Products].[Product line], 1 )

Result

Quantity sold for Product Aggregate

Aster Data

Aster Data String

overlay

Returns the "string_exp1" replacing "string_exp2" from character position numeric_exp.

Syntax

overlay ( string_exp1, string_exp2, numeric_exp1 [, numeric_exp2 ] )
**btrim**
Returns string Exp1 after removing the longest string of characters in string Exp2.

**Syntax**
btrim( string Exp1 [, string Exp2] )

**initcap**
Returns "string Exp", with the first letter of each word in uppercase and all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

**Syntax**
initcap ( string Exp )

**lpad**
Returns "string Exp1" padded to length "integer Exp" with occurrences of "string Exp2". If "string Exp1" is longer than "integer Exp", the appropriate portion of "string Exp1" is returned.

**Syntax**
lpad ( string Exp1, integer Exp [, string Exp2 ] )

**ltrim**
Returns "string Exp1", with leading characters removed up to the first character not in "string Exp2", e.g., ltrim('xyxXxyAB', 'xy') returns 'XxyAB'.

**Syntax**
ltrim ( string Exp1 [, string Exp2 ] )

**md5**
Returns the MD5 hash of "string Exp1".

**Syntax**
md5 ( string Exp1 )

**to_hex**
Returns the hexadecimal string representation of "numeric Exp1".

**Syntax**
to_hex ( numeric Exp1 )

**repeat**
Returns the "string Exp" repeated "numeric Exp1" times.

**Syntax**
repeat ( string Exp, numeric Exp1 )

**replace**
Returns "string Exp" having replaced "string Exp2" with "string Exp3".

**Syntax**
replace( string Exp, string Exp2, string Exp3)
rpad
Returns "string_exp1" right-padded to length "integer_exp" with occurrences of "string_exp2". If "string_exp1" is longer than "integer_exp", the appropriate portion of "string_exp1" is returned. If "string_exp2" is not specified, then spaces are used.

Syntax
rpad ( string_exp1, integer_exp [ , string_exp2 ] )

rtrim
Returns "string_exp1", with final characters removed after the last character not in "string_exp2", e.g., rtrim('ABxXyx', 'xy') returns 'ABx'. If "string_exp2" is not specified, the final space characters are removed.

Syntax
rtrim ( string_exp1 [ , string_exp2 ] )

split_part
Returns "numeric_exp" field having split "string_exp1" on "string_exp2".

Syntax
split_part ( string_exp1 , string_exp2 , numeric_exp )

Aster Data Data Type Formatting

to_char
Returns the string representation of "exp" with the format of "string_exp". "Exp" can be either a date value or a numeric value.

Syntax
to_char ( exp , string_exp )

to_date
Converts "string_exp1" to a date value as specified by the format "string_exp2".

Syntax
to_date ( string_exp1 , string_exp2 )

to_number
Converts "string_exp1" to a numeric value as specified by the format "string_exp2".

Syntax
to_number ( string_exp1, string_exp2 )

to_timestamp
Converts "string_exp1" to a timestamp value as specified by the format "string_exp2".

Syntax
to_timestamp ( string_exp1, string_exp2 )

Aster Data Math

log
Returns the base 10 logarithm of "numeric_exp1" or logarithm to the base "numeric_exp2".
Syntax
log ( numeric_exp1[, numeric_exp2] )

**ln**
Returns the natural logarithm of "numeric_exp1".

Syntax
ln ( numeric_exp )

cbrt
Returns the cube root of "numeric_exp1".

Syntax
cbrt ( numeric_exp )

pi
Returns the constant of pi.

Syntax
pi ( )

**Aster Data Trigonometry**

acos
Returns the arccosine of "numeric_exp" in radians. The arccosine is the angle whose cosine is "numeric_exp".

Syntax
acos ( numeric_exp )

asin
Returns the arcsine of "numeric_exp" in radians. The arcsine is the angle whose sine is "numeric_exp".

Syntax
asin ( numeric_exp )

atan
Returns the arctangent of "numeric_exp" in radians. The arctangent is the angle whose tangent is "numeric_exp".

Syntax
atan ( numeric_exp )

atan2
Returns the arctangent of the x and y coordinates specified by "numeric_exp1" and "numeric_exp2", respectively, in radians. The arctangent is the angle whose tangent is "numeric_exp2" / "numeric_exp1".

Syntax
atan2 ( numeric_exp1, numeric_exp2 )

cos
Returns the cosine of "numeric_exp", where "numeric_exp" is an angle expressed in radians.
Syntax
\texttt{cos ( numeric\_exp )}

\textbf{cot}
Returns the cotangent of "numeric\_exp", where "numeric\_exp" is an angle expressed in radians.

Syntax
\texttt{cot ( numeric\_exp )}

\textbf{degrees}
Returns the degrees where "numeric\_exp" is an angle expressed in radians.

Syntax
\texttt{degrees ( numeric\_exp )}

\textbf{radians}
Returns the radians where "numeric\_exp" is an angle expressed in degrees.

Syntax
\texttt{radians ( numeric\_exp )}

\textbf{sin}
Returns the sine of "numeric\_exp", where "numeric\_exp" is an angle expressed in radians.

Syntax
\texttt{sin ( numeric\_exp )}

\textbf{tan}
Returns the tangent of "numeric\_exp", where "numeric\_exp" is an angle expressed in radians.

Syntax
\texttt{tan ( numeric\_exp )}

\textbf{ascii}
Returns a number representing the ASCII code value of the leftmost character of "string\_exp", e.g., ascii('A') is 65.

Syntax
\texttt{ascii ( string\_exp )}

\textbf{chr}
Returns the character that has the ASCII code value specified by "integer\_exp". "Integer\_exp" should be between 0 and 255.

Syntax
\texttt{chr ( integer\_exp )}

\textbf{current\_schema}
Returns the name of the current schema

Syntax
**translate**
Returns "string_exp1", with each occurrence of each character in "string_exp2" replaced by its corresponding character in 'string_exp3'.

**Syntax**
`translate ( string_exp1, string_exp2, string_exp3 )`

**date_trunc**
Returns the timestamp to the specified precision.

**Syntax**
`date_trunc ( string_exp, timestamp_exp)`

**version**
Returns the string value of the database version.

**Syntax**
`version ()`

---

**DB2 Math**

**log**
Returns the natural logarithm of "numeric_expression".

**Syntax**
`log ( numeric_expression )`

**log10**
Returns the base ten logarithm of "numeric_expression".

**Syntax**
`log10 ( numeric_expression )`

**rand**
Generates a random number using "integer_expression" as a seed value.

**Syntax**
`rand ( integer_expression )`

---

**DB2 Trigonometry**

**acos**
Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

**Syntax**
`acos ( numeric_expression )`
**asin**
Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

**Syntax**
```
asin ( numeric_expression )
```

**atan**
Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".

**Syntax**
```
atan ( numeric_expression )
```

**atanh**
Returns the hyperbolic arctangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
```
atanh ( numeric_expression )
```

**atan2**
Returns the arctangent of the x and y coordinates specified by "numeric_expression1" and "numeric_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric_expression2" / "numeric_expression1".

**Syntax**
```
atan2 ( numeric_expression1 , numeric_expression2 )
```

**cos**
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
```
cos ( numeric_expression )
```

**cosh**
Returns the hyperbolic cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
```
cosh ( numeric_expression )
```

**cot**
Returns the cotangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
```
cot ( numeric_expression )
```

**degrees**
Returns "numeric_expression" radians converted to degrees.

**Syntax**
degrees ( numeric_expression )

**sin**
Returns the sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
\[ \text{sin ( numeric_expression )} \]

**sinh**
Returns the hyperbolic sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
\[ \text{sinh ( numeric_expression )} \]

**tan**
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
\[ \text{tan ( numeric_expression )} \]

**tanh**
Returns the hyperbolic tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
\[ \text{tanh ( numeric_expression )} \]

**ascii**
Returns the ASCII code value of the leftmost character of the argument as an integer.

**Syntax**
\[ \text{ascii ( string_expression )} \]

**Example**
\[ \text{ascii ( a )} \]

**Result**
Returns 65, the ASCII code value of "a".

**ceiling**
Returns the smallest integer greater than or equal to "numeric_expression".

**Syntax**
\[ \text{ceiling ( numeric_expression )} \]

**Example**
\[ \text{ceiling ( 0.75 )} \]

**Result**
Returns 0.8.

**char**

Returns a string representation of a date/time value or a decimal number.

**Syntax**

```
char ( expression )
```

**chr**

Returns the character that has the ASCII code value specified by "integer_expression". "Integer_expression" should be between 0 and 255.

**Syntax**

```
chr ( integer_expression )
```

**Example**

```
chr ( 65 )
```

**Result**

Returns a, the character for the ASCII code value of 65.

**concat**

Returns a string that is the result of concatenating "string_expression1" with "string_expression2".

**Syntax**

```
concat ( string_expression1, string_expression2 )
```

**Example**

```
concat ( [Sales target (query)].[Sales staff].[First name], [Sales target (query)].[Sales staff].[Last name] )
```

**Result**

Returns the first name and last name; e.g., Bob Smith.

**date**

Returns a date from a single input value. "Expression" can be a string or integer representation of a date.

**Syntax**

```
date ( expression )
```

**Example**

```
date ( '1998-01-08' )
```

**Result**

Returns 8 January 1998.
**day**

Returns the day of the month (1-31) from "date_expression". "Date_expression" can be a date value or a string representation of a date.

**Syntax**

```plaintext
day ( date_expression )
```

**Example**

```plaintext
day ( '1998-01-08' )
```

**Result**

Returns 8.

---

**dayname**

Returns a character string containing the data source-specific name of the day (for example, Sunday through Saturday or Sun. through Sat. for a data source that uses English, or Sonntag through Samstag for a data source that uses German) for the day portion of "date_expression". "Date_expression" can be a date value or a string representation of a date.

**Syntax**

```plaintext
dayname ( date_expression )
```

**Example**

```plaintext
dayname ( '1998-01-08' )
```

**Result**

Returns Thursday.

---

**dayofweek**

Returns the day of the week in "date_expression" as an integer in the range 1 to 7, where 1 represents Sunday. "date_expression" can be a date value or a string representation of a date.

**Syntax**

```plaintext
dayofweek ( date_expression )
```

**Example**

```plaintext
dayofweek ( '1998-01-08' )
```

**Result**

Returns 5.

---

**dayofweek_iso**

Returns the day of the week in "date_expression" as an integer in the range 1 to 7, where 1 represents Monday. "date_expression" can be a date value or a string representation of a date.

**Syntax**

```plaintext
dayofweek_iso ( date_expression )
```
Example
dayofweek_iso( '1998-01-08' )

Result
Returns 4.

dayofyear
Returns the day of the year in "date_expression" as an integer in the range 1 to 366. 'Date_expression' can be a date value or a string representation of a date.

Syntax
dayofyear( date_expression )

Example
dayofyear( current_date )

Result
Returns the day of the year for the current date; e.g., if it was January 28, the expression would return 28.

days
Returns an integer representation of a date. "Expression" can be a date value or a string representation of a date.

Syntax
days( expression )

dec
Returns the decimal representation of "string-expression" using "precision-integer" and "scale-integer". The "decimal-character" can be used to specify the single-byte character constant used to delimit the decimal digits in "string-expression". The "string-expression" must be formatted as an SQL Integer or Decimal constant.

Syntax

decimal
Returns the decimal representation of "string-expression" using "precision-integer" and "scale-integer". The "decimal-character" can be used to specify the single-byte character constant used to delimit the decimal digits in "string-expression". The "string-expression" must be formatted as an SQL Integer or Decimal constant.

Syntax
decimal (string-expression [, precision-integer [, scale-integer [, decimal-character ]]])

difference

Returns an integer value representing the difference between the values returned by the data source-specific soundex function for "string_expression1" and "string_expression2". The value returned ranges from 0 to 4, with 4 indicating the best match. Note that 4 does not mean that the strings are equal.

Syntax

difference (string_expression1, string_expression2)

Example 1

difference ([Sales target (query)].[Sales staff].[First name],[Sales (query)].[Retailers].[Contact first name])

Result

0

Example 2

difference ([Sales target (query)].[Sales staff].[First name],[Sales target (query)].[Sales staff].[First name])

Result

4

digits

Returns the character string representation of a non-floating point number.

Syntax

digits (numeric_expression)

double

Returns the floating-point representation of an expression. "Expression" can either be a numeric or string expression.

Syntax

double (expression)

event_mon_state

Returns the operational state of a particular state monitor.

Syntax

event_mon_state (string_expression)

float

Returns the floating-point representation of a number.

Syntax

float (numeric_expression)
**hex**

Returns the hexadecimal representation of a value.

**Syntax**

```plaintext
hex ( expression )
```

**hour**

Returns the hour, an integer from 0 (midnight) to 23 (11:00 pm), from "time_expression". "Time_expression" can be a time value or a string representation of a time.

**Syntax**

```plaintext
hour ( time_expression )
```

**Example**

```plaintext
hour ( 01:22:45 )
```

**Result**

Returns 1.

**insert**

Returns a string where "integer_expression2" characters have been deleted from "string_expression1" beginning at "integer_expression1" and where "string_expression2" has been inserted into "string_expression1" at its start. The first character in the string is at position 1.

**Syntax**

```plaintext
insert ( string_expression1, integer_expression1, integer_expression2, string_expression2 )
```

**integer**

Returns the integer representation of an expression. "Expression" can be a numeric value or a string representation of a number.

**Syntax**

```plaintext
integer ( expression )
```

**Example**

```plaintext
integer ( 84.95 )
```

**Result**

85

**int**

Returns the integer representation of an expression. "Expression" can be a numeric value or a string representation of a number.

**Syntax**

```plaintext
int ( expression )
```
**Example**

```java
int ( 84.95 )
```

**Result**

85

**julian_day**

Returns an integer value representing the number of days from January 1, 4712 BC (the start of the Julian date calendar) to the date value specified in "expression". "Expression" can be a date value or a string representation of a date.

**Syntax**

```java
julian_day ( expression )
```

**Example**

```java
julian_day ( '2009-06-29' )
```

**Result**

2455012.22130739595741034

**lcase**

Returns "string_expression" with all uppercase characters shifted to lowercase.

**Syntax**

```java
lcase ( string_expression )
```

**Example**

```java
lcase ( [Sales (query)].[Sales staff].[Last name] )
```

**Result**

Returns last names with no uppercase letters.

**left**

Returns the leftmost "integer_expression" characters of "string_expression".

**Syntax**

```java
left ( string_expression, integer_expression )
```

**Example**

```java
left ( [Sales (query)].[Sales staff].[Last name] , 3 )
```

**Result**

Returns the first three characters of each last name.

**length**

Returns the length of the operand in bytes. Exception: double byte string types return the length in characters.

**Syntax**
length ( expression )

Example
length ( [Sales (query)].[Sales staff].[Record start date] )

Result
Returns 4; dates always return a value of 4.

locate
Returns the starting position of the first occurrence of "string_expression1" within "string_expression2". The search starts at position start "integer_expression" of "string_expression2". The first character in a string is at position 1. If "string_expression1" is not found, zero is returned.

Syntax
locate ( string_expression1, string_expression2 [ ,
    integer_expression ] )

Example
locate ( A, [Sales (query)].[Sales staff].[Last name],2 )

Result
Returns the position of the character A in the last names starting at the second character of the last name.

long_varchar
Returns a long string.

Syntax
long_varchar ( string_expression )

ltrim
Returns "string_expression" with leading spaces removed.

Syntax
ltrim ( string_expression )

Example
ltrim ( [Sales (query)].[Sales staff].[Last name] )

Result
Returns last names with any leading spaces removed.

microsecond
Returns the microsecond (time-unit) part of a value. "Expression" can be a timestamp or a string representation of a timestamp.

Syntax
microsecond ( expression )
Example
microsecond ( 01:45:34.056 )

Result
Returns 056.

**midnight_seconds**
Returns an integer value in the range 0 to 86400 representing the number of seconds between midnight and time value specified in the argument. "Expression" can be a time value, a timestamp or a string representation of a time.

**Syntax**
midnight_seconds ( expression )

**Example**
midnight_seconds ( 01:45:34.056 )

**Result**
Returns 6334.

**minute**
Returns the minute (an integer from 0-59) from "time_expression".
"Time_expression" can be a time value, a timestamp, or a string representation of a time.

**Syntax**
minute ( time_expression )

**Example**
minute ( 01:45:34.056 )

**Result**
Returns 45.

**month**
Returns the month (an integer from 1-12) from "date_expression".

**Syntax**
month ( date_expression )

**Example**
month ( 2005-11-01 )

**Result**
Returns 11.
**monthname**

Returns a character string containing the data source-specific name of the month (for example, January through December or Jan. through Dec. for an English data source, or Januar through Dezember for a German data source) for the month portion of "date_expression".

**Syntax**

\[ \text{monthname ( date_expression )} \]

**Example**

\[ \text{monthname ( 2005-11-01 )} \]

**Result**

November

**quarter**

Returns the quarter in "date_expression" as a number in the range 1 to 4, where 1 represents January 1 through March 31.

**Syntax**

\[ \text{quarter ( date_expression )} \]

**Example**

\[ \text{quarter ( 2005-11-01 )} \]

**Result**

Returns 4.

**radians**

Returns the number of radians converted from "numeric_expression" degrees.

**Syntax**

\[ \text{radians ( numeric_expression )} \]

**repeat**

Returns a string consisting of "string_expression" repeated "integer_expression" times.

**Syntax**

\[ \text{repeat ( string_expression, integer_expression )} \]

**Example**

\[ \text{repeat ( XYZ, 3 )} \]

**Result**

Returns XYZXYZXYZ.
replace
Replaces all occurrences of "string_expression2" in "string_expression1" with "string_expression3".

Syntax
replace ( string_expression1, string_expression2, string_expression3 )

Example
replace ( [Sales (query)].[Sales staff].[Position code], A, a )

Result
Returns position codes with all occurrences of "A" replaced by "a".

right
Returns the rightmost "integer_expression" characters of "string_expression".

Syntax
right ( string_expression, integer_expression )

Example
right ( [Sales (query)].[Sales staff].[Position code], 3 )

Result
Returns the rightmost 3 characters of each position code.

round
Returns "numeric_expression" rounded to "integer_expression" places to the right of the decimal point. If "integer_expression" is negative, "numeric_expression" is rounded to the nearest absolute value "integer_expression" places to the left of the decimal point. Rounding takes place before data formatting is applied.

Syntax
round ( numeric_expression, integer_expression )

Example
round ( 3.14159265, 3 )

Result
Returns 3.142.

rtrim
Returns "string_expression" with trailing spaces removed.

Syntax
rtrim ( string_expression )

Example
rtrim ( [Sales (query)].[Sales staff].[Last name] )

Result
Returns last names with any spaces at the end of the name removed.

**second**

Returns the second (an integer from 0-59) from "time_expression".

**Syntax**

second ( time_expression )

**Example**

second ( 01:45:34.056 )

**Result**

Returns 34.

**sign**

Returns an indicator of the sign of "numeric_expression": +1 if "numeric_expression" is positive, 0 if zero, or -1 if negative.

**Syntax**

sign ( numeric_expression )

**Example**

sign ( [Revenue] )

**Result**

Returns + for positive values and - for negative values.

**smallint**

Returns the small integer representation of a number.

**Syntax**

smallint ( expression )

**soundex**

Returns a 4 character string code obtained by systematically abbreviating words and names in "string_expression" according to phonetics. Can be used to determine if two strings sound the same. For example, does sound-of ('SMITH') = sound-of ('SMYTH').'.

**Syntax**

soundex ( string_expression )

**space**

Returns a string consisting of "integer_expression" spaces.

**Syntax**

space ( integer_expression )

**Example**

space ( 5 )
Result

Returns 5 spaces.

**substr**

Returns the substring of "string_expression" that starts at position
"integer_expression1" for "integer_expression2" characters. The first character in
"string_expression" is at position 1.

**Syntax**

`substr ( string_expression , integer_expression1 [ ,
integer_expression2 ] )`

**Example**

`substr ( [Sales (query)].[Sales staff].[Position code], 3, 5 )`

Result

Returns characters 3 to 7 of the position codes.

**table_name**

Returns an unqualified name of a table or view based on the object name in
"string_expression1" and the schema name given in "string_expression2". It is used
to resolve aliases.

**Syntax**

`table_name ( string_expression1 [ , string_expression2 ] )`

**table_schema**

Returns the schema name portion of the two-part table or view name based on the
object name in "string_expression1" and the schema name in "string_expression2".
It is used to resolve aliases.

**Syntax**

`table_schema ( string_expression1 [ , string_expression2 ] )`

**time**

Returns a time from a value.

**Syntax**

`time ( expression )`

**timestamp**

Returns a timestamp from a value or a pair of values. "Expression1" must represent
a date value, and "expression2" must represent a time value.

**Syntax**

`timestamp ( expression1 [ , expression2 ] )`

**Example**

`timestamp ( 11 November 2005 , 12:00:00.000000 )`
Result

Returns 2005-11-11-12:00:00.000000.

timestamp_iso

Returns a datetime in the ISO format (yyyy-mm-dd hh:mm:ss.nnnnnn) converted from the IBM format (yyyy-mm-dd-hh.mm.ss.nnnnn). If "expression" is a time, it inserts the value of the CURRENT DATE for the date elements and zero for the fractional time element.

Syntax

timestamp_iso ( expression )

Example

timestamp_iso ( 11 November 2005 , 12:00:00.000000 )

Result

Returns 2005-11-11 12:00:00.000000.

timestampdiff

Returns an estimated number of intervals of type "expression1" based on the difference between two timestamps. "Expression2" is the result of subtracting two timestamp types and converting the result to CHAR. Valid values of "expression1" are: 1 Fractions of a second; 2 Seconds; 4 Minutes; 8 Hours; 16 Days; 32 Weeks; 64 Months; 128 Quarters; 256 Years.

Syntax

timestampdiff ( expression1, expression2 )

to_char

Returns the string representation of a timestamp with the format of "string_expression".

Syntax

to_char ( timestamp_expression , string_expression )

translate

Returns "string_expression1" in which characters from "string_expression3" are translated to the equivalent characters in "string_expression2". "String_expression4" is a single character that is used to pad "string_expression2" if it is shorter than "string_expression3". If only "string_expression1" is present, then this function translates it to uppercase characters.

Syntax

translate ( string_expression1 [ , string_expression2, string_expression3 [ , string_expression4 ] ] )

Example 1

translate ( 'abcdefg' )

Result
Returns ABCDEFG.

Example 2
translate ( 'mnlop' , n , m , - )

Result
Returns n-nlop.

trunc

Returns "numeric_expression1" truncated to "numeric_expression2" places to the right of the decimal point. If "numeric_expression2" is negative, "numeric_expression1" is truncated to the absolute value of "numeric_expression2" places to the left of the decimal point.

Syntax
trunc ( numeric_expression1 , numeric_expression2 )

Example
trunc ( 3.14159265 , 3 )

Result
Returns 3.141.

truncate

Returns "numeric_expression1" truncated to "numeric_expression2" places to the right of the decimal point. If "numeric_expression2" is negative, "numeric_expression1" is truncated to the absolute value of "numeric_expression2" places to the left of the decimal point.

Syntax
truncate ( numeric_expression1 , numeric_expression2 )

Example
truncate ( 3141.59265 , -3 )

Result
Returns 3.

ucase

Returns "string_expression" with all lowercase characters shifted to uppercase.

Syntax
ucase ( string_expression )

Example
ucase ( XY8962bcd789 )

Result
Returns XY896ZBCED789.
value
Returns the first non-null argument (or null if all arguments are null). The Value function takes two or more arguments.

Syntax
value ( expression_list )

Example
value ( [Unit cost], [Unit price], [Unit sale price] )

Result
Returns the first non-null value.

varchar
Returns a VARCHAR representation of expression, with length numeric_expression.

Syntax
varchar ( expression [ , numeric_expression ] )

week
Returns the week of the year in "date_expression" as an integer value in the range 1 to 53.

Syntax
week ( date_expression )

Example
week ( 11 November 2005 )

Result
Returns 45.

year
Returns the year from "date_expression".

Syntax
year ( date_expression )

Example
year ( 11 November 2005 )

Result
Returns 2005.
Greenplum String

**overlay**
Returns the "string_expression1" replacing "string_expression2" from character position "numeric_expression".

**Syntax**
```
overlay ( string_expression1 , string_expression2 , numeric_expression1 [ , numeric_expression2 ] )
```

**btrim**
Returns "string_expression1" after removing the longest string of characters in "string_expression2".

**Syntax**
```
btrim ( string_expression1 [ , string_expression2 ] )
```

**initcap**
Returns "string_expression" with the first letter of each word in uppercase and all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

**Syntax**
```
initcap ( string_expression )
```

**lpad**
Returns "string_expression1" padded to length "integer_expression" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression", the appropriate portion of "string_expression1" is returned.

**Syntax**
```
lpad ( string_expression1 , integer_expression [ , string_expression2 ] )
```

**ltrim**
Returns "string_expression1", with leading characters removed up to the first character not in "string_expression2"; for example, ltrim ( 'xyxXxyAB' , 'xy' ) returns XxyAB.

**Syntax**
```
ltrim ( string_expression1 [ , string_expression2 ] )
```

**md5**
Returns the MD5 hash of "string_expression1".

**Syntax**
```
md5 ( string_expression1 )
```

**to_hex**
Returns the hexadecimal string representation of "numeric_expression1".

**Syntax**
```
to_hex ( numeric_expression1 )
```
**repeat**
Returns the "string_expression" repeated "numeric_expression1" times.

**Syntax**
repeat ( string_expression , numeric_expression1 )

**replace**
Returns "string_expression" having replaced "string_expression2" with "string_expression3".

**Syntax**
replace ( string_expression , string_expression2 , string_expression3)

**rpad**
Returns "string_expression1" right-padded to length "integer_expression" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression", the appropriate portion of "string_expression1" is returned. If "string_expression2" is not specified, then spaces are used.

**Syntax**
rpad ( string_expression1 , integer_expression [ , string_expression2 ])

**rtrim**
Returns "string_expression1", with final characters removed after the last character not in 'string_expression2'; for example, rtrim ('ABxXxyx', 'xy') returns ABxX. If "string_expression2" is not specified, the final space characters are removed.

**Syntax**
rtrim ( string_expression1 [ , string_expression2 ])

**split_part**
Returns "numeric_expression" field having split "string_expression1" on "string_expression2".

**Syntax**
split_part ( string_expression1 , string_expression2 , numeric_expression )

**Greenplum Data type formatting**

**to_char**
Returns the string representation of "expression" with the format of "string_expression". "Expression" can either be a date value or a numeric value.

**Syntax**
to_char ( expression , string_expression )

**to_date**
Converts "string_expression1" to a date value as specified by the format "string_expression2".

**Syntax**
to_date ( string_expression1 , string_expression2 )
**to_number**
Converts "string_expression1" to a numeric value as specified by the format "string_expression2".

**Syntax**
```
to_number ( string_expression1 , string_expression2 )
```

**to_timestamp**
Converts "string_expression1" to a timestamp value as specified by the format "string_expression2".

**Syntax**
```
to_timestamp ( string_expression1 , string_expression2 )
```

---

**Greenplum Math**

**log**
Returns the base 10 logarithm of "numeric_expression1" or logarithm to the base "numeric_expression2".

**Syntax**
```
log ( numeric_expression1 [ , numeric_expression2 ] )
```

**ln**
Returns the natural logarithm of "numeric_expression1".

**Syntax**
```
ln ( numeric_expression )
```

**cbrt**
Returns the cube root of "numeric_expression1".

**Syntax**
```
cbrt ( numeric_expression )
```

**pi**
Returns the constant of pi.

**Syntax**
```
pi ()
```

---

**Greenplum Trigonometry**

**acos**
Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

**Syntax**
```
acos ( numeric_expression )
```

**asin**
Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

**Syntax**
```
asin ( numeric_expression )
```
asin ( numeric_expression )

atan
Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".

Syntax
atan ( numeric_expression )

atan2
Returns the arctangent of the x and y coordinates specified by "numeric_expression1" and "numeric_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric_expression2" / "numeric_expression1".

Syntax
atan2 ( numeric_expression1 , numeric_expression2 )

cos
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
cos ( numeric_expression )

cot
Returns the cotangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
cot ( numeric_expression )

degrees
Returns the degrees where "numeric_expression" is an angle expressed in radians.

Syntax
degrees ( numeric_expression )
radians
Returns the radians where "numeric_expression" is an angle expressed in degrees.

Syntax
radians ( numeric_expression )

sin
Returns the sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
sin ( numeric_expression )

tan
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
tan ( numeric_expression )

**ascii**

Returns a number representing the ascii code value of the leftmost character of "string_expression"; for example, ascii ('A') is 65.

**Syntax**

```sql
ascii ( string_expression )
```

**chr**

Returns the character that has the ASCII code value specified by "integer_expression". "Integer_expression" should be between 0 and 255.

**Syntax**

```sql
chr ( integer_expression )
```

**current_database**

Returns the name of the current database.

**Syntax**

```sql
current_database ()
```

**current_schema**

Returns the name of the current schema.

**Syntax**

```sql
current_schema ()
```

**{current_user}**

**Syntax**

```sql
{current_user}
```

**{session_user}**

**Syntax**

```sql
{session_user}
```

**translate**

Returns "string_expression1" with each occurrence of each character in "string_expression2" replaced by its corresponding character in "string_expression3".

**Syntax**

```sql
translate ( string_expression1 , string_expression2 , string_expression3 )
```

**date_trunc**

Returns the timestamp to the specified precision.

**Syntax**

```sql
date_trunc ( string_expression , timestamp_expression)
```
version
  Returns the string value of the database version.
  Syntax
  version ()

Informix

Informix Math

log10
  Returns the logarithm of "numeric_expression" to base 10.
  Syntax
  log10 ( numeric_expression )

logn
  Returns the natural logarithm of "numeric_expression".
  Syntax
  logn ( numeric_expression )

root
  Returns the root value of "numeric_expression1". Requires at least one numeric argument (the radians argument). If only "numeric_expression1" is supplied, 2 is used as a default value for "numeric_expression2". Zero cannot be used as the value of "numeric_expression2".
  Syntax
  root ( numeric_expression1 [ , numeric_expression2 ] )

Informix Trigonometry

acos
  Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".
  Syntax
  acos ( numeric_expression )

asin
  Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".
  Syntax
  asin ( numeric_expression )

atan
  Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".
  Syntax
  atan ( numeric_expression )
atan2
Returns the arctangent of the x and y coordinates specified by
"numeric_expression1" and "numeric_expression2", respectively, in radians. The
arctangent is the angle whose tangent is "numeric_expression1".

Syntax
atan2 ( numeric_expression1 , numeric_expression2 )

cos
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle
epressed in radians.

Syntax
cos ( numeric_expression )

sin
Returns the sine of "numeric_expression" where "numeric_expression" is an angle
expressed in radians.

Syntax
sin ( numeric_expression )

tan
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle
expressed in radians.

Syntax
tan ( numeric_expression )

cardinality
Returns the number of elements in a collection column (SET, MULTISET, LIST).

Syntax
cardinality ( string_expression )

char_length
Returns the number of logical characters in "string_expression". The number of
logical characters can be distinct from the number of bytes in some East Asian
locales.

Syntax
char_length ( string_expression )

concat
Returns a string that is the result of concatenating, or joining, "string_expression1" to "string_expression2".

Syntax
concat ( string_expression1 , string_expression2 )

Example
concat ( [Sales (query)].[Sales staff].[First name], [Sales (query)].
[Sales staff].[Last name] )
Result

Returns the first name and last name; e.g., Bob Smith.

date

Returns the date value of "string_expression", "date_expression", or "integer_expression".

Syntax

date ( string_expression|date_expression|integer_expression )

day

Returns an integer that represents the day of the month (1-31).

Syntax

day ( date_expression )

extend

Adjusts the precision of a datetime or date expression. The expression cannot be a quoted string representation of a date value. If you do not specify first and last qualifiers, the default qualifiers are year to fraction (3). If the expression contains fields that are not specified by the qualifiers, the unwanted fields are discarded. If the first qualifier specifies a larger (more significant) field than what exists in the expression, the new fields are filled in with values returned by the current function. If the last qualifier specifies a smaller (less significant) field than what exists in the expression, the new fields are filled in with constant values. A missing month or day field is filled in with 1, and missing hour to fraction fields are filled in with 0.

Syntax

extend ( date_expression,'{ year to second}' )

Example

extend ( some_date_column , { year to second } )

hex

Returns the hexadecimal encoding of "integer_expression".

Syntax

hex ( integer_expression )

initcap

Returns "string_expression" with the first letter of each word in uppercase and all other letters in lowercase. A word begins after any character other than a letter. Thus, in addition to a blank space, symbols such as commas, periods, and colons can introduce a new word.

Syntax

initcap ( string_expression )
**length**

Returns the number of bytes in "string_expression", not including any trailing blank spaces. For byte or text "string_expression", length returns the full number of bytes, including any trailing blank spaces.

**Syntax**

```
length ( string_expression )
```

**lpad**

Returns "string_expression1" left-padded by "string_expression2" to the total number of characters specified by "integer_expression". The sequence of "string_expression2" occurs as many times as necessary to make the return string the length specified by "integer_expression".

**Syntax**

```
lpad ( string_expression1 , integer_expression , string_expression2 )
```

**mdy**

Returns a type date value with three expressions that evaluate to integers that represent the month (integer_expression1), day (integer_expression2), and year (integer_expression3).

**Syntax**

```
mdy ( integer_expression1 , integer_expression2 ,
      integer_expression3 )
```

**month**

Returns an integer corresponding to the month portion of "date_expression".

**Syntax**

```
month ( date_expression )
```

**nvl**

Returns the value of "expression1" if "expression1" is not NULL. If "expression1" is NULL, then returns the value of "expression2".

**Syntax**

```
nvl ( expression1 , expression2 )
```

**Example**

```
nvl ( [Unit sale price] , [Unit price] )
```

**Result**

Returns the unit sale price, or returns the unit price if the unit sale price is NULL.

**octet_length**

Returns the number of bytes in "string_expression", including any trailing spaces.

**Syntax**

```
octet_length ( string_expression )
```
replace

Returns "string_expression1" in which every occurrence of "string_expression2" is replaced by "string_expression3". If you omit the "string_expression3" option, every occurrence of "string_expression2" is omitted from the return string.

Syntax
replace ( string_expression1 , string_expression2 [ , string_expression3 ] )

Example
replace ( [Sales (query)].[Products].[Product line code] , - )

Result
Returns all product line codes without the character "-".

round

Returns the rounded value of "numeric_expression". If you omit "integer_expression", the value is rounded to zero digits or to the units place. The digit range of 32 (+ and -) refers to the entire decimal value. Rounding takes place before data formatting is applied.

Syntax
round ( numeric_expression [ , integer_expression ] )

Example
round (125, -1)

Result
130

rpad

Returns "string_expression1" right-padded by "string_expression2" to the total number of characters specified by "integer_expression". The sequence of "string_expression2" occurs as many times as necessary to make the return string the length specified by "integer_expression".

Syntax
rpad ( string_expression1 , integer_expression , string_expression2 )

substr

Returns the substring of "string_expression" that starts at position "integer_expression1" for "integer_expression2" characters. The first character in "string_expression" is at position 1. If you omit "integer_expression2", returns the substring of "string_expression" that starts at position "integer_expression1" and ends at the end of "string_expression".

Syntax
substr ( string_expression , integer_expression1 [ , integer_expression2 ] )

Example
substr ( [Sales (query)].[Sales staff].[Position code], 3, 5 )

Result

Returns characters 3 to 7 of the position codes.

to_char

Returns the character string "date_expression" with the specified "string_expression" formatting. You can use this function only with built-in data types.

Syntax

to_char ( date_expression , string_expression )

to_date

Returns "string_expression1" as a date according to the date format you specify in "string_expression2". If "string_expression1" is NULL, then a NULL value is returned.

Syntax

to_date ( string_expression1 , string_expression2 )

trunc

Returns the truncated value of "numeric_expression". If you omit "integer_expression", then "numeric_expression" is truncated to zero digits or to the unit's place. The digit limitation of 32 (+ and -) refers to the entire decimal value.

Syntax

trunc ( numeric_expression [ , integer_expression ] )

weekday

Returns an integer that represents the day of the week of "date_expression". Zero (0) represents Sunday, one (1) represents Monday, and so on.

Syntax

weekday ( date_expression )

year

Returns a four-digit integer that represents the year of "date_expression".

Syntax

year ( date_expression )

---

**MS Access**

**MS Access Cast**

**cast_decimal**

Returns the value of "expression" cast as a decimal.

Syntax

cast_decimal ( expression )
**cast_float**  
Returns the value of "expression" cast as a float.

**Syntax**  
cast_float ( expression )

**cast_integer**  
Returns the value of "expression" cast as an integer.

**Syntax**  
cast_integer ( expression )

**Example**  
cast_integer ( 84.95 )

**Result**  
84

**cast_numeric**  
Returns "string_expression" cast as a numeric value.

**Syntax**  
cast_numeric ( string_expression )

**cast_real**  
Returns the value of "expression" cast as a real value.

**Syntax**  
cast_real ( expression )

**cast_smallint**  
Returns "expression" cast as a small integer.

**Syntax**  
cast_smallint ( expression )

**cast_varchar**  
Returns the value of "expression" cast as a variable character field.

**Syntax**  
cast_varchar ( expression )

---

**MS Access Math**

**log**  
Returns the natural logarithm of "numeric_expression".

**Syntax**  
log ( numeric_expression )

**rand**  
Generates a random number using "integer_expression" as a seed value.

**Syntax**
rand ( integer_expression )

MS Access Trigonometry

atan
Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".

Syntax
atan ( numeric_expression )

cos
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
cos ( numeric_expression )

sin
Returns the sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
sin ( numeric_expression )

tan
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
tan ( numeric_expression )

ascii
Returns the ascii code value of the leftmost character of "string_expression".

Syntax
ascii ( string_expression )

ceiling
Returns the smallest integer greater than or equal to "numeric_expression".

Syntax
ceiling ( numeric_expression )

chr
Returns the character that has the ASCII code value specified by "integer_expression". "Integer_expression" should be between 0 and 255.

Syntax
chr ( integer_expression )
**concat**

Returns a string that is the result of concatenating, or joining, "string_expression1" to "string_expression2".

**Syntax**

\[
\text{concat} \left( \text{string_expression1}, \text{string_expression2} \right)
\]

**Example**

\[
\text{concat} \left( \text{[Sales (query)].[Sales staff].[First name]}, \text{[Sales (query)].[Sales staff].[Last name]} \right)
\]

**Result**

Returns the first name and last name; e.g., Bob Smith.

**curdate**

Returns a date value representing the current date of the computer that the database software runs on.

**Syntax**

\[
\text{curdate}()
\]

**curtime**

Returns a time value representing the current time of the computer that the database software runs on.

**Syntax**

\[
\text{curtime}()
\]

**dayname**

Returns a character string containing the data source-specific name of the day (for example, Sunday through Saturday or Sun. through Sat. for an English data source, or Sonntag through Samstag for a German data source) for the day portion of "date_expression".

**Syntax**

\[
\text{dayname} \left( \text{date_expression} \right)
\]

**dayofmonth**

Returns the day of the month (1-31) from "date_expression". Returns the days field (a signed integer) from "interval_expression".

**Syntax**

\[
\text{dayofmonth} \left( \text{date_expression} | \text{interval_expression} \right)
\]

**dayofweek**

Returns the day of the week in "date_expression" as an integer (1-7), where 1 represents Monday.

**Syntax**

\[
\text{dayofweek} \left( \text{date_expression} \right)
\]
dayofyear
Returns the day of the year in "date_expression" as an integer (1-366).

Syntax
dayofyear ( date_expression )

hour
Returns the hour from "time_expression" as an integer from 0 (midnight) to 23 (11:00 pm).

Syntax
hour ( time_expression )

instr
Searches "string_expression1" for the first occurrence of "string_expression2" and returns an integer specifying the position of "string_expression2". "Integer_expression1" sets the starting position for the search. If "integer_expression1" is omitted, the search begins at the first character position of "string_expression1". "Integer_expression2" specifies the type of string comparison. "Integer_expression1" is required if "integer_expression2" is specified.

Syntax
instr ( [ integer_expression1 , ] string_expression1 , string_expression2 [ , integer_expression2 ] )

lcase
Returns "string_expression" with all uppercase characters converted to lowercase.

Syntax
lcase ( string_expression )

left
Returns the leftmost "integer_expression" characters of "string_expression".

Syntax
left ( string_expression , integer_expression )

Example
left ( [Sales (query)].[Sales staff].[Last name] , 3 )

Result
Returns the first three characters of each last name.

length
Returns the number of characters in "string_expression", excluding trailing blanks and the string termination character.

Syntax
length ( string_expression )
locate

Returns the starting position of the first occurrence of "string_expression1" within "string_expression2". The search starts at position "integer_expression" of "string_expression2". The first character in a string is at position 1. If "string_expression1" is not found, then zero is returned.

Syntax
locate ( string_expression1 , string_expression2 [ , integer_expression ] )

ltrim

Returns "string_expression" with leading spaces removed.

Syntax
ltrim ( string_expression )

minute

Returns the minute (an integer from 0-59) from "time_expression".

Syntax
minute ( time_expression )

month

Returns the month (an integer from 1-12) from "date_expression".

Syntax
month ( date_expression )

monthname

Returns a character string containing the data source-specific name of the month (for example, January through December or Jan. through Dec. for an English data source, or Januar through Dezember for a German data source) for the month portion of "date_expression".

Syntax
monthname ( date_expression )

Example
monthname ( 2005-11-01 )

Result
November

now

Returns a datetime value representing the current date and time of the computer that the database software runs on.

Syntax
now ()
**position**

Returns the starting position of "string_expression1" in "string_expression2". The first character in a string is at position 1.

**Syntax**

position ( string_expression1 , string_expression2 )

**quarter**

Returns the quarter in "date_expression" as a number (1-4), where 1 represents January 1 through March 31.

**Syntax**

quarter ( date_expression )

**right**

Returns the rightmost "integer_expression" characters of "string_expression".

**Syntax**

right ( string_expression , integer_expression )

**round**

Returns "numeric_expression" rounded to the nearest value "integer_expression" places right of the decimal point. If "integer_expression" is negative, "numeric_expression" is rounded to the nearest absolute value "integer_expression" places to the left of the decimal point. Rounding takes place before data formatting is applied.

**Syntax**

round ( numeric_expression , integer_expression )

**Example**

round (125, -1)

**Result**

130

**rtrim**

Returns "string_expression" with trailing spaces removed.

**Syntax**

rtrim ( string_expression )

**Example**

rtrim ( [Sales (query)].[Sales staff].[Last name] )

**Result**

Returns last names with any spaces at the end of the name removed.
sign

Returns an indicator of the sign of "numeric_expression", +1 if positive, 0 if zero, or -1 if negative.

Syntax

sign ( numeric_expression )

space

Returns a string consisting of "integer_expression" spaces.

Syntax

space ( integer_expression )

substr

Returns the substring of "string_expression" that starts at position "integer_expression1" for "integer_expression2" characters. The first character in "string_expression" is at position 1.

Syntax

substr ( string_expression , integer_expression1 , integer_expression2 )

Example

substr ( [Sales (query)].[Sales staff].[Position code] , 3 , 5 )

Result

Returns characters 3 to 7 of the position codes.

substring

Returns the substring of "string_expression" that starts at position "integer_expression1" for "integer_expression2" characters. The first character in "string_expression" is at position 1.

Syntax

substring ( string_expression , integer_expression1 , integer_expression2 )

Example

substring ( [Sales (query)].[Sales staff].[Position code] , 3 , 5 )

Result

Returns characters 3 to 7 of the position codes.

truncate

Returns "string_expression" with trailing spaces removed.

Syntax

truncate ( string_expression )
ucase
Returns "string_expression" with all lowercase characters converted to uppercase.

Syntax
ucase ( string_expression )

week
Returns the week of the year in "date_expression" as an integer value (1-53), where 1 represents the first week of the year.

Syntax
week ( date_expression )

year
Returns the year from "date_expression".

Syntax
year ( date_expression )

MySQL

MySQL String

lpad
Returns "string_expression1" padded to length "integer_expression1" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression1", the appropriate portion of "string_expression1" is returned.

Syntax
lpad ( string_expression1 , integer_expression [ , string_expression2 ] )

ltrim
Returns "string_expression1", with leading characters removed up to the first character not in "string_expression2"; for example, ltrim ( 'xyXxyAB' , 'xy' ) returns XxyAB.

Syntax
ltrim ( string_expression1 [ , string_expression2 ] )

hex
Returns the hexadecimal string representation of "numeric_expression1".

Syntax
hex ( numeric_expression1 )

repeat
Returns the "string_expression" repeated "numeric_expression1" times.

Syntax
repeat ( string_expression , numeric_expression1 )
**replace**
Returns "string_expression" having replaced "string_expression2" with "string_expression3".

**Syntax**
replace ( string_expression , string_expression2 , string_expression3 )

**reverse**
Returns "string_expression" reversed.

**Syntax**
reverse ( string_expression )

**right**
Returns the rightmost "numeric_expression" characters from "string_expression1".

**Syntax**
right ( string_expression1 , numeric_expression )

**rpad**
Returns "string_expression1" right-padded to length "integer_expression" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression", the appropriate portion of "string_expression1" is returned. If "string_expression2" is not specified, then spaces are used.

**Syntax**
rpad ( string_expression1 , integer_expression [ , string_expression2 ] )

**rtrim**
Returns "string_expression1", with final characters removed after the last character not in 'string_expression2'; for example, rtrim ( 'ABxXyx' , 'xy' ) returns ABxX. If "string_expression2" is not specified, the final space characters are removed.

**Syntax**
rtrim ( string_expression1 [ , string_expression2 ] )

**soundex**
Returns a soundex string of "string_expression1".

**Syntax**
soundex ( string_expression1 )

**MySQL Math**

**log**
Returns the base 10 logarithm of "numeric_expression1" or logarithm to the base "numeric_expression2".

**Syntax**
log ( numeric_expression )
**In**

Returns the natural logarithm of "numeric_expression1".

**Syntax**

```sql
in ( numeric_expression )
```

**pi**

Returns the constant of pi.

**Syntax**

```sql
pi ()
```

## MySQL Trigonometry

**acos**

Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

**Syntax**

```sql
acos ( numeric_expression )
```

**asin**

Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

**Syntax**

```sql
asin ( numeric_expression )
```

**atan**

Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".

**Syntax**

```sql
atan ( numeric_expression )
```

**atan2**

Returns the arctangent of the x and y coordinates specified by "numeric_expression1" and "numeric_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric_expression2" / "numeric_expression1".

**Syntax**

```sql
atan2 ( numeric_expression1 ,numeric_expression2 )
```

**cos**

Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**

```sql
cos ( numeric_expression )
```

**cot**

Returns the cotangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.
Syntax
cot ( numeric_expression )

degrees
Returns the degrees where "numeric_expression" is an angle expressed in radians.

Syntax
degrees ( numeric_expression )

radians
Returns the radians where "numeric_expression" is an angle expressed in degrees.

Syntax
radians ( numeric_expression )

sin
Returns the sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
sin ( numeric_expression )

tan
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
tan ( numeric_expression )

ascii
Returns a number representing the ASCII code value of the leftmost character of "string_expression"; for example, ascii ( 'A' ) is 65.

Syntax
ascii ( string_expression )

database
Returns the current database name

Syntax
database ( )

schema
Returns the current schema name

Syntax
schema ( )

session_user
Return the user name returned by the client

Syntax
session_user ( )
**system_user**

Return the user name returned by the client

**Syntax**

```sql
system_user()
```

**version**

Returns the string value of the database version.

**Syntax**

```sql
version()
```

---

### Netezza

#### Netezza Math

**log**

Returns the logarithm of "numeric_expression2" to the base "numeric_expression1". When the optional argument "numeric_expression1" is not specified, the base 10 is used.

**Syntax**

```sql
log([numeric_expression1], numeric_expression2)
```

#### Netezza Trigonometry

**acos**

Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

**Syntax**

```sql
acos(numeric_expression)
```

**asin**

Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

**Syntax**

```sql
asin(numeric_expression)
```

**atan**

Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".

**Syntax**

```sql
atan(numeric_expression)
```

**atan2**

Returns the arctangent of the x and y coordinates specified by "numeric_expression1" and "numeric_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric_expression2" / "numeric_expression1".
**Syntax**
atan2 ( numeric_expression1 , numeric_expression2 )

**cos**
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
cos ( numeric_expression )

**degrees**
Returns the degrees where "numeric_expression" is an angle expressed in radians.

**Syntax**
degrees ( numeric_expression )

**radians**
Returns the radians where "numeric_expression" is an angle expressed in degrees.

**Syntax**
radians ( numeric_expression )

**sin**
Returns the sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
sin ( numeric_expression )

**tan**
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
tan ( numeric_expression )

**Netezza Fuzzy**

**le_dst**
Returns a value indicating how different the two input strings are, calculated according to the Levenshtein edit distance algorithm.

**Syntax**
le_dst ( string_expression1 , string_expression2 )

**dle_dst**
Returns a value indicating how different the two input strings are, calculated according to the Damerau-Levenshtein distance algorithm.

**Syntax**
dle_dst ( string_expression1 , string_expression2 )
Netezza Phonetic

**nysiis**
Returns a Soundex representation of "string_expression" using the New York State Identification and Intelligence System (NYSIIS) variation of Soundex.

**Syntax**
nysiis ( string_expression )

**dbl_mp**
Returns a composite 32-bit value of "string_expression".

**Syntax**
dbl_mp ( string_expression )

**pri_mp**
Returns the 4 character primary metaphone string from "numeric_expression" returned by dbl_mp.

**Syntax**
pri_mp ( numeric_expression )

**sec_mp**
Returns the 4 character secondary metaphone string from "numeric_expression" returned by dbl_mp.

**Syntax**
sec_mp ( numeric_expression )

**score_mp**
Returns a score for how closely "numeric_expression" and "numeric_expression2" match.

**Syntax**
score_mp ( numeric_expression , numeric_expression2 , numeric_expression3 , numeric_expression4 , numeric_expression5 , numeric_expression6 )

**ascii**
Returns a number representing the ASCII code value of the leftmost character of "string_expression"; for example, ascii ( 'A' ) is 65.

**Syntax**
ascii ( string_expression )

**chr**
Returns the character that has the ASCII code value specified by "integer_expression". "Integer_expression" should be between 0 and 255.

**Syntax**
chr ( integer_expression )
decode

Compared “expr” to each search value one by one. If “expr” is equal to a search, then it returns the corresponding result. If no match is found, it returns “default”. If “default” is omitted, it returns null.

Syntax

decode ( expr , search , result [ , search , result]... [ , default] )

initcap

Returns "string_expression", with the first letter of each word in uppercase, all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

Syntax

initcap ( string_expression )

instr

Searches "string_expression1" starting at position "integer_expression1" for the "integer_expression2" occurrence of "string_expression2". If "integer_expression1" is negative then the search is backwards from the end of "string_expression1".
Returns an integer indicating the position of "string_expression2".

Syntax

instr ( string_expression1 , string_expression2 [ , integer_expression1 [ , integer_expression2 ] ] )

lpad

Returns "string_expression1" padded to length "integer_expression" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression", the appropriate portion of "string_expression1" is returned.

Syntax

lpad ( string_expression1 , integer_expression [ , string_expression2 ] )

ltrim

Returns "string_expression1", with leading characters removed up to the first character not in "string_expression2"; for example, ltrim ( ‘xyxXyAB’, ‘xy’ ) returns XxyAB.

Syntax

ltrim ( string_expression1 [ , string_expression2 ] )

months_between

Returns the number of months from "date_expression1" to "date_expression2". If "date_expression1" is later than "date_expression2" then the result will be a positive number. The days and time portions of the difference are ignored, i.e., the months are not rounded, except if "date_expression1" and "date_expression2" are the last days of a month.

Syntax

months_between ( date_expression1 , date_expression2 )
**next_day**

Returns the datetime of the first weekday named by "string_expression" that is later than "datetime_expression". The return value has the same hours, minutes, and seconds as "datetime_expression".

Syntax
next_day ( datetime_expression , string_expression )

**nvl**

Returns the value of "expression1" if "expression1" is not NULL. If "expression1" is NULL, then returns the value of "expression2".

Syntax
nvl ( expression1 , expression2 )

Example
nvl ( [Unit sale price] , 0 )

Result
Returns the unit sale price, or returns 0 if the unit sale price is NULL.

**round**

Returns "numeric_expression" rounded to the nearest value "integer_expression" places right of the decimal point. If "integer_expression" is negative, "numeric_expression" is rounded to the nearest absolute value "integer_expression" places to the left of the decimal point; for example, round (125, -1) rounds to 130.

Syntax
round ( numeric_expression [ , integer_expression ] )

**rpad**

Returns "string_expression1" right-padded to length "integer_expression" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression", the appropriate portion of "string_expression1" is returned. If "string_expression2" is not specified, then spaces are used.

Syntax
rpad ( string_expression1 , integer_expression [ , string_expression2 ] )

**rtrim**

Returns "string_expression1", with final characters removed after the last character not in "string_expression2"; for example, rtrim ( 'ABxXyyx', 'xy' ) returns ABX. If "string_expression2" is not specified, the final space characters are removed.

Syntax
rtrim ( string_expression1 [ , string_expression2 ] )
**substr**

Returns the substring of "string_expression" that starts at position "integer_expression1". The first character in "string_expression" is at position 1. "Integer_expression2" can be used to select fewer characters; by default it selects characters to the end of the string.

**Syntax**

```
substr ( string_expression , integer_expression1 [ , integer_expression2 ] )
```

**to_char**

Returns the string representation of "expression" with the format of "string_expression". "Expression" can be either a date value or a numeric value.

**Syntax**

```
to_char ( expression [ , string_expression ] )
```

**to_date**

Converts "string_expression1" to a datetime value as specified by the format "string_expression2".

**Syntax**

```
to_date ( string_expression1 , string_expression2 )
```

**to_number**

Converts "string_expression1" to a numeric value as specified by the format "string_expression2".

**Syntax**

```
to_number ( string_expression1 , string_expression2 )
```

**translate**

Returns "string_expression1", with all occurrences of each character in "string_expression2" replaced by its corresponding character in "string_expression3".

**Syntax**

```
translate ( string_expression1 , string_expression2 , string_expression3 )
```
**date_trunc**
Truncates "date_expression1" to a value as specified by the format "string_expression1".

Syntax

date_trunc ( string_expression1 , date_expression1 )

**trunc**
Truncates digits from "numeric_expression1" using "numeric_expression2" as the precision.

Syntax

trunc ( numeric_expression1 [ , numeric_expression2 ] )

**version**
Returns the "string_expression1" value of the database version.

Syntax

version ()

---

**Oracle**

**Oracle Math**

**log**
Returns the logarithm of "numeric_expression2" to the base "numeric_expression1". When the optional argument "numeric_expression1" is not specified, the base used is the constant e (which is approximately equal to 2.71282).

Syntax

log ( [ numeric_expression1 , ] numeric_expression2 )

**Oracle Trigonometry**

**acos**
Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

Syntax

acos ( numeric_expression )

**asin**
Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

Syntax

asin ( numeric_expression )

**atan**
Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".
Syntax
atan ( numeric_expression )

atan2
Returns the arctangent of the x and y coordinates specified by
"numeric_expression1" and "numeric_expression2", respectively, in radians. The
arctangent is the angle whose tangent is "numeric_expression2" / 
"numeric_expression1".

Syntax
atan2 ( numeric_expression1 , numeric_expression2 )

cos
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle
expressed in radians.

Syntax
cos ( numeric_expression )

cosh
Returns the hyperbolic cosine of "numeric_expression" where "numeric_expression" is
an angle expressed in radians.

Syntax
cosh ( numeric_expression )

sin
Returns the sine of "numeric_expression" where "numeric_expression" is an angle
expressed in radians.

Syntax
sin ( numeric_expression )

sinh
Returns the hyperbolic sine of "numeric_expression" where "numeric_expression" is
an angle expressed in radians.

Syntax
sinh ( numeric_expression )

tan
Returns the tangent of "numeric_expression" where "numeric_expression" is an
angle expressed in radians.

Syntax
tan ( numeric_expression )

tanh
Returns the hyperbolic tangent of "numeric_expression" where
"numeric_expression" is an angle expressed in radians.

Syntax
tanh ( numeric_expression )
add_months

Returns the datetime resulting from adding "integer_expression" months to "date_expression".

Syntax
add_months ( date_expression , integer_expression )

ascii

Returns a number representing the ASCII code value of the leftmost character of "string_expression".

Syntax
ascii ( string_expression )

Example
ascii ( 'A' )

Result
Returns '65'

ceil

Returns the smallest integer greater than or equal to "numeric_expression".

Syntax
ceil ( numeric_expression )

char_length

Returns the number of logical characters in "string_expression". The number of logical characters can be distinct from the number of bytes in some East Asian locales.

Syntax
char_length ( string_expression )

chr

Returns the character that has the ASCII code value specified by "integer_expression". "Integer_expression" should be between 0 and 255.

Syntax
chr ( integer_expression )

concat

Returns a string that is the result of concatenating, or joining, "string_expression1" to "string_expression2".

Syntax
concat ( string_expression1 , string_expression2 )

Example
concat ( [Sales (query)].[Sales staff].[First name], [Sales (query)].[Sales staff].[Last name] )
Result

Returns the first name and last name; e.g., Bob Smith.

decode

Compares "expression" to each search value one by one. If "expression" is equal to a search, then it returns the corresponding result. If no match is found, it returns "default", or if "default" is omitted, it returns null.

Syntax

decode ( expression , search , result [ , search , result ]... [ , default ]

dump

Returns internal representation of "expression" with the format of "numeric_expression1" starting from position "numeric_expression2" for "numeric_expression3" characters.

Syntax

dump ( expression [ , numeric_expression1 [ , numeric_expression2 [ , numeric_expression3 ] ] ]

greatest

Returns the greatest value in "expression_list".

Syntax

greatest ( expression_list )

initcap

Returns "string_expression" with the first letter of each word in uppercase and all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

Syntax

initcap ( string_expression )

instr

Searches "string_expression1" starting at position "integer_expression1" for the "integer_expression2" occurrence of "string_expression2". If "integer_expression1" is negative, then the search occurs backwards from the end of "string_expression1". Returns an integer indicating the position of "string_expression2".

Syntax

instr ( string_expression1, string_expression2 [ , integer_expression1 [ , integer_expression2 ] ] )

instrb

Searches "string_expression1" starting at position "integer_expression1" for the "integer_expression2" occurrence of "string_expression2". If "integer_expression1" is negative, then the search occurs backwards from the end of "string_expression1". Returns the position (byte number) where "string_expression2" was found.

Syntax
least

Returns the least value in "expression_list".

Syntax
least ( expression_list )

length

Returns the number of characters in "string_expression".

Syntax
length ( string_expression )

lengthb

Returns the number of bytes in "string_expression".

Syntax
lengthb ( string_expression )

lpad

Returns "string_expression1" left-padded to the length defined by "integer_expression" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression", the appropriate portion of "string_expression1" is returned.

Syntax
lpad ( string_expression1, integer_expression [ , string_expression2 ] )

ltrim

Returns "string_expression1" with leading characters removed up to the first character not in "string_expression2".

Syntax
ltrim ( string_expression1 [ , string_expression2 ] )

Example
ltrim ( 'xyxXxyAB' , 'xy' )

Result
XxyAB

months_between

Returns the number of months from "date_expression1" to "date_expression2". If "date_expression1" is later than "date_expression2" then the result will be a positive number. The days and time portion of the difference are ignored, so the months are not rounded unless "date_expression1" and "date_expression2" are the last days of a month.
Syntax

months_between ( date_expression1 , date_expression2 )

**new_time**

Returns the datetime in "new_timezone" for "datetime_expression" in "old_timezone". "Old_timezone" and "new_timezone" can be one of 'AST', 'ADT', 'BST', 'BDT', 'CST', 'CDT', 'EST', 'EDT', 'HST', 'HDT', 'MST', 'MDT', 'NST', 'PST', 'PDT', 'YST', or 'YDT'.

Syntax

new_time ( datetime_expression , old_timezone , new_timezone )

**next_day**

Returns the datetime of the first weekday named by "string_expression" that is later than "datetime_expression". The return value has the same format as "datetime_expression".

Syntax

next_day ( datetime_expression , string_expression )

**nls_initcap**

Returns "string_expression1" with the first letter of each word in uppercase and all other letters in lowercase. A word begins after any character other than a letter. Thus, in addition to a blank space, symbols such as commas, periods, and colons can introduce a new word. "String_expression2" specifies the sorting sequence.

Syntax

nls_initcap ( string_expression1 [ , string_expression2 ] )

**nls_lower**

Returns "string_expression1" with all letters in lowercase. "String_expression2" specifies the sorting sequence.

Syntax

nls_lower ( string_expression1 [ , string_expression2 ] )

**nls_upper**

Returns "string_expression1" with all letters in uppercase. "String_expression2" specifies the sorting sequence.

Syntax

nls_upper ( string_expression1 [ , string_expression2 ] )

**nvl**

Returns the value of "expression1" if "expression1" is not NULL. If "expression1" is NULL, then returns the value of "expression2".

Syntax

nvl ( expression1 , expression2 )

Example

nvl ( [Unit sale price] , 0 )
Result

Returns the unit sale price, or returns 0 if the unit sale price is NULL.

replace

Replaces all occurrences of "string_expression2" in "string_expression1" with "string_expression3". If "string_expression3" is not specified, then it removes all occurrences of "string_expression2".

Syntax

replace ( string_expression1 , string_expression2 [ , string_expression3 ] )

round

Returns "numeric_expression" rounded to the nearest value "integer_expression" places right of the decimal point. If "integer_expression" is negative, "numeric_expression" is rounded to the nearest absolute value "integer_expression" places to the left of the decimal point. Rounding takes place before data formatting is applied.

Syntax

round ( numeric_expression [ , integer_expression ] )

Example

round ( 125 , -1 )

Result

Returns 130

rpad

Returns "string_expression1" right-padded to length "integer_expression" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression", the appropriate portion of "string_expression1" is returned. If "string_expression2" is not specified, then occurrences of "string_expression2" are replaced with spaces.

Syntax

rpad ( string_expression1 , integer_expression [ , string_expression2 ] )

rtrim

Returns "string_expression1" with the final characters removed after the last character not in "string_expression2". If "string_expression2" is not specified, the final space characters are removed.

Syntax

rtrim ( string_expression1 [ , string_expression2 ] )

Example

rtrim ( 'ABxxyx' , 'xy' )

Result
Returns 'ABxX'

**sign**

Returns an indicator of the sign of "numeric_expression", +1 if positive, 0 if zero, or -1 if negative.

**Syntax**

sign ( numeric_expression )

**soundex**

Returns a character string containing the phonetic representation of "string_expression".

**Syntax**

soundex ( string_expression )

**substr**

Returns the substring of "string_expression" that starts at position "integer_expression1" for "integer_expression2" characters or to the end of "string_expression" if "integer_expression2" is omitted. The first character in "string_expression" is at position 1.

**Syntax**

substr ( string_expression , integer_expression1 [ , integer_expression2 ] )

**Example**

substr ( [Sales (query)].[Sales staff].[Position code], 3, 5 )

**Result**

Returns characters 3 to 7 of the position codes.

**substrb**

Returns the substring of "string_expression" that starts at position "numeric_expression1" and ends after "numeric_expression2" bytes. The first byte in "string_expression" is at position 1. If you omit "numeric_expression2", returns the substring of "string_expression" that starts at position "numeric_expression1" and ends at the end of "string_expression".

**Syntax**

substrb ( string_expression , numeric_expression1 [ , numeric_expression2 ] )

**Example**

substrb ( [Sales (query)].[Sales staff].[Position code], 3, 5 )

**Result**

Returns characters 3 to 7 of the position codes.
**{sysdate}**

Returns a datetime value representing the current date and time of the computer that the database software runs on.

**Syntax**

```
{ sysdate }
```

**to_char**

Returns the string representation of "expression" with the format of "string_expression". "Expression" can be either a date value or a numeric value.

**Syntax**

```
to_char ( expression [ , string_expression ] )
```

**to_date**

Converts "string_expression1" to a datetime value as specified by the format "string_expression2". "String_expression3" specifies the format elements, such as language.

**Syntax**

```
to_date ( string_expression1 [ , string_expression2 [ ,
string_expression3 ] ] )
```

**to_number**

Converts "string_expression1" to a numeric value as specified by the format "string_expression2". "String_expression3" specifies the format elements, such as currency information.

**Syntax**

```
to_number ( string_expression1 , string_expression2 ,
string_expression3 )
```

**translate**

Returns "string_expression1" with all occurrences of each character in "string_expression2" replaced by the corresponding character in "string_expression3".

**Syntax**

```
translate ( string_expression1 , string_expression2 ,
string_expression3 )
```

**trunc**

Truncates "date_expression" using the format specified by "string_expression". For example, if "string_expression" is 'year', then "date_expression" is truncated to the first day of the year.

**Syntax**

```
trunc ( date_expression , string_expression )
```

**Example**

```
trunc ( 2003-08-22 , 'year' )
```
Result

Returns 2003-01-01.

**trunc**

Truncates digits from "numeric_expression1" using "numeric_expression2" as the precision.

**Syntax**

t trunc ( numeric_expression1 , numeric_expression2 )

**{user}**

Returns the username of the current Oracle user.

**Syntax**

{ user }

**vsize**

Returns the number of bytes in the internal representation of "expression". "Expression" must be a string expression.

**Syntax**

vsize ( expression )

---

**Paraccel String**

**overlay**

Returns the "string_expression1", replacing "string_expression2" from character position numeric_expression.

**Syntax**

overlay ( string_expression1 , string_expression2 , numeric_expression1 [ , numeric_expression2 ] )

**ltrim**

Returns "string_expression1", with leading characters removed up to the first character not in "string_expression2"; for example, ltrim ( 'xyxXxyAB' , 'xy' ) returns XxyAB.

**Syntax**

ltrim ( string_expression1 [ , string_expression2 ] )

**replace**

Returns "string_expression", having replaced "string_expression2" with "string_expression3".

**Syntax**

replace ( string_expression , string_expression2 , string_expression3 )
**rtrim**
Returns "string_expression1", with final characters removed after the last character not in "string_expression2"; for example, rtrim ('ABxXxyx', 'xy') returns ABx. If "string_expression2" is not specified, the final space characters are removed.

**Syntax**
rtrim ( string_expression1 [ , string_expression2 ] )

**Paracel Data type formatting**

**to_char**
Returns the string representation of "expression" with the format of "string_expression". "Expression" can be either a date value or a numeric value.

**Syntax**
to_char ( expression , string_expression )

to_date
Converts "string_expression1" to a date value as specified by the format "string_expression2".

**Syntax**
to_date ( string_expression1 , string_expression2 )

to_number
Converts "string_expression1" to a numeric value as specified by the format "string_expression2".

**Syntax**
to_number ( string_expression1 , string_expression2 )

**Paracel Math**

**cbrt**
Returns the cube root of "numeric_expression1".

**Syntax**
cbrt ( numeric_expression )

**pi**
Returns the constant of pi.

**Syntax**
pi ()

current_database
Returns the name of the current database.

**Syntax**
current_database ()
**current_schema**

Returns the name of the current schema

Syntax

current_schema ()

**{current_user}**

Syntax

{current_user}

**{session_user}**

Syntax

{session_user}

**translate**

Returns "string_expression1", with each occurrence of each character in "string_expression2" replaced by its corresponding character in "string_expression3".

Syntax

translate ( string_expression1 , string_expression2 , string_expression3 )

**version**

Returns the string value of the database version.

Syntax

version ()

---

**Postgres**

**Postgres String**

**overlay**

Returns the "string_expression1" replacing "string_expression2" from character position numeric_expression.

Syntax

overlay ( string_expression1 , string_expression2 , numeric_expression1 [ , numeric_expression2 ] )

**btrim**

Returns string_expression1 after removing the longest string of characters in "string_expression2".

Syntax

btrim ( string_expression1 [ , string_expression2 ] )

**initcap**

Returns "string_expression", with the first letter of each word in uppercase and all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.
**Syntax**

*initcap ( string_expression )*

**Ipad**

Returns "string_expression1" padded to length "integer_expression" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression", the appropriate portion of "string_expression1" is returned.

**Syntax**

*ipad ( string_expression1 , integer_expression [ , string_expression2 ] )*

**ltrim**

Returns "string_expression1", with leading characters removed up to the first character not in "string_expression2"; for example, ltrim ( 'xyxXyAB' , 'xy' ) returns XyAB.

**Syntax**

*ltrim ( string_expression1 [ , string_expression2 ] )*

**md5**

Returns the MD5 hash of "string_expression1".

**Syntax**

*md5 ( string_expression1 )*

**to_hex**

Returns the hexadecimal string representation of "numeric_expression1".

**Syntax**

*to_hex ( numeric_expression1 )*

**repeat**

Returns the "string_expression" repeated "numeric_expression1" times.

**Syntax**

*repeat ( string_expression , numeric_expression1 )*

**replace**

Returns "string_expression" with "string_expression2" replaced with "string_expression3".

**Syntax**

*replace ( string_expression , string_expression2 , string_expression3)*

**rpad**

Returns "string_expression1" right-padded to length "integer_expression" with occurrences of "string_expression2". If "string_expression1" is longer than "integer_expression", the appropriate portion of "string_expression1" is returned. If "string_expression2" is not specified, then spaces are used.

**Syntax**

*rpad ( string_expression1 , integer_expression [ , string_expression2 ] )*
**rtrim**
Returns "string_expression1", with final characters removed after the last character not in "string_expression2"; for example, rtrim ( 'ABxXyx' , 'xy' ) returns ABxX. If "string_expression2" is not specified, the final space characters are removed.

**Syntax**
rtrim ( string_expression1 [ , string_expression2 ] )

**split_part**
Returns "numeric_expression" field having split "string_expression1" on "string_expression2".

**Syntax**
split_part ( string_expression1 , string_expression2 , numeric_expression )

**Postgres Data type formatting**

**to_char**
Returns the string representation of "expression" with the format of "string_expression". "Expression" can be either a date value or a numeric value.

**Syntax**
to_char ( expression , string_expression )

**to_date**
Converts "string_expression1" to a date value as specified by the format "string_expression2".

**Syntax**
to_date ( string_expression1 , string_expression2 )

**to_number**
Converts "string_expression1" to a numeric value as specified by the format "string_expression2".

**Syntax**
to_number ( string_expression1 , string_expression2 )

**to_timestamp**
Converts "string_expression1" to a timestamp value as specified by the format "string_expression2". Alternate syntax: to_timestamp ( numeric-expression )
Converts an Unix epoch clock time to a timestamp value.

**Syntax**
to_timestamp ( string_expression1 , string_expression2 )

**Postgres Math**

**log**
Returns the base 10 logarithm of "numeric_expression1" or logarithm to the base "numeric_expression2".

**Syntax**
log ( numeric_expression1 [ , numeric_expression2 ] )
**In**  
Returns the natural logarithm of "numeric_expression1".

**Syntax**  
```sql  
ln ( numeric_expression )  
```  

**cbrt**  
Returns the cube root of "numeric_expression1".

**Syntax**  
```sql  
cbrt ( numeric_expression )  
```  

**div**  
Returns the integer quotient of "numeric_expression1" divided by "numeric_expression2".

**Syntax**  
```sql  
div ( numeric_expression1 , numeric_expression2 )  
```  

**pi**  
Returns the constant of pi.

**Syntax**  
```sql  
pi ()  
```  

**Postgres Trigonometry**

**acos**  
Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

**Syntax**  
```sql  
acos ( numeric_expression )  
```  

**asin**  
Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

**Syntax**  
```sql  
asin ( numeric_expression )  
```  

**atan**  
Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".

**Syntax**  
```sql  
atan ( numeric_expression )  
```  

**atan2**  
Returns the arctangent of the x and y coordinates specified by "numeric_expression1" and "numeric_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric_expression2" / "numeric_expression1".

**Syntax**  
```sql  
atan2 ( numeric_expression1 , numeric_expression2 )  
```
atan2 ( numeric_expression1 , numeric_expression2 )

cos
Returns the cosine of "numeric_expression", where "numeric_expression" is an angle expressed in radians.

Syntax
cos ( numeric_expression )

cot
Returns the cotangent of "numeric_expression", where "numeric_expression" is an angle expressed in radians.

Syntax
cot ( numeric_expression )

degrees
Returns the degrees where "numeric_expression" is an angle expressed in radians.

Syntax
degrees ( numeric_expression )

radians
Returns the radians where "numeric_expression" is an angle expressed in degrees.

Syntax
radians ( numeric_expression )

sin
Returns the sine of "numeric_expression", where "numeric_expression" is an angle expressed in radians.

Syntax
sin ( numeric_expression )

tan
Returns the tangent of "numeric_expression", where "numeric_expression" is an angle expressed in radians.

Syntax
tan ( numeric_expression )

ascii
Returns a number representing the ASCII code value of the leftmost character of "string_expression"; for example, ascii ( 'A' ) is 65.

Syntax
ascii ( string_expression )

chr
Returns the character that has the ASCII code value specified by "integer_expression". "Integer_expression" should be between 0 and 255.

Syntax
chr ( integer_expression )

{current_db}
Syntax
 current_database()

{current_catalog}
Syntax
 {current_catalog}

{current_schema}
Syntax
 {current_schema}

{current_user}
Syntax
 {current_user}

{session_user}
Syntax
 {session_user}

translate
Returns "string_expression1", with each occurrence of each character in
"string_expression2" replaced by its corresponding character in
"string_expression3".

Syntax
translate ( string_expression1 , string_expression2 ,
 string_expression3 )

date_trunc
Returns the timestamp to the specified precision.

Syntax
date_trunc ( string_expression , timestamp_expression )

version
Returns the string value of the database version.

Syntax
version ()

Red Brick

ceil
Returns the smallest integer greater than or equal to "numeric_expression" or
"string_expression". Note that "string_expression" must represent a valid numeric
value.
Syntax
ceil ( numeric_expression|string_expression )

**concat**
Returns a string that is the result of concatenating, or joining, "string_expression1" to "string_expression2".

Syntax
concat ( string_expression1 , string_expression2 )

Example
concat ( [Sales (query)].[Sales staff].[First name] , [Sales (query)].
[Sales staff].[Last name] )

Result
Returns the first name and last name; e.g., Bob Smith.

**{current_user}**
Returns the database username (authorization ID) of the current user.

Syntax
{ current_user }

date
Returns a date value. "Expression" can be either characters or a timestamp.

Syntax
date ( expression )

dateadd
Adds "interval" to "datetime_expression" and returns a result that is the same
datetime data type as "datetime_expression". "Datepart" refers to the year, month,
day, hour, minute, second. "Interval" must be an integer and "datetime_expression"
can be a date, time, or timestamp.

Syntax
dateadd ( { datepart } , interval , datetime_expression )

datediff
Determines the difference between two datetime expressions and returns an integer
result in "datepart" units. "Datepart" refers to a year, month, day, hour, minute, or
second. "Datetme_expression1" and "datetime_expression2" can be dates, times, or
timestamps.

Syntax
datediff ( { datepart } , datetime_expression1 , datetime_expression2 )

datename
Extracts "datepart" of "datetime_expression" and returns its value as a character
string. "Datepart" refers to a year, month, day, hour, minute, or second.
"Datetme_expression" can be a date, a time, or a timestamp.
Syntax
datename ( { datepart } , datetime_expression )

dec
Converts "expression" to a decimal value with the data type decimal (precision, scale). The default value of precision is 9. The default value of scale is 0.

Syntax
dec ( expression , [ precision , scale ] )

decimal
Converts "expression" to a decimal value with the data type decimal (precision, scale). The default value of precision is 9. The default value of scale is 0.

Syntax
decimal ( expression , [ precision , scale ] )

decode
Compares and converts "expression" to another value. If "expression" matches "target", it is replaced, otherwise it is replaced by "default" or null if no default is specified. The expressions can be any data type as long as they are all the same data type.

Syntax
decode ( expression , target , replacement [ ,default ] )

float
Converts "numeric_expression" into a double-precision floating-point value.

Syntax
float ( numeric_expression )

ifnull
Tests "expression" for missing values and replaces each one with "substitute". If "expression" is null, "substitute" is returned, otherwise it returns the value of "expression". The expressions can be any data type as long as they are all the same data type.

Syntax
ifnull ( expression , substitute )

int
Converts "numeric_expression" into an integer value and returns an integer value. If "numeric_expression" is null, it returns null.

Syntax
int ( numeric_expression )

integer
Converts "numeric_expression" into an integer value and returns an integer value. If "numeric_expression" is null, it returns null.
Syntax
integer ( numeric_expression )

Example
integer ( 84.95 )

Result
85

length
Returns an integer result specifying the number of characters in "string_expression". If "string_expression" is null, it returns null.

Syntax
length ( string_expression )

lengthb
Returns an integer result specifying the number of bytes in "string_expression". If "string_expression" is null, it returns null.

Syntax
lengthb ( string_expression )

ltrim
Removes leading blanks from "string_expression". If "string_expression" is null, it returns null.

Syntax
ltrim ( string_expression )

nullif
Returns null if both "expression1" and "expression2" have the same value. If they have different values, the value of "expression1" is returned. "Expression1" and "expression2" can be any data type as long as they are the same data type.

Syntax
nullif ( expression1 , expression2 )

positionb
Returns an integer that is relative to the beginning byte position of "string_expression1" in "string_expression2". If "string_expression1" is not located, the result is 0. If "string_expression1" is of zero length, the result is 1. If "string_expression1" is null, an error message is returned. If "string_expression2" is null, the result is 0.

Syntax
positionb ( string_expression1, string_expression2 )
**real**

Returns a real value. If "numeric_expression" is null, it returns null.

**Syntax**

```plaintext
real ( numeric_expression )
```

**round**

Returns "numeric_expression" rounded to the nearest value "integer_expression" places to the right of the decimal point. If "integer_expression" is negative, "numeric_expression" is rounded to the nearest absolute value "integer_expression" places to the left of the decimal point. Rounding takes place before data formatting is applied.

**Syntax**

```plaintext
round ( numeric_expression , integer_expression )
```

**Example**

```plaintext
round (125, -1)
```

**Result**

130

**rtrim**

Removes trailing blanks from "string_expression". If "string_expression" is null, it returns null.

**Syntax**

```plaintext
rtrim ( string_expression )
```

**Example**

```plaintext
rtrim ( [Sales (query)].[Sales staff].[Last name] )
```

**Result**

Returns last names with any spaces at the end of the name removed.

**sign**

Determines the sign of "numeric_expression", and returns 1 for a positive value, -1 for a negative value, and 0 for zero.

**Syntax**

```plaintext
sign ( numeric_expression )
```

**string**

Converts "expression" to a character string. "Expression" can be either numeric or datetime.

**Syntax**

```plaintext
string ( expression [ , length [ , scale ] ] )
```
**substr**

Returns a substring of "string_expression" that begins at position "start_integer" and continues for "length_integer" characters. If "length_integer" is not specified, a substring from "start_integer" to the end of "string_expression" is returned.

**Syntax**

```plaintext
substr ( string_expression , start_integer , length_integer )
```

**Example**

```plaintext
substr ( [Sales (query)].[Sales staff].[Position code], 3, 5)
```

**Result**

Returns characters 3 to 7 of the position codes.

**substrb**

Returns a substring of "string_expression" that begins at position "start_integer" and continues for "length_integer" bytes. If "length_integer" is not specified, a substring from "start_integer" to the end of "string_expression" is returned.

**Syntax**

```plaintext
substrb ( string_expression , start_integer , length_integer )
```

**time**

Creates a time value from "expression", which can be a character string or a time-stamp data type expression.

**Syntax**

```plaintext
time ( expression )
```

**timestamp**

Creates a time-stamp value from "timestamp_expression", which is a character string.

**Syntax**

```plaintext
timestamp ( timestamp_expression )
```

**timestamp**

Creates a time-stamp value from "time_expression" and "date_expression". If either "time_expression" or "date_expression" is null, the resulting time-stamp expression is also null.

**Syntax**

```plaintext
timestamp ( date_expression , time_expression )
```

**to_char**

Converts "source_date" to the character string specified by "format_string". "Source_date" can be a date, time, or timestamp data type.

**Syntax**

```plaintext
to_char ( source_date, format_string )
```
SAP BW Trigonometry

**arccos**
Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

Syntax
arccos ( numeric_expression )

**arcsin**
Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

Syntax
arcsin ( numeric_expression )

**arctan**
Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".

Syntax
arctan ( numeric_expression )

**cos**
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
cos ( numeric_expression )

**sin**
Returns the sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
sin ( numeric_expression )

**tan**
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
tan ( numeric_expression )

**coshyp**
Returns the hyperbolic cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
coshyp ( numeric_expression )
**sinhyp**
Returns the hyperbolic sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
sinhyp ( numeric_expression )

**tanhyp**
Returns the hyperbolic tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**
tanhyp ( numeric_expression )

---

**SAP BW Math**

**log10**
Returns the base ten logarithm of "numeric_expression".

**Syntax**
log10 ( numeric_expression )

---

**Salesforce.com**

**Date Functions**

**CALENDAR_MONTH**
Returns a number representing the calendar month of "date_expression".

**Syntax**
CALENDAR_MONTH ( date_expression )

**Example**
CALENDAR_MONTH ( '2012-02-29' )

**Result**
2

**CALENDAR_QUARTER**
Returns a number representing the calendar quarter of "date_expression".

**Syntax**
CALENDAR_QUARTER ( date_expression )

**Example**
CALENDAR_QUARTER ( '2012-02-29' )

**Result**
1
**CALENDAR_YEAR**
Returns a number representing the calendar year of "date_expression".

**Syntax**
CALENDAR_YEAR ( date_expression )

**Example**
CALENDAR_YEAR ( '2012-02-29' )

**Result**
2012

**DAY_IN_MONTH**
Returns a number representing the day in the month of "date_expression".

**Syntax**
DAY_IN_MONTH ( date_expression )

**Example**
DAY_IN_MONTH ( '2012-02-29' )

**Result**
29

**DAY_IN_WEEK**
Returns a number representing the day of the week for "date_expression" 1 for Sunday, 7 for Saturday.

**Syntax**
DAY_IN_WEEK ( date_expression )

**Example**
DAY_IN_WEEK ( '2012-02-29' )

**Result**
4 (Wednesday)

**DAY_IN_YEAR**
Returns a number representing the day in the year for "date_expression".

**Syntax**
DAY_IN_YEAR ( date_expression )

**Example**
DAY_IN_YEAR ( '2012-02-29' )

**Result**
60
**DAY_ONLY**
Returns a date representing the day portion of "dateTime_expression".

**Syntax**
DAY_ONLY ( dateTime_expression )

**Example**
DAY_ONLY ( '2012-02-29T23:00:01Z' )

**Result**
2012-02-29

**FISCAL_MONTH**
Returns a number representing the fiscal month of "date_expression". This differs from CALENDAR_MONTH() if your organization uses a fiscal year that does not match the Gregorian calendar. If your fiscal year starts in March, 1 for March 12 for February.

**Syntax**
FISCAL_MONTH ( date_expression )

**Example**
FISCAL_MONTH ( '2012-02-29' )

**Result**
12

**FISCAL_QUARTER**
Returns a number representing the fiscal quarter of "date_expression". This differs from CALENDAR_QUARTER() if your organization uses a fiscal year that does not match the Gregorian calendar. If your fiscal year starts in July, 1 for July 15 and 4 for June 6.

**Syntax**
FISCAL_QUARTER ( date_expression )

**Example**
FISCAL_QUARTER ( '2012-02-29' )

**Result**
3

**FISCAL_YEAR**
Returns a number representing the fiscal year of "date_expression". This differs from CALENDAR_YEAR() if your organization uses a fiscal year that does not match the Gregorian calendar.

**Syntax**
FISCAL_YEAR ( date_expression )

**Example**
FISCAL_YEAR ( '2012-02-29' )
**Result**

2012

**HOUR_IN_DAY**
Returns a number representing the hour in the day for "dateTime_expression".

**Syntax**
HOUR_IN_DAY ( dateTime_expression )

**Example**
HOUR_IN_DAY ( '2012-02-29T23:00:01Z' )

**Result**
23

**WEEK_IN_MONTH**
Returns a number representing the week in the month for "date_expression". The first week is from the first through the seventh day of the month

**Syntax**
WEEK_IN_MONTH ( date_expression )

**Example**
WEEK_IN_MONTH ( '2012-02-29' )

**Result**
5

**WEEK_IN_YEAR**
Returns a number representing the calendar week of a "date_expression". The first week is from January 1 through January 7

**Syntax**
WEEK_IN_YEAR ( date_expression )

**Example**
WEEK_IN_YEAR ( '2012-02-29' )

**Result**
9

**convertCurrency**
Converts "numeric_expression" to the user's currency when multicurrency is enabled.

**Syntax**
calculateCurrency ( numeric_expression )
**convertTimezone**
Converts "dateTime_expression" to the user's time zone.

**Syntax**
`convertTimezone ( dateTime_expression )`

---

**SQL Server Math**

**log**
Returns the natural logarithm of "numeric_expression".

**Syntax**
`log ( numeric_expression )`

**log10**
Returns the base ten logarithm of "numeric_expression".

**Syntax**
`log10 ( numeric_expression )`

**pi**
Returns the constant value of pi as a floating point value.

**Syntax**
`pi ()`

**rand**
Generates a random number using "integer_expression" as the seed value.

**Syntax**
`rand ( integer_expression )`

---

**SQL Server Trigonometry**

**acos**
Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

**Syntax**
`acos ( numeric_expression )`

**asin**
Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

**Syntax**
`asin ( numeric_expression )`

**atan**
Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".
Syntax
atan ( numeric_expression )

atan2
Returns the arctangent of the x and y coordinates specified by "numeric_expression1" and "numeric_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric_expression1".

Syntax
atan2 ( numeric_expression1, numeric_expression2 )

cos
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
cos ( numeric_expression )

cot
Returns the cotangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
cot ( numeric_expression )

degrees
Returns "numeric_expression" radians converted to degrees.

Syntax
degrees ( numeric_expression )

radians
Returns the number of radians converted from "numeric_expression" degrees.

Syntax
radians ( numeric_expression )

sin
Returns the sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
sin ( numeric_expression )

tan
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
tan ( numeric_expression )

ascii
Returns a number representing the ascii code value of the leftmost character of "string_expression".
Syntax
ascii ( string_expression )

Example
ascii ( 'A' )

Result
65

char
Returns the character that has the ASCII code value specified by
"integer_expression". "Integer_expression" should be between 0 and 255.

Syntax
char ( integer_expression )

Example
char ( 65 )

Result
A

charindex
Searches "string_expression2" for the first occurrence of "string_expression1" and
returns an integer. "Start_location" is the character position to start searching for
"string_expression1" in "string_expression2". If "start_location" is not specified, is a
negative number, or is zero, the search starts at the beginning of
"string_expression2".

Syntax
charindex ( string_expression1 , string_expression2 [ ,
          start_location ] )

{current_user}
Returns the name of the current user.

Syntax
{ current_user }

datalength
Returns the length in bytes of "string_expression".

Syntax
datalength ( string_expression )

dateadd
Returns the date resulting from the addition of "integer_expression" units
(indicated by "datepart" (day, month, year)) to "date_expression".

Syntax
```
dateadd ( { datepart } , integer_expression , date_expression )
```

**datediff**

Returns the number of "datepart" (day, month, year) units between "date_expression1" and "date_expression2".

**Syntax**

datediff ( {datepart} , date_expression1 , date_expression2 )

**Example**

datediff ( {yy} , 1984-01-01 , 1997-01-01 )

**Result**

13

**datename**

Returns "datepart" from "date_expression", which can be a datetime, smalldatetime, date, or time value as an ASCII string. Note that "datepart" must be a keyword representing a datepart or its abbreviation recognized by Microsoft® SQL Server and must be enclosed in curly brackets.

**Syntax**

datename ( ' { ' datepart ' } ' , date_expression )

**Example**

datename ( {mm} , 2000-01-01 )

**Result**

January

**datepart**

Returns part of "date_expression" (for example, the month) as an integer. 
"date_expression" can be a datetime, smalldatetime, date, or time value. Note that "datepart" must be a keyword representing a datepart or its abbreviation recognized by Microsoft® SQL Server and must be enclosed in curly brackets.

**Syntax**

datepart ( ' { ' datepart ' } ' , date_expression )

**Example**

datepart ( {wk} , 2000-01-01 )

**Result**

1 (first week of the year)

**day**

Returns the day portion of "date_expression". Same as extract (day from date_expression).

**Syntax**
difference

Returns an integer value representing the difference between the values returned by the data source-specific soundex function for "string_expression1" and "string_expression2". The value returned ranges from 0 to 4, with 4 indicating the best match. Note that 4 does not mean that the strings are equal.

Syntax

difference ( string_expression1 , string_expression2 )

Example 1

difference ([Sales target (query)].[Sales Staff].[First name],[Sales (query)].[Retailers].[Contact first name])

Result

0

Example 2

difference ([Sales target (query)].[Sales Staff].[First name],[Sales target (query)].[Sales Staff].[First name])

Result

4

getdate

Returns a datetime value representing the current date and time of the computer that the database software runs on.

Syntax

getdate ()

isnull

Returns the first non-null argument (or null if both arguments are null). It is recommended to use the sql standard COALESCE function instead.

Syntax

isnull ( expression , expression )

Example

isnull ( [Sales (query)].[Sales Fact].[Sales quantity] , 0 )

Result

Returns the sales quantity if it's not null, otherwise returns 0.

left

Returns the leftmost "integer_expression" characters of "string_expression".

Syntax

left ( string_expression , integer_expression )
Example
left ( [Sales (query)].[Sales staff].[Last name] , 3 )

Result
Returns the first three characters of each last name.

ltrim
Returns "string_expression" with leading spaces removed.

Syntax
ltrim ( string_expression )

month
Returns the month portion of "date_expression". Same as extract (month from date_expression).

Syntax
month ( date_expression )

patindex
Returns an integer that represents the starting position of the first occurrence of "string_expression1" in the "string_expression2". Returns 0 if "string_expression1" is not found. The % wildcard character must precede and follow "string_expression1", except when searching for first or last characters.

Syntax
patindex ( string_expression1 , string_expression2 )

Example
patindex ( '%po%', 'Report' )

Result
3

replace
Replaces all occurrences of "string_expression2" in "string_expression1" with "string_expression3".

Syntax
replace ( string_expression1 , string_expression2 , string_expression3 )

replicate
Returns a string consisting of "string_expression" repeated "integer_expression" times.

Syntax
replicate ( string_expression , integer_expression )
reverse
Returns "string_expression" in reverse order.

Syntax
reverse ( string_expression )

right
Returns the rightmost "integer_expression" characters of "string_expression".

Syntax
right ( string_expression , integer_expression )

cround
Returns "numeric_expression" rounded to the nearest value "integer_expression" places to the right of the decimal point. Rounding takes place before data formatting is applied.

Syntax
cround ( numeric_expression , integer_expression )

Example
cround (125, -1)

Result
130

crtrim
Returns "string_expression" with trailing spaces removed.

Syntax
crtrim ( string_expression )

Example
crtrim ([Sales (query)].[Sales staff].[Last name])

Result
Returns last names with any spaces at the end of the name removed.

csign
Returns an indicator of the sign "numeric_expression": +1 if "numeric_expression" is positive, 0 if zero or -1 if negative.

Syntax
csign ( numeric_expression )

csoundex
Returns a four character string representing the sound of the words in "string_expression".

Syntax
soundex ( string_expression )

space
Returns a string consisting of "integer_expression" spaces.

Syntax
space ( integer_expression )

str
Returns a string representation of "numeric_expression" where
"integer_expression1" is the length of the string returned and "integer_expression2"
is the number of decimal digits.

Syntax
str ( numeric_expression [ , integer_expression1 [ ,
integer_expression2 ] ] )

stuff
Returns a string where "integer_expression2" characters have been deleted from
"string_expression1" beginning at "integer_expression1", and where
"string_expression2" has been inserted into "string_expression1" at its start. The
first character in a string is at position 1.

Syntax
stuff ( string_expression1 , integer_expression1 ,
integer_expression2 , string_expression2 )

day
Returns the year portion of "date_expression". Same as extract (year from
date_expression).

Syntax
day ( date_expression )

Sybase

Sybase Math

log
Returns the natural logarithm of "numeric_expression".

Syntax
log ( numeric_expression )

log10
Returns the base ten logarithm of "numeric_expression".

Syntax
log10 ( numeric_expression )
\textbf{pi}
Returns the constant value of pi as a floating point value.

\textbf{Syntax}
pi ()

\textbf{sign}
Returns an indicator denoting the sign of "numeric_expression": +1 if "numeric_expression" is positive, 0 if "numeric_expression" is zero, or -1 if "numeric_expression" is negative.

\textbf{Syntax}
\texttt{sign ( numeric_expression )}

\section*{Sybase Trigonometry}

\textbf{acos}
Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

\textbf{Syntax}
\texttt{acos ( numeric_expression )}

\textbf{asin}
Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

\textbf{Syntax}
\texttt{asin ( numeric_expression )}

\textbf{atan}
Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".

\textbf{Syntax}
\texttt{atan ( numeric_expression )}

\textbf{tan}
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

\textbf{Syntax}
\texttt{tan ( numeric_expression )}

\textbf{atn2}
Returns the angle, in radians, whose tangent is "numeric_expression1" / "numeric_expression2".

\textbf{Syntax}
\texttt{atn2 ( numeric_expression1, numeric_expression2 )}

\textbf{cos}
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.
Syntax

\[
\text{cos ( numeric_expression )}
\]

**cot**
Returns the cotangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax

\[
cot ( \text{numeric_expression} )
\]

**degrees**
Returns "numeric_expression" radians converted to degrees.

Syntax

\[
degrees ( \text{numeric_expression} )
\]

**radians**
Returns the degree equivalent of "numeric_expression". Results are of the same type as "numeric_expression". For numeric or decimal expressions, the results have an internal precision of 77 and a scale equal to that of "numeric_expression". When the money datatype is used, an internal conversion to float may cause some loss of precision.

Syntax

\[
radians ( \text{numeric_expression} )
\]

**sin**
Returns the sine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax

\[
sin ( \text{numeric_expression} )
\]

**ascii**
Returns a number representing the ascii code value of the leftmost character of "string_expression".

Syntax

\[
\text{ascii ( string_expression )}
\]

Example

\[
\text{ascii ( 'A' )}
\]

Result

65

**char**
Converts "integer_expression" to a character value. Char is usually used as the inverse of ascii where "integer_expression" must be between 0 and 255. If the resulting value is the first byte of a multibyte character, the character may be undefined.
char ( integer_expression )

**charindex**

Returns an integer that represents the starting position of "string_expression1" within "string_expression2". If "string_expression1" is not found, zero is returned. If "string_expression1" contains wildcard characters, charindex treats them as literals.

**Syntax**

charindex ( string_expression1 , string_expression2 )

datalength

Returns the length in bytes of "string_expression".

**Syntax**

datalength ( string_expression )

dateadd

Returns the date resulting from adding "integer_expression" units indicated by datepart (day, month, year) to "date_expression". Note that "datepart" must be enclosed in curly brackets.

**Syntax**

dateadd ( ' { datepart } ' , integer_expression ,
    date_expression )

**Example**

dateadd ( {dd} , 16 , 1997-06-16 )

**Result**

Jul 2, 1997

datediff

Returns the number of units indicated by "datepart" (day, month, year) between "date_expression1" and "date_expression2". Note that "datepart" must be enclosed in curly brackets.

**Syntax**

datediff ( ' { datepart } ' , date_expression1 ,
    date_expression2 )

**Example**

datediff ( {yy} , 1984-01-01 , 1997-01-01 )

**Result**

13

datename

Returns "datepart" of "date_expression" as an ASCII string. "Date_expression" can be a datetime, smalldatetime, date, or time value. Note that "datepart" must be enclosed in curly brackets.

Appendix F. Using the expression editor  681
Syntax

datename ('{ datepart }', date_expression )

Example
datename ({mm}, 1999-05-01 )

Result

May

datepart

Returns "datepart" of "date_expression" as an integer. "Date_expression" can be a
datetime, smalldatetime, date, or time value. Note that the datepart argument must
be enclosed in curly brackets.

Syntax
datepart ('{ datepart }', date_expression )

Example
datepart ({mm}, 1999-05-01 )

Result

5

day

Returns the day of the month (1-31) from 'date_expression'.

Syntax
day ( date_expression )

difference

Returns an integer value representing the difference between the values returned
by the data source-specific soundex function for "string_expression1" and
"string_expression2". The value that is returned ranges from 0 to 4, with 4
indicating the best match. Note that 4 does not mean that the strings are equal.

Syntax
difference ( string_expression1 , string_expression2 )

Example 1
difference ([Sales target (query)].[Sales staff].[First name],[Sales (query)].[Retailers].[Contact first name])

Result

0

Example 2
difference ([Sales target (query)].[Sales staff].[First name],[Sales target (query)].[Sales staff].[First name])

Result
getdate

Returns current system date and time.

Syntax

gdate ()

left

Returns the leftmost "integer_expression" characters of "string_expression".

Syntax

left ( string_expression , integer_expression )

Example

left ( Sales (query).Sales staff.Last name , 3 )

Result

Returns the first three characters of each last name.

ltrim

Returns "string_expression" with any leading spaces removed.

Syntax

ltrim ( string_expression )

month

Returns the month number (1-12) from "date_expression".

Syntax

month ( date_expression )

patindex

Returns an integer representing the starting position of the first occurrence of "string_expression1" in "string_expression2" or returns 0 if "string_expression1" is not found. By default, patindex returns the offset in characters. The offset can be returned in bytes by setting the return type to bytes. The % wildcard character must precede and follow the pattern in "string_expression1", except when searching for first or last characters.

Syntax

patindex ( string_expression1 , string_expression2 [ using {bytes|chars|characters} ] )

rand

Returns a random float value between 0 and 1, using the optional "integer_expression" as a seed value.

Syntax

rand ( integer_expression )
replicate
Returns a string with the same datatype as "string_expression", containing the same expression repeated "integer_expression" times or as many times as will fit into a 225-byte space, whichever is less.

Syntax
replicate ( string_expression , integer_expression )

reverse
Returns the reverse of "string_expression".

Syntax
reverse ( string_expression )

right
Returns the rightmost "integer_expression" characters of "string_expression".

Syntax
right ( string_expression , integer_expression )

round
Returns "numeric_expression" rounded to the nearest value "integer_expression" places to the right of the decimal point. Rounding takes place before data formatting is applied.

Syntax
round ( numeric_expression, integer_expression )

Example
round (125, -1)

Result
130

rtrim
Returns "string_expression" with trailing spaces removed.

Syntax
rtrim ( string_expression )

Example
rtrim ( [Sales (query)].[Sales staff].[Last name] )

Result
Returns last names with any spaces at the end of the name removed.

soundex
Returns a four-character soundex code for character strings that are composed of a contiguous sequence of valid single- or double-byte Roman letter.
**Syntax**

`soundex ( string_expression )`

**space**

Returns a string with "integer_expression" single-byte spacing.

**Syntax**

`space ( integer_expression )`

**str**

Returns a string representation of "numeric_expression". "Integer_expression1" is the length of the returned string and has a default setting of 10. "Integer_expression2" is the number of decimal digits and has a default setting of 0. Both are optional values.

**Syntax**

`str ( numeric_expression [, integer_expression1 [, integer_expression2 ] ] )`

**stuff**

Deletes "integer_expression2" characters from "string_expression1" starting at "integer_expression1", and inserts "string_expression2" into "string_expression1" at that position. To delete characters without inserting other characters, "string_expression2" should be null and not " ", which indicates a single space.

**Syntax**

`stuff ( string_expression1 , integer_expression1 , integer_expression2 , string_expression2 )`

**substring**

Returns the substring of "string_expression" that starts at position "integer_expression1". "Integer_expression2" specifies the number of characters in the substring.

**Syntax**

`substring ( string_expression , integer_expression1 , integer_expression2 )`

**Example**

`substring ( [Sales (query)].[Sales staff].[Position code], 3, 5 )`

**Result**

Returns characters 3 to 7 of the position codes.

**to_unichar**

Returns a unichar expression with the value "integer_expression". If "integer_expression" is in the range 0xD800..0xDFFF, the operation is aborted. If the "integer_expression" is in the range 0..0xFFFF, a single Unicode value is returned. If "integer_expression" is in the range 0x10000..0x10FFFF, a surrogate pair is returned.

**Syntax**

`to_unichar ( integer_expression )`
uhighsurr

Returns 1 if the Unicode value at "integer_expression" is the high half of a surrogate pair (which should appear first in the pair). Otherwise, it returns 0. This function allows you to write explicit code for surrogate handling. Particularly, if a substring starts on a Unicode character where uhighsurr () is true, extract a substring of at least 2 Unicode values, as substr() does not extract just 1. Substr () does not extract half of a surrogate pair.

Syntax

uhighsurr ( string_expression , integer_expression )

ulowsurr

Returns 1 if the Unicode value at "integer_expression" is the low half of a surrogate pair (which should appear second in the pair). Otherwise, it returns 0. This function allows you to explicitly code around the adjustments performed by substr (), stuff (), and right (). Particularly, if a substring ends on a Unicode value where ulowsurr () is true, extract a substring of 1 less characters (or 1 more), since substr () does not extract a string that contains an unmatched surrogate pair.

Syntax

ulowsurr ( string_expression , integer_expression )

uscalar

Returns the Unicode scalar value for the first Unicode character in "string_expression". If the first character is not the high-order half of a surrogate pair, then the value is in the range 0..0xFFFF. If the first character is the high-order half of a surrogate pair, a second value must be a low-order half, and the return value is in the range 0x10000..0x10FFFF. If this function is called on a Unicode character expression containing an unmatched surrogate half, the operation is aborted.

Syntax

uscalar ( string_expression )

year

Returns the year from date_expression.

Syntax

year ( date_expression )

Teradata

Teradata Trigonometry

acos

Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression". "Numeric_expression" must be between -1 and 1, inclusive.

Syntax

acos ( numeric_expression )
**acosh**
Returns the inverse hyperbolic cosine of "numeric_expression" where "numeric_expression" can be any real number equal to or greater than 1.

Syntax
acosh ( numeric_expression )

**asin**
Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression". "Numeric_expression" must be between -1 and 1, inclusive.

Syntax
asin ( numeric_expression )

**asinh**
Returns the inverse hyperbolic sine of "numeric_expression" where "numeric_expression" can be any real number.

Syntax
asinh ( numeric_expression )

**atan**
Returns the arctangent of "numeric_expression" in radians where the arctangent is the angle whose tangent is "numeric_expression".

Syntax
atan ( numeric_expression )

**atan2**
Returns the arctangent of the x and y coordinates specified by "numeric_expression1" and "numeric_expression2", respectively, in radians. The returned angle will be between - and π radians, excluding π.

Syntax
atan2 ( numeric_expression1, numeric_expression2 )

**atanh**
Returns the inverse hyperbolic tangent of "numeric_expression" where "numeric_expression" can be any real number between 1 and -1, excluding 1 and -1.

Syntax
atanh ( numeric_expression )

**cos**
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
cos ( numeric_expression )

**cosh**
Returns the hyperbolic cosine of "numeric_expression" where "numeric_expression" can be any real number.
Syntax

\[ \cosh \left( \text{numeric\_expression} \right) \]

**sin**
Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

Syntax

\[ \sin \left( \text{numeric\_expression} \right) \]

**sinh**
Returns the hyperbolic sine of "numeric\_expression" where "numeric\_expression" can be any real number.

Syntax

\[ \sinh \left( \text{numeric\_expression} \right) \]

**tan**
Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

Syntax

\[ \tan \left( \text{numeric\_expression} \right) \]

**tanh**
Returns the hyperbolic tangent of "numeric\_expression" where "numeric\_expression" can be any real number.

Syntax

\[ \tanh \left( \text{numeric\_expression} \right) \]

**account**
Returns the account string for the current user.

Syntax

\{ \text{account} \}

**add\_months**
Returns the date or the datetime resulting from adding "integer\_expression" months to "date\_expression" or "datetime\_expression".

Syntax

\[ \text{add\_months} \left( \text{date\_expression|datetime\_expression , integer\_expression} \right) \]

**bytes**
Returns the number of bytes contained in "byte\_expression". "Byte\_expression" is restricted to BYTE or VARBYTE.

Syntax

\[ \text{bytes} \left( \text{byte\_expression} \right) \]
**case_n**
Evaluates "condition_expression_list" and returns the position of the first true condition, provided that no prior condition in the list evaluates to unknown. The keywords must be enclosed in curly brackets. No case is an optional condition that evaluates to true if every expression in the list evaluates to false. No case or unknown is an optional condition that evaluates to true if every expression in the list evaluates to false, or if an expression evaluates to unknown and all prior conditions in the list evaluate to false. Unknown is an optional condition that evaluates to true if an expression evaluates to unknown and all prior conditions in the list evaluate to false.

**Syntax**
```
case_n ( condition_expression_list [ , NO CASE|UNKNOWN|NO CASE OR UNKNOWN [ , UNKNOWN ] ] )
```

**char2hexint**
Returns the hexadecimal representation for "string_expression".

**Syntax**
```
char2hexint ( string_expression )
```

**characters**
Returns an integer value representing the number of logical characters or bytes contained in "string_expression".

**Syntax**
```
characters ( string_expression )
```

**database**
Returns the name of the default database for the current user.

**Syntax**
```
{database}
```

**date**
Returns the current date.

**Syntax**
```
{date}
```

**format**
Returns the declared format for "expression" as a variable character string of up to 30 characters.

**Syntax**
```
format ( expression )
```

**index**
Returns the starting position of "string_expression2" in "string_expression1".

**Syntax**
index ( string_expression1, string_expression2 )

log
Computes the base 10 logarithm of "numeric_expression". "Numeric_expression" must be a non-zero, positive, numeric expression.

Syntax
log ( numeric_expression )

nullif
Returns null if "scalar_expression1" and "scalar_expression2" are equal. Otherwise, it returns "scalar_expression1". "Scalar_expression1" and "scalar_expression2" can be any data type.

Syntax
nullif ( scalar_expression1, scalar_expression2 )

nullifzero
If "numeric_expression" is zero, converts it to null to avoid division by zero.

Syntax
nullifzero ( numeric_expression )

profile
Returns the current profile for the session or null if none.

Syntax
{profile}

random
Returns a random integer number for each row of the results table. "Lower_bound" and "upper_bound" are integer constants. The limits for "lower_bound" and "upper_bound" range from -2147483648 to 2147483647 inclusive. "Upper_bound" must be greater than or equal to "lower_bound".

Syntax
random ( lower_bound, upper_bound )

role
Returns the current role for the session or null if none.

Syntax
{role}

session
Returns the number of the session for the current user.

Syntax
{session}
**soundex**  
Returns a character string that represents the Soundex code for "string_expression".

**Syntax**  
soundex ( string_expression )

**substr**  
Returns the substring of "string_expression" that starts at position "integer_expression1" for "integer_expression2" characters. The first character in "string_expression" is at position 1. If "integer_expression2" is omitted, returns the substring of "string_expression" that starts at position "integer_expression1" and ends at the end of "string_expression".

**Syntax**  
substr ( string_expression , integer_expression1 [ , integer_expression2 ] )

**Example**  
substr ( [Sales (query)].[Sales staff].[Position code], 3, 5 )

**Result**  
Returns characters 3 to 7 of the position codes.

**time**  
Returns the current time based on a 24-hour day. According to Teradata documentation, the system function TIME is deprecated. Use CURRENT_TIME function instead. Please note that in Compatible Mode TIME returns time data type, while in Dynamic Query Mode time returns a value of type FLOAT.

**Syntax**  
{time}

**type**  
Returns the data type defined for "expression".

**Syntax**  
type ( expression )

**user**  
Returns the user name of the current user.

**Syntax**  
{user}

**vargraphic**  
Returns a character string that represents the vargraphic code for "string_expression".

**Syntax**  
vargraphic ( string_expression )
**zeroifnull**

Converts data from null to 0 to avoid errors created by a null value. If "numeric_expression" is not null, returns the value of "numeric_expression". If "numeric_expression" is a character string, it is converted to a numeric value of float data type. If "numeric_expression" is null or zero, it returns zero.

**Syntax**

zeroifnull ( numeric_expression )

---

**Vectorwise String**

**lpad**

Returns "string_exp1" padded to length "integer_exp" with occurrences of "string_exp2". If "string_exp1" is longer than "integer_exp", the appropriate portion of "string_exp1" is returned.

**Syntax**

lpad ( string_exp1, integer_exp [ , string_exp2 ] )

**ltrim**

Returns "string_exp1", with leading blank characters removed.

**Syntax**

ltrim ( string_exp1 )

**rtrim**

Returns "string_exp1", with trailing blank characters removed.

**Syntax**

rtrim ( string_exp1 )

**shift**

Returns "string_exp1" shifted by numeric_exp character. If "numeric_exp" is greater than zero shift to the right else shift to the left.

**Syntax**

shift ( string_exp1, numeric_exp )

**soundex**

Returns a four character code for the "string_exp1".

**Syntax**

soundex ( string_exp1 )

**squeeze**

Returns "string_exp1" with multiple whitespace characters collapsed into a single whitespace character.

**Syntax**

squeeze ( string_exp1 )
Vectorwise Math

**log**
Returns the base 10 logarithm of "numeric_exp1".

**Syntax**
\[
\text{log} \ ( \text{numeric\_exp1} )
\]

**ln**
Returns the natural logarithm of "numeric_exp1".

**Syntax**
\[
\text{ln} \ ( \text{numeric\_exp} )
\]

**pi**
Returns the constant of pi.

**Syntax**
\[
\text{pi}( )
\]

Vectorwise Trigonometry

**acos**
Returns the arccosine of "numeric_exp" in radians. The arccosine is the angle whose cosine is "numeric_exp".

**Syntax**
\[
\text{acos} \ ( \text{numeric\_exp} )
\]

**asin**
Returns the arcsine of "numeric_exp" in radians. The arcsine is the angle whose sine is "numeric_exp".

**Syntax**
\[
\text{asin} \ ( \text{numeric\_exp} )
\]

**atan**
Returns the arctangent of "numeric_exp" in radians. The arctangent is the angle whose tangent is "numeric_exp".

**Syntax**
\[
\text{atan} \ ( \text{numeric\_exp} )
\]

**atan2**
Returns the arctangent of the x and y coordinates specified by "numeric_exp1" and "numeric_exp2", respectively, in radians. The arctangent is the angle whose tangent is "numeric_exp2" / "numeric_exp1".

**Syntax**
\[
\text{atan2} \ ( \text{numeric\_exp1} , \text{numeric\_exp2} )
\]

**cos**
Returns the cosine of "numeric_exp" where "numeric_exp" is an angle expressed in radians.
Syntax
\[ \cos \left( \text{numeric\_exp} \right) \]

**sin**
Returns the sine of "numeric\_exp" where "numeric\_exp" is an angle expressed in radians.

Syntax
\[ \sin \left( \text{numeric\_exp} \right) \]

**tan**
Returns the tangent of "numeric\_exp" where "numeric\_exp" is an angle expressed in radians.

Syntax
\[ \tan \left( \text{numeric\_exp} \right) \]

**\{initial\_user\}**
Returns the initial username.

Syntax
\[ \{ \text{initial\_user} \} \]

**\{session\_user\}**
Returns the session username.

Syntax
\[ \{ \text{session\_user} \} \]

**numeric\_trunc**
Returns the numeric\_exp1 truncated at the decimal place or numeric\_exp2 digits to the left or right of the decimal place.

Syntax
\[ \text{trunc} \left( \text{numeric\_exp1}, \text{numeric\_exp2} \right) \]

---

**Vertica**

**Vertica String**

**overlay**
Returns the "string\_expression1", replacing "string\_expression2" from character position numeric\_expression.

Syntax
\[ \text{overlay} \left( \text{string\_expression1}, \text{string\_expression2}, \text{numeric\_expression1} [\,, \text{numeric\_expression2}] \right) \]

**btrim**
Returns string\_expression1 after removing the longest string of characters in string\_expression2.

Syntax
initcap
Returns 'string_expression', with the first letter of each word in uppercase and all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

Syntax
initcap ( string_expression )

lpad
Returns 'string_expression1' padded to length 'integer_expression' with occurrences of 'string_expression2'. If 'string_expression1' is longer than 'integer_expression', the appropriate portion of 'string_expression1' is returned.

Syntax
lpad ( string_expression1, integer_expression [ , string_expression2 ] )

ltrim
Returns 'string_expression1', with leading characters removed up to the first character not in 'string_expression2'; for example, ltrim ('xyxyAB', 'xy') returns XxyAB.

Syntax
ltrim ( string_expression1 [ , string_expression2 ])

to_hex
Returns the hexadecimal string representation of 'numeric_exp1'.

Syntax
to_hex ( numeric_expression1 )

repeat
Returns the 'string_expression' repeated 'numeric_expression1' times.

Syntax
repeat ( string_expression , numeric_expression1 )

replace
Returns 'string_expression' having replaced 'string_expression2' with 'string_expression3'.

Syntax
replace ( string_expression , string_expression2 , string_expression3 )

rpad
Returns 'string_expression1' right-padded to length 'integer_expression' with occurrences of 'string_expression2'. If 'string_expression1' is longer than 'integer_expression', the appropriate portion of 'string_expression1' is returned. If 'string_expression2' is not specified, then spaces are used.

Syntax
rpad ( string_expression1, integer_expression [ , string_expression2 ] )
**rtrim**
Returns "string_expression1", with final characters removed after the last character not in "string_expression2"; for example, rtrim ('ABxXyx', 'xy') returns ABxX. If "string_expression2" is not specified, the final space characters are removed.

**Syntax**
rtrim ( string_expression1 [ , string_expression2 ] )

Vertica Data type formatting

**to_char**
Returns the string representation of "expression" with the format of "string_expression". "Expression" can be either a date value or a numeric value.

**Syntax**
to_char ( expression , string_expression )

to_date
Converts "string_expression1" to a date value as specified by the format "string_expression2".

**Syntax**
to_date ( string_expression1 , string_expression2 )

to_number
Converts "string_expression1" to a numeric value as specified by the format "string_expression2".

**Syntax**
to_number ( string_expression1, string_expression2 )

to_timestamp
Converts "string_expression1" to a timestamp value as specified by the format "string_expression2".

**Syntax**
to_timestamp ( string_expression1, string_expression2 )

Vertica Math

**log**
Returns the base 10 logarithm of "numeric_expression1" or logarithm to the base "numeric_expression2".

**Syntax**
log ( numeric_expression1 [ , numeric_expression2 ] )

**ln**
Returns the natural logarithm of "numeric_expression1".

**Syntax**
ln ( numeric_expression )
**Vertica Trigonometry**

**acos**
Returns the arccosine of "numeric_expression" in radians. The arccosine is the angle whose cosine is "numeric_expression".

**Syntax**
acos ( numeric_expression )

**asin**
Returns the arcsine of "numeric_expression" in radians. The arcsine is the angle whose sine is "numeric_expression".

**Syntax**
asin ( numeric_expression )

**atan**
Returns the arctangent of "numeric_expression" in radians. The arctangent is the angle whose tangent is "numeric_expression".

**Syntax**
atan ( numeric_expression )

**atan2**
Returns the arctangent of the x and y coordinates specified by "numeric_expression1" and "numeric_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric_expression2" / "numeric_expression1".

**Syntax**
atan2 ( numeric_expression1 , numeric_expression2 )

**cos**
Returns the cosine of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

**Syntax**

cos ( numeric_expression )
Syntax
cot ( numeric_expression )

**degrees**
Returns the degrees where "numeric_expression" is an angle expressed in radians.

Syntax
degrees ( numeric_expression )

**radians**
Returns the radians where "numeric_expression" is an angle expressed in degrees.

Syntax
radians ( numeric_expression )

**sin**
Returns the sine of "numeric_exp" where "numeric_expression" is an angle expressed in radians.

Syntax
sin ( numeric_expression )

**tan**
Returns the tangent of "numeric_expression" where "numeric_expression" is an angle expressed in radians.

Syntax
tan ( numeric_expression )

**ascii**
Returns a number representing the ASCII code value of the leftmost character of "string_expression"; for example, ascii ( 'A' ) is 65.

Syntax
ascii ( string_expression )

**chr**
Returns the character that has the ASCII code value specified by "integer_expression". "Integer_expression" should be between 0 and 255.

Syntax
chr ( integer_expression )

**current_database**
Returns the name of the current database.

Syntax
current_database ()

**current_schema**
Returns the name of the current schema.

Syntax
current_schema ()

{current_user}

Syntax
{current_user}

{session_user}

Syntax
{session_user}

translate

Returns "string_expression1", with each occurrence of each character in
"string_expression2" replaced by its corresponding character in
"string_expression3".

Syntax
translate ( string_expression1 , string_expression2 ,
string_expression3 )

date_trunc

Returns the timestamp to the specified precision.

Syntax
date_trunc ( string_expression , timestamp_expression)

version

Returns the string value of the database version.

Syntax
version ()

---

Macro Functions

This list contains functions that can be used within a macro. A macro may contain
one or more macro functions. A macro is delimited by a number sign (#) at the
beginning and at the end. Everything between the number signs is treated as a
macro expression and is executed at run time. For macro functions that accept
expressions of datatype timestamp with time zone as arguments, the accepted
format is 'yyyy-mm-dd hh:mm:ss[.ff]+hh:mm' where fractional seconds are optional
and can be represented by 1 to 9 digits. In lieu of a space separating the date
portion to the time portion, the character 'T' is also accepted. Also, in lieu of the
time zone '+hh:mm', the character 'Z' is accepted and will be processed internally
as '+00:00'. The macro functions that return expressions of datatype timestamp
with time zone return 9 digits by default for their fractional seconds. The macro
function timestampMask () can be used to trim the output if required.

+

Concatenates two strings.

Syntax
value1 + value2
Example

# '{ ' + $runLocale + '}' #

Result

[en-us]

_add_days

Returns the timestamp with time zone (as a string) that results from adding "integer_expression" number of days to "string_expression", where "string_expression" represents a timestamp with time zone.

Syntax

_add_days ( string_expression , integer_expression )

Example 1

# _add_days ( '2005-11-01 12:00:00.000-05:00' , -1 ) #

Result

2005-10-31 12:00:00.000000000-05:00

Example 2

# _add_days ( $current_timestamp,1)#

Result

2005-11-02 12:00:00.000000000-05:00

Example 3

# timestampMask ( _add_days ( $current_timestamp,1),

'yyyy-mm-dd' ) #

Result

2005-11-02

_add_months

Returns the timestamp with time zone (as a string) that results from adding "integer_expression" number of months to "string_expression", where "string_expression" represents a timestamp with time zone.

Syntax

_add_months ( string_expression , integer_expression )

Example 1

# _add_months ( '2005-11-01 12:00:00.000-05:00' , -1 ) #

Result

2005-10-01 12:00:00.000000000-05:00

Example 2

# _add_months ( $current_timestamp , 1 ) #
Example 3

```sql
# timestampMask (_add_months ($current_timestamp , 1) , 'yyyy-mm-dd') #
```

Result

2005-12-01

---

### _add_years

Returns the timestamp with time zone (as a string) that results from adding "integer_expression" number of years to "string_expression", where "string_expression" represents a timestamp with time zone.

**Syntax**

```
_add_years ( string_expression , integer_expression )
```

**Example 1**

```sql
# _add_years ( '2005-11-01 12:00:00.000-05:00' , -1 ) #
```

**Result**

2004-11-01 12:00:00.000000000-05:00

**Example 2**

```sql
# _add_years ( $current_timestamp , 1 ) #
```

**Result**

2006-11-01 12:00:00.000000000-05:00

**Example 3**

```sql
# timestampMask ( _add_years ( $current_timestamp , 1) , 'yyyy-mm-dd' ) #
```

**Result**

2006-11-01

---

### array

Constructs an array out of the list of parameters.

**Syntax**

```
array ( string_expression|array_expression { ,
                                string_expression|array_expression } )
```

**Example**

```sql
# csv ( array ( 'a1' , array ( 'x1' , 'x2' ) , 'a2' ) ) #
```

**Result**

2005-12-01
'a1', 'x1', 'x2', 'a2'

**csv**

Constructs a string from the elements of the array where the values are separated by commas. Optionally, the separator and quote strings can be specified. The default separator is a comma (,) and the default quote character is a single quote (').

**Syntax**

csv (array_expression [ , separator_string [ , quote_string ] ])

**Example**

# csv (array ( 'a1', 'a2')) #

**Result**

'a1', 'a2'

dq

Surrounds "string_expression" with double quotes.

**Syntax**

dq (string_expression)

**Example**

# dq ( 'zero' ) #

**Result**

"zero"

**getConfigurationEntry**

Get an entry from the IBM® Cognos® configuration file. The force_decode_flag is optional and must be one of: 'true', '1', 'false', '0', 0. The default is 'true'. When true, the value of the configuration entry will be decrypted into plain text if it is encrypted.

**Syntax**

getConfigurationEntry (entry_string , force_decode_flag)

**Example**

# getConfigurationEntry ( 'serverLocale' ) #

**Result**

en

grep

Searches for and returns elements of an array that match the pattern specified in "pattern_string".

**Syntax**

grep ( pattern_string , array_expression )
Example

# csv ( grep ( 's' , array ( 'as', 'an', 'arts' ) ) )

Result

'as', 'arts'

_function_of_month_

Returns a timestamp with time zone (as a string) by converting the day value in "string_expression" to 1, where "string_expression" is a timestamp with time zone.

Syntax

_function_of_month ( string_expression )

Example 1

# _first_of_month ( '2005-11-11 12:00:00.000-05:00' )#

Result

2005-11-01 12:00:00.000000000-05:00

Example 2

# timestampMask ( _first_of_month ( '2005-11-11 12:00:00.000-05:00' ) , 'yyyyMMdd' )#

Result

20051101

_function_of_month_

Returns a timestamp with time zone (as a string) that is the last day of the month represented by "string_expression", where "string_expression" is a timestamp with time zone.

Syntax

_function_of_month ( string_expression )

Example 1

# _last_of_month ( '2005-11-11 12:00:00.000-05:00' )#

Result

2005-11-30 12:00:00.000000000-05:00

Example 2

# timestampMask ( _last_of_month ( '2005-11-11 12:00:00.000-05:00' ) , 'yyyy-mm-dd' )#

Result

2005-11-30
**join**

Joins the elements of an array using "separator_string".

**Syntax**

join ( separator_string , array_expression )

**Example**

```sql
# sq( join ( ' | ' , array ( 'as', 'an', 'arts' )))#
```

**Result**

'as || an || arts'

**lstrip**

Strips the leading characters from the first argument. The optional second argument defines the set of characters to strip. By default, this function strips white space (ie. space, tab, carriage return and line feed).

**Syntax**

lstrip ( string_expression [ , set_of_characters ] )

**Example 1**

```sql
# sq( lstrip ( ' abc '))#
```

**Result**

'abc '

**Example 2**

' ' #

**Result**

53.2100

**prompt**

Prompts the user for a single value. Only "prompt_name" is required. The datatype defaults to "string" when it is not specified. The prompt is optional when "defaultText" is specified. "Text", when specified, will precede the value. "QueryItem" can be specified to take advantage of the prompt information properties of "queryItem". "Trailing_text", when specified, will be appended to the value.

**Syntax**

prompt ( prompt_name , datatype , defaultText , text , queryItem , trailing_text )

**Example 1**

```sql
select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE > #prompt ( 'Starting CountryCode' , 'integer' , '10' )#
```

**Result**

select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE > 10
Example 2

```sql
[gosales].[COUNTRY].[COUNTRY] = # prompt('countryPrompt','string','''Canada''' )#
```

Result

```sql
[gosales].[COUNTRY].[COUNTRY] = 'Canada'
```

Notes

- The "defaultText" parameter must be specified such that it is literally valid in the context of the macro since no formatting takes place on this value. The default string "Canada" in Example 2 is specified as a string using single quotes, in which the embedded single quotes are doubled up, thus 3 quotes. This results in the string being properly displayed within single quotes in the expression. As a general rule for the string datatype, "defaultText" should always be specified like this, except in the context of a stored procedure parameter. For "defaultText" of types 'date' or 'datetime', a special format should be used in the context of SQL. Examples of these formats are 'DATE ''2001-12-25''' and 'DATETIME ''2001-12-25 12:00:00'''. In all other contexts, use the date/datetime without the keyword and escaped single quotes (e.g., '2001-12-25').

promptmany

Prompts the user for one or more values. Only "prompt_name" is required. The datatype defaults to string when it is not specified. The prompt is optional when "defaultText" is specified. "Text", when specified, will precede the list of values. "QueryItem" can be specified to take advantage of the prompt information properties of "queryItem". "Trailing_text", when specified, will be appended to the list of values.

Syntax

```sql
promptmany ( prompt_name , datatype , defaultText , text , queryItem , trailing_text )
```

Example 1

```sql
select... where COUNTRY_MULTILINGUAL.COUNTRY in ( # promptmany('CountryName')#)
```

Result

```sql
select... where COUNTRY_MULTILINGUAL.COUNTRY_CODE in ( 'Canada' , 'The Netherlands' , 'Russia' )
```

Example 2

```sql
select... from gosales.gosales.dbo.COUNTRY_MULTILINGUAL
COUNTRY_MULTILINGUAL , gosales.gosales.dbo.COUNTRY XX where
COUNTRY_MULTILINGUAL.COUNTRY_CODE = XX.COUNTRY_CODE # promptmany ( 'Selected CountryCodes' , 'integer' , '' , '' , '' and
COUNTRY_MULTILINGUAL.COUNTRY_CODE in ( ' ' , ' ' , ' ' ) )#
```

Result

```sql
select... from gosales.gosales.dbo.COUNTRY_MULTILINGUAL
COUNTRY_MULTILINGUAL , gosales.gosales.dbo.COUNTRY XX where
COUNTRY_MULTILINGUAL.COUNTRY_CODE = XX.COUNTRY_CODE and
COUNTRY_MULTILINGUAL.COUNTRY_CODE in ( 'Canada' , 'The Netherlands' , 'Russia' )
```
**rstrip**

Strips the trailing characters from the first argument. The optional second argument defines the set of characters to strip. By default, this function strips white space (i.e. space, tab, carriage return and line feed).

**Syntax**

```
rstrip ( string_expression [ , set_of_characters ] )
```

**Example 1**

```
# sq( rstrip ( ' abc '))#
```

**Result**

' abc'

**Example 2**

```
' ) #
```

**Result**

0053.21

**sb**

Surrounds "string_expression" with square brackets.

**Syntax**

```
sb ( string_expression )
```

**Example**

```
# sb ( 'abc' ) #
```

**Result**

[abc]

**sq**

Surrounds "string_expression" with single quotes.

**Syntax**

```
sq ( string_expression )
```

**Example**

```
# sq ( 'zero' ) #
```

**Result**

'zero'

**sort**

Sorts the elements of the array in alphabetical order. Duplicates are retained.

**Syntax**

```
sort ( array_expression )
```
Example

# csv ( sort ( array ( 's3', 'a', 'x') ) ) #

Result

'a', 's3', 'x'

split

Splits a string or string elements of the array into separate elements.

Syntax

split ( pattern_string, string_expression | array_expression )

Example 1

# csv ( split ( '::', 'ab=c::de=f::gh=i') ) #

Result

'ab=c', 'de=f', 'gh=i'

Example 2

# csv ( split ( '=', split ( '::', 'ab=c::de=f::gh=i') ) ) #

Result

'ab', 'c', 'de', 'f', 'gh', 'i'

strip

Strips the leading and trailing characters from the first argument. The optional second argument defines the set of characters to strip. By default, this function strips white space (i.e. space, tab, carriage return and line feed).

Syntax

strip ( string_expression [ , set_of_characters ] )

Example 1

# sq ( strip ( ' abc ' ) ) #

Result

'abc'

Example 2

' ' ) #

Result

53.21

substitute

Searches for a pattern in a string or in the string elements of an array and substitutes the first occurrence of "pattern_string" with "replacement_string".
Syntax
substitute ( pattern_string, replacement_string,
string_expression|array_expression )

Example 1
# sq ( substitute ( ^cn=", ***", 'cn=help' ) )#

Result
'***help'

Example 2
# csv ( substitute ( ^cn=", ***", array ( 'cn=help' , 'acn=5' )))

Result
'***help', 'acn=5'

Example 3
# csv ( substitute ( 'cn=', '', array ( 'cn=help' , 'acn=5')))#

Result
'help', 'a5'

timestampMask

Returns "string_expression1", representing a timestamp with time zone, trimmed to
the format specified in "string_expression2". The format in "string_expression2"
must be one of the following: 'yyyy', 'mm', 'dd', 'yyyy-mm', 'yyyyymm',
'yyyy-mm-dd', 'yyyyymmd', 'yyyy-yy-mm-dd hh:mm:ss', 'yyyy-mm-dd
hh:mm:ss+hh:mm', 'yyyy-mm-dd hh:mm:ss.ff3', 'yyyy-yy-mm-dd
hh:mm:ss.ff3+hh:mm', 'yyyy-mm-ddThh:mm:ss', 'yyyy-mm-ddThh:mm:ss+hh:mm',
'yyyy-mm-ddThh:mm:ss.ff3+hh:mm', or 'yyyy-mm-ddThh:mm:ss.ff3+hh:mm'. The
macro functions that return a string representation of a timestamp with time zone
show a precision of 9 digits for the fractional part of the seconds by default. The
format options allow this to be trimmed down to a precision of 3 or 0.

Syntax
timestampMask ( string_expression1 , string_expression2 )

Example 1
# timestampMask ( $current_timestamp , 'yyyy-dd-mm' )#

Result
2005-11-01

Example 2
# timestampMask ( '2005-11-01 12:00:00.000-05:00' , 'yyyy-mm-dd
hh:mm:ss+hh:mm' )#

Result
2005-11-01 12:00:00-05:00
Example 3
# timestampMask ( '2005-11-01 12:00:00.123456789-05:00' ,
  'yyyy-mm-ddThh:mm:ss+hh:mm.ff3+hh:mm' ) #

Result
2005-11-01T12:00:00.123-05:00

toLocal
Returns the string representing a timestamp with time zone resulting from
adjusting "string_expression" to the time zone of the operating system. Note that
the macro function timestampMask () can be used to trim the output.

Syntax
toLocal ( string_expression )

Example 1
# toLocal ( '2005-11-01 17:00:00.000-00:00' ) # where OS local time
zone is -05:00

Result
2005-11-01 12:00:00.000000000-05:00

Example 2
# timestampMask ( toLocal ( '2005-11-01 17:00:00.000-00:00' ) ,
  'yyyy-mm-dd hh:mm:ss+hh:mm' ) # where OS local time zone is -05:00

Result
2005-11-01 12:00:00-05:00

Example 3
# toLocal ( '2005-11-01 13:30:00.000-03:30' ) # where OS local time
zone is -05:00

Result
2005-11-01 12:00:00.000000000-05:00

tolower
Returns the string "string_expression" with all the characters converted to lower
case using the rules of the locale "locale_string". If no locale is specified, the locale
'en' is used.

Syntax
tolower ( string_expression [ , locale_string ] )

Example 1
# tolower ( 'ABC' ) #

Result
abc
Example 2

# tolower ( 'ABC' , 'fr' ) #

Result

abc

toupper

Returns the string "string_expression" with all the characters converted to upper case using the rules of the locale defined in "locale_string". If "locale_string" is not specified, the locale 'en' is used.

Syntax
toupper ( string_expression [ , locale_string ] )

Example 1

# toupper ( 'abc' ) #

Result

ABC

Example 2

# toupper ( 'abc' , 'fr' ) #

Result

ABC

toUTC

Returns the string representing a timestamp with time zone resulting from adjusting "string_expression" to the zero-point reference UTC time zone, also known as GMT time. Note that the macro function timestampMask () can be used to trim the output.

Syntax
toUTC ( string_expression )

Example 1

# toUTC ( '2005-11-01 12:00:00.000-05:00' ) #

Result

2005-11-01 17:00:00.000000000-00:00

Example 2

# timestampMask( toUTC ( '2005-11-01 12:00:00.000-05:00' ) , 'yyyy-mm-dd hh:mm:ss.ff3+hh:mm' ) #

Result

2005-11-01 17:00:00.000-00:00

Example 3
# toUTC ( $current_timestamp ) #

Result

2005-11-01 17:00:00.000000000-00:00

**unique**

Removes duplicate entries from the array. The order of the elements is retained.

**Syntax**

unique ( array_expression )

**Example**

# csv ( unique ( array ( 's3', 'a', 's3', 'x') ) ) #

Result

's3', 'a', 'x'

**urlencode**

URL-encodes the passed argument. This function is useful when specifying XML connection strings.

**Syntax**

urlencode ( prompt ( 'userValue' ) )

**Example**

urlencode ( prompt ( 'some_val' ) )

Result

%27testValue%27

**CSVIdentityName**

Uses the identity information of the current authenticated user to look up values in the specified parameter map. Each individual piece of the user's identity (account name, group names, role names) is used as a key into the map. The unique list of values that is retrieved from the parameter map is then returned as a string, where each value is surrounded by single quotes and where multiple values are separated by commas.

**Syntax**

CSVIdentityName ( %parameter_map_name [ , separator_string ] )

**Example**

# CSVIdentityName ( %security_clearance_level_map ) #

Result

'level_500', 'level_501', 'level_700'
**CSVIdentityNameList**

Returns the pieces of the user's identity (account name, group names, role names) as a list of strings. The unique list of values is returned as a string, where each value is surrounded by single quotes and where multiple values are separated by commas.

**Syntax**

```plaintext
CSVIdentityNameList ( [ separator_string ] )
```

**Example**

```plaintext
# CSVIdentityNameList ( ) #
```

**Result**

'Everyone', 'Report Administrators', 'Query User'

**CAMPassport**

Returns the Cognos® Access Manager passport.

**Syntax**

```plaintext
CAMPassport ( )
```

**Example**

```plaintext
# CAMPassport ( ) #
```

**Result**

111:98812d62-4fd4-037b-4354-26414cf7ebef:3677162321

**CAMIDList**

Returns the pieces of the user's Cognos® Access Manager ID (CAMID), such as account name, group names, or role names, as a list of values separated by commas.

**Syntax**

```plaintext
CAMIDList ( [ separator_string ] )
```

**Example**

```plaintext
# CAMIDList ( ) #
```

**Result**

CAMID (::Everyone) , CAMID (::Authors) , CAMID (::Query Users) , CAMID (::Consumers) , CAMID (::Metrics Authors)

**CAMIDListForType**

Returns an array of the user's Cognos® Access Manager IDs (CAMIDs) based on the identity type (account, group, or role). CAMIDListForType can be used with the macro functions csv or join.

**Syntax**

```plaintext
CAMIDListForType ( identity type )
```
Example

\[
[qs].[userRole] \text{ in } ( \text{ # csv ( CAMIDListForType ( 'role' ) ) } )
\]

Result

\[
[qs].[userRole] \text{ in } ( \text{ CAMID ( "::System Administrators" ) } , \text{ CAMID ( "::Authors" ) })
\]

**simple case**

This macro construct is the template for a simple case, including the case, when, else, and end functions. Note that this macro construct is only supported in DQM mode.

**Syntax**

\[
\text{CASE } \text{ expression} \text{ WHEN } \text{ literal} \text{ THEN } \text{ expression} \text{ [ELSE } \text{ expression}] \text{ END}
\]

**Example**

#\text{CASE } \text{ prompt('pDateRange','token') WHEN 'Current Week' THEN '[PR Current Week]' ELSE '[PR Prior Week]' END}#

**Result**

[PR Current Week]

**Common Functions**

**abs**

Returns the absolute value of "numeric_expression". Negative values are returned as positive values.

**Syntax**

\[
\text{abs ( numeric_expression )}
\]

**Example 1**

\[
\text{abs ( 15 )}
\]

**Result**

15

**Example 2**

\[
\text{abs ( -15 )}
\]

**Result**

15

**cast**

Converts "expression" to a specified data type. Some data types allow for a length and precision to be specified. Make sure that the target is of the appropriate type and size. The following can be used for "datatype_specification": character, varchar, char, numeric, decimal, integer, smallint, real, float, date, time, timestamp, time
with time zone, timestamp with time zone, and interval. When type casting to an
interval type, one of the following interval qualifiers must be specified: year,
month, or year to month for the year-to-month interval datatype; day, hour,
minute, second, day to hour, day to minute, day to second, hour to minute, hour
to second, or minute to second for the day-to-second interval datatype. Notes:
When you convert a value of type timestamp to type date, the time portion of the
timestamp value is ignored. When you convert a value of type timestamp to type
time, the date portion of the timestamp is ignored. When you convert a value of
type date to type timestamp, the time components of the timestamp are set to zero.
When you convert a value of type time to type timestamp, the date component is
set to the current system date. It is invalid to convert one interval datatype to the
other (for instance because the number of days in a month is variable). Note that
you can specify the number of digits for the leading qualifier only, i.e. YEAR(4) TO
MONTH, DAY(5). Errors will be reported if the target type and size are not
compatible with the source type and size.

Syntax

cast ( expression , datatype_specification )

Example 1

cast ( '123' , integer )

Result

123

Example 2

cast ( 12345 , varchar ( 10 ) )

Result

a string containing 12345

ceil

Returns the smallest integer that is greater than or equal to "numeric_expression".

Syntax

ceil ( numeric_expression )

ceiling

Returns the smallest integer that is greater than or equal to "numeric_expression".

Syntax

ceiling ( numeric_expression )

Example 1

ceiling ( 4.22 )

Result

5

Example 2

ceiling ( -1.23 )
**char_length**

Returns the number of logical characters in "string_expression". The number of logical characters can be distinct from the number of bytes in some East Asian locales.

**Syntax**

```sql
char_length ( string_expression )
```

**Example**

```sql
char_length ( 'Canada' )
```

**Result**

6

**character_length**

Returns the number of characters in "string_expression".

**Syntax**

```sql
character_length ( string_expression )
```

**Example**

```sql
character_length ( 'Canada' )
```

**Result**

6

**coalesce**

Returns the first non-null argument (or null if all arguments are null). Requires two or more arguments in "expression_list".

**Syntax**

```sql
coalesce ( expression_list )
```

**Example**

```sql
coalesce ( [Unit price], [Unit sale price] )
```

**Result**

Returns the unit price, or the unit sale price if the unit price is null.

**current_date**

Returns a date value representing the current date of the computer that the database software runs on.

**Syntax**

```sql
current_date
```
Example

current_date

Result

2003-03-04

**current_time**

Returns a time with time zone value, representing the current time of the computer that runs the database software if the database supports this function. Otherwise, it represents the current time of the computer that runs IBM® Cognos® BI software.

Syntax

current_time

Example

current_time

Result

16:33:11.354+05:00

**current_timestamp**

Returns a datetime with time zone value, representing the current time of the computer that runs the database software if the database supports this function. Otherwise, it represents the current time of the computer that runs IBM® Cognos® BI software.

Syntax

current_timestamp

Example

current_timestamp

Result

2003-03-03 16:40:15.535+05:00

**exp**

Returns 'e' raised to the power of "numeric_expression". The constant 'e' is the base of the natural logarithm.

Syntax

exp ( numeric_expression )

Example

exp ( 2 )

Result

7.389056
**extract**

Returns an integer representing the value of datepart (year, month, day, hour, minute, second) in "datetime_expression".

**Syntax**

\[
\text{extract ( datepart , datetime_expression )}
\]

**Example 1**

\[
\text{extract ( year , 2003-03-03 16:40:15.535 )}
\]

**Result**

2003

**Example 2**

\[
\text{extract ( hour , 2003-03-03 16:40:15.535 )}
\]

**Result**

16

**floor**

Returns the largest integer that is less than or equal to "numeric_expression".

**Syntax**

\[
\text{floor ( numeric_expression )}
\]

**Example 1**

\[
\text{floor ( 3.22 )}
\]

**Result**

3

**Example 2**

\[
\text{floor ( -1.23 )}
\]

**Result**

-2

**ln**

Returns the natural logarithm of "numeric_expression".

**Syntax**

\[
\text{ln ( numeric_expression )}
\]

**Example**

\[
\text{ln ( 4 )}
\]

**Result**

1.38629
**localtime**

Returns a time value, representing the current time of the computer that runs the database software.

**Syntax**

```
localtime
```

**Example**

```
localtime
```

**Result**

16:33:11

---

**localtimestamp**

Returns a datetime value, representing the current timestamp of the computer that runs the database software.

**Syntax**

```
localtimestamp
```

**Example**

```
localtimestamp
```

**Result**

2003-03-03 16:40:15

---

**lower**

Returns "string_expression" with all uppercase characters shifted to lowercase.

**Syntax**

```
lower ( string_expression )
```

**Example**

```
lower ( 'ABCDEF' )
```

**Result**

`abcdef`

---

**mod**

Returns the remainder (modulus) of "integer_expression1" divided by "integer_expression2". "Integer_expression2" must not be zero or an exception condition is raised.

**Syntax**

```
mod ( integer_expression1, integer_expression2 )
```

**Example**

```
mod ( 20, 3 )
```

**Result**

718
nullif
Returns null if "expression1" equals "expression2", otherwise returns "expression1".

Syntax
nullif ( expression1, expression2 )

octet_length
Returns the number of bytes in "string_expression".

Syntax
octet_length ( string_expression )

Example 1
octet_length ( 'ABCDEF' )
Result
6

Example 2
octet_length ( '' )
Result
0

position
Returns the integer value representing the starting position of "string_expression1" in "string_expression2" or 0 when the "string_expression1" is not found.

Syntax
position ( string_expression1, string_expression2 )

Example 1
position ( 'C', 'ABCDEF' )
Result
3

Example 2
position ( 'H', 'ABCDEF' )
Result
0

power
Returns "numeric_expression1" raised to the power "numeric_expression2". If "numeric_expression1" is negative, then "numeric_expression2" must result in an integer value.
Syntax
power ( numeric_expression1, numeric_expression2 )

Example
power ( 3, 2 )

Result
9

_round
Returns "numeric_expression" rounded to "integer_expression" places to the right of the decimal point. Notes: "integer_expression" must be a non-negative integer. Rounding takes place before data formatting is applied.

Syntax
_round ( numeric_expression, integer_expression )

Example
_round ( 1220.42369, 2 )

Result
1220.42

row
The row constructor represents a collection of values organized as a row of data. It can be used in conditional expressions (i.e. IF-THEN-ELSE) and filter expressions (i.e. IN clause).

Syntax
row ( expression_list )

Example 1
if ( row([RetailerName],[OrderMethodCode]) = row('ActiForme',4) )
then ('A')
else ('B')

Result
Returns 'A' if the Retailer Name is 'ActiForme' and the order method code is 4. Otherwise, the value 'B' is returned.

Example 2
case row([RetailerName],[OrderMethodCode])
when row('Advanced Climbing Ltd',3) then 1
when row('ActiForme',5) then 2
else 3
end

Result
Returns 1 if the Retailer Name is 'Advanced Climbing Ltd' and the order method code is 3. Returns 2 if the Retailer Name is 'ActiForme' and the order method code is 5. Otherwise, the value 3 is returned.

**Example 3**

```
row ( [OrderMethodCode], [Year] ) in ( [Query].[OMC], [Query].[YR] )
```

**Result**

The returned data is filtered on the following two conditions: 1) [OrderMethodCode] in ([Query].[OMC]) 2) [Year] in ([Query].[YR])

**sqrt**

Returns the square root of "numeric_expression". "Numeric_expression" must be non-negative.

**Syntax**

```
sqrt ( numeric_expression )
```

**Example**

```
sqrt ( 9 )
```

**Result**

```
3
```

**substring**

Returns the substring of "string_expression" that starts at position "integer_expression1" for "integer_expression2" characters or to the end of "string_expression" if "integer_expression2" is omitted. The first character in "string_expression" is at position 1.

**Syntax**

```
substring ( string_expression , integer_expression1 [ , integer_expression2 ] )
```

**Example**

```
substring ( 'abcdefg', 3, 2 )
```

**Result**

```
cd
```

**trim**

Returns "string_expression" trimmed of leading and trailing blanks or trimmed of a certain character specified in "match_character_expression". "Both" is implicit when the first argument is not stated and blank is implicit when the second argument is not stated.

**Syntax**

```
trim ( [ [ trailing|leading|both ] [ match_character_expression ] , ] string_expression )
```

**Example 1**
trim ( trailing 'A' , 'ABCDEF' )

Result

ABCDEF

Example 2

trim ( both , ' ABCDEF ' )

Result

ABCDEF

upper

Returns "string_expression" with all lowercase characters converted to uppercase.

Syntax

upper ( string_expression )

Example

upper ( 'abcdef' )

Result

ABCDEF

width-bucket

For a given expression, this function returns the bucket number into which the value of this expression would fall after being evaluated.

Syntax

width-bucket ( numeric_expression , min_value , max_value , num_of_buckets )

Example

width-bucket ( Quantity , 100 , 5000 , 10 )

Result

For each row, returns the bucket number (from 0 to 11) for the current Quantity value.

Result data

<table>
<thead>
<tr>
<th>Quantity</th>
<th>width-bucket (Quantity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>450</td>
<td>1</td>
</tr>
<tr>
<td>1400</td>
<td>3</td>
</tr>
<tr>
<td>3600</td>
<td>8</td>
</tr>
<tr>
<td>4900</td>
<td>10</td>
</tr>
<tr>
<td>5000</td>
<td>11</td>
</tr>
</tbody>
</table>
**Trigonometric functions**

**arccos**
This inverse trigonometric function returns the arc cosine of the argument, where the argument is in the range of -1 to 1 and the result is a value expressed in radians.

**Syntax**
arccos ( numeric_expression )

**Example**
arccos ( -1 )

**Result**
3.1415

**arcsin**
This inverse trigonometric function returns the arc sine of the argument, where the argument is in the range of -1 to 1 and the result is a value expressed in radians.

**Syntax**
arcsin ( numeric_expression )

**Example**
arcsin ( 0 )

**Result**
3.1415

**arctan**
This inverse trigonometric function returns the arc tangent of the argument, where the argument is in the range of -1 to 1 and the result is a value expressed in radians.

**Syntax**
arctan ( numeric_expression )

**Example**
arctan ( 0 )

**Result**
3.1415

**cos**
This trigonometric function returns the cosine of the argument, where the argument is an angle expressed in radians.

**Syntax**
cos ( numeric_expression )

**Example**
cos ( 0.3333 * 3.1415 )
Result

0.5

coshyp
This trigonometric function returns the hyperbolic cosine of the argument, where the argument is an angle expressed in radians.

Syntax
coshyp ( numeric_expression )

Example
coshyp ( 0 )

Result

1

sin
This trigonometric function returns the sine of the argument, where the argument is an angle expressed in radians.

Syntax
sin ( numeric_expression )

Example
sin ( 0.1667 * 3.1415 )

Result

0.5

sinhyp
This trigonometric function returns the hyperbolic sine of the argument, where the argument is an angle expressed in radians.

Syntax
sinhyp ( numeric_expression )

Example
sinhyp ( 0 )

Result

0

tan
This trigonometric function returns the tangent of the argument, where the argument is an angle expressed in radians.

Syntax
tan ( numeric_expression )

Example
tan ( 0.25 * 3.1415 )
**tanhyp**
This trigonometric function returns the hyperbolic tangent of the argument, where the argument is an angle expressed in radians.

**Syntax**
tanhyp ( numeric_expression )

**Example**
tanhyp ( 0 )

**Result**
0

---

**Dimensional Functions**

**ancestor**
Returns the ancestor of "member" at "level" or at "integer" number of levels above "member". Note: The result is not guaranteed to be consistent when there is more than one such ancestor.

**Syntax**
ancestor ( member, level|integer )

**Example 1**
ancestor ( [TrailChef Water Bag], 1 )

**Result**
Cooking Gear

**Example 2**
ancestor ( [TrailChef Water Bag], 2 )

**Result**
Camping Equipment

**Example 3**
ancestor ( [TrailChef Water Bag], [great_outdoors_company].[Products].[Products].[Product type] )

**Result**
Cooking Gear

**ancestors**
Returns all the ancestors of "member" at "level" or "index" distance above the member. (Most data sources support only one ancestor at a specified level. If the data source supports more than one ancestor, the result is a member set.)
ancestors ( member , level|index )

Example 1
ancestors ( [TrailChef Water Bag] , 1 )

Result
Cooking Gear

Example 2
ancestors ( [TrailChef Water Bag] , 2 )

Result
Camping Equipment

Example 3

Result
Cooking Gear

bottomCount
Sorts a set according to the value of "numeric_expression" evaluated at each of the members of "set_expression" and returns the bottom "index_expression" members.

Syntax
bottomCount ( set_expression , index_expression , numeric_expression )

Example

Result
Returns the bottom two members of the set sorted by revenue.

Result data

<table>
<thead>
<tr>
<th>Product line</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outdoor Protection</td>
<td>$3,171,114.92</td>
</tr>
<tr>
<td>Mountaineering Equipment</td>
<td>$20,891,350.60</td>
</tr>
</tbody>
</table>

bottomPercent
Sorts the set specified in "set_expression" in ascending order and returns the bottommost elements from the sorted set whose cumulative percentage of the total is greater than or equal to "percentage".

Syntax
bottomPercent ( set_expression , percentage , numeric_expression )
Example
bottomPercent ( set ( [Camping Equipment] , [Golf Equipment] ,

Result
For the set of Camping Equipment, Golf Equipment, and Mountaineering
Equipment, returns the members with the smallest Gross profit whose total for the
year 2006 is at least 40% of the overall total.

bottomSum
Sorts the set specified in "set_expression" in ascending order and returns the
bottommost elements from the sorted set whose cumulative total is greater than or
equal to "value".

Syntax
caption ( level|member|set_expression )
Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment

children

Returns the set of children of a specified member.

Syntax
children ( member )

Example
children ( [Camping Equipment] )

Result
Returns the set of children for Camping Equipment.

Result data
Cooking Gear
Tents
Sleeping Bags
Packs
Lanterns

closingPeriod

Returns the last sibling member among the descendants of a member at "level". This function is typically used with a time dimension.

Syntax
closingPeriod ( level [ , member ] )

Example 1
closingPeriod ( [great_outdoors_company].[Years].[Years].[Month] )

Result
2006/Dec

Example 2
closingPeriod ( [great_outdoors_company].[Years].[Years].[Year] )

Result
2006

Example 3
closingPeriod ( [great_outdoors_company].[Years].[Years].[Month] , [2006 Q 4] )

Result
2006/Dec
cousin

Returns the child member of "member2" with the same relative position as "member1" to its parent. This function appears in the Revenue by GO Subsidiary 2005 sample report in the GO Data Warehouse (analysis) package.

Syntax
cousin ( member1 , member2 )

Example 1
cousin ( [Irons] , [Camping Equipment] )

Result
Cooking Gear

Example 2
cousin ( [Putters] , [Camping Equipment] )

Result
Sleeping Bags

completeTuple

Identifies a cell location (intersection) based on the specified members, each of which must be from a different dimension. However, completeTuple () implicitly includes the default member from all dimensions not otherwise specified in the arguments, rather than the current member. CompleteTuple will use the default measure rather than the currentMeasure in the query if the measure is not defined in the completetuple function. This function appears in the Planned Headcount sample report in the GO Data Warehouse (analysis) package.

Syntax
completeTuple ( member { , member } )

Example 1
completeTuple ( [Mountaineering Equipment] , [Fax] )

Result

The completeTuple does not pick up the currentMember by default as the tuple function does. The values in the first column are identical across each year because the default member of the Years dimension, the root member, is used rather than the current member. Likewise, the first column displays Revenue rather than Quantity Sold because the Revenue measure is the default from the Measures dimension. CompleteTuple will use the default measure rather than the currentMeasure in the query if the measure is not defined in the completetuple function.

Result data

<table>
<thead>
<tr>
<th>Quantity Sold</th>
<th>Mountaineering Sales by Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>$1,220,329.38</td>
</tr>
<tr>
<td>2005</td>
<td>$1,220,329.38</td>
</tr>
<tr>
<td>2006</td>
<td>$1,220,329.38</td>
</tr>
</tbody>
</table>
Example 2

```
completeTuple ( [Mountaineering Equipment], [Fax], [Quantity sold],
currentMember ( [great_outdoors_company].[Years].[Years] ) )
```

**Result**

The `completeTuple` function uses the `currentMember` of the Years dimension and the Quantity sold measure.

**Result data**

<table>
<thead>
<tr>
<th>Quantity Sold</th>
<th>Mountaineering Sales by Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>0</td>
</tr>
<tr>
<td>2005</td>
<td>8,746</td>
</tr>
<tr>
<td>2006</td>
<td>7,860</td>
</tr>
</tbody>
</table>

**currentMember**

Returns the current member of the hierarchy during an iteration. If "hierarchy" is not present in the context in which the expression is being evaluated, its default member is assumed. This function appears in the Rolling and Moving Averages interactive sample report.

**Syntax**

```
currentMember ( hierarchy )
```

**defaultMember**

Returns the default member of "hierarchy".

**Syntax**

```
defaultMember ( hierarchy )
```

**Example 1**

```
defaultMember ( [great_outdoors_company].[Products].[Products] )
```

**Result**

Products

**Example 2**

```
defaultMember ( [great_outdoors_company].[Years].[Years] )
```

**Result**

Year

**Example 3**

```
defaultMember ( hierarchy ( [great_outdoors_company].[Measures].[Quantity sold] ) )
```

**Result**

Revenue
descendants

Returns the set of descendants of "member" or "set_expression" at "level" (qualified name) or "distance" (integer 0..n) from the root. Multiple options may be specified (separated by a space) to determine which members are returned. self: Only the members at the specified level are included in the final set (this is the default behaviour in the absence of any options). before: If there are any intermediate levels between the member's level and the one specified, members from those levels are included. If the level specified is the same as the member upon which the function is applied, the member is included in the final set. beforewithmember: If there are any intermediate levels between the member's level and the one specified, members from those levels are included. The member upon which the function is applied is also included in the final set. after: If other levels exist after the specified level, members from those levels are included in the final set. This function appears in the Sales Commissions for Central Europe sample report in the GO Data Warehouse (analysis) package.

Syntax

descendants ( member|set_expression , level|distance [ ,
{ self|before|beforewithmember|after } ] )

Example 1

descendants ( [great_outdoors_company].[Products].[Products].
[Products] , [great_outdoors_company].[Products].[Products].[Product
type] )

Result

Returns the set of descendants of the Products set at the Product type level. Note: [great_outdoors_company].[Products].[Products].[Products] is the root member of the Products hierarchy.

Result data
Cooking Gear
Sleeping Bags
Packs
Tents
...
Eyewear
Knives
Watches

Example 2

descendants ( [great_outdoors_company].[Products].[Products].
[Products] , 1 )

Result

Returns the set of descendants of the Products set at the first level.

Result data
Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories
descendants ([great_outdoors_company].[Products].[Products].
[Products], 3, before)

Result

Returns the descendants of the Products set before the third level.

Result data
Camping Equipment
Cooking Gear
Sleeping Bags
Packs
Tents
...
Eyewear
Knives
Watches

Example 4
descendants ([great_outdoors_company].[Products].[Products].
[Products], 2, self before)

Result

Returns the set of descendants of the Products set before and including the second
level.

Result data
Camping Equipment
Cooking Gear
Sleeping Bags
Packs
Tents
...
Eyewear
Knives
Watches

except

Returns the members of "set_expression1" that are not also in "set_expression2".
Duplicates are retained only if the optional keyword "all" is supplied as the third
argument.

Syntax
except ( set_expression1 , set_expression2 [ , all ] )

Example
except ( set ([Camping Equipment] , [Mountaineering Equipment] ) ,
set ([Camping Equipment] , [Golf Equipment] ) )

Result
Mountaineering Equipment
filter

Returns the set resulting from filtering a specified set based on the Boolean condition. Each member is included in the result if and only if the corresponding value of "Boolean_expression" is true.

**Syntax**

filter ( set_expression , Boolean_expression )

**Example**

filter ( [Product line] , [Gross margin] > .30 )

**Result**

Mountaineering Equipment

firstChild

Returns the first child of "member".

**Syntax**

firstChild ( member )

**Example 1**

firstChild ( [By Product Lines] )

**Result**

Camping Equipment

**Example 2**

firstChild ( [Camping Equipment] )

**Result**

Cooking Gear

firstSibling

Returns the first child of the parent of "member".

**Syntax**

firstSibling ( member )

**Example 1**

firstSibling ( [Outdoor Protection] )

**Result**

Camping Equipment

**Example 2**

firstSibling ( [Camping Equipment] )

**Result**
Camping Equipment

_format

Associates a format with the expression. The format_keyword can be PERCENTAGE_0, PERCENTAGE_1, or PERCENTAGE_2. PERCENTAGE_1 returns a percentage with one digit to the right of the decimal point, PERCENTAGE_2 returns a percentage with two digits to the right of the decimal point, and PERCENTAGE_3 returns a percentage value out of one with three digits to the right of the decimal point (for example, 0.965).

Syntax
_format ( expression , format_keyword )

Example
_format ( [Unit Sale Price] / [Unit Price] , PERCENTAGE_2 )

Result
75.12%

emptySet

Returns an empty member set for "hierarchy". This is most often used as a placeholder during development or with dynamic report design (either with the IBM® Cognos® Software Development Kit or via report design). By creating a data item that contains the emptyset function, it is possible to build complex expressions that can later be revised by redefining the emptyset data item.

Syntax
eemptySet ( hierarchy )

Example
eexcept ( [great_outdoors_company].[Products].[Products].[Product line] , emptySet ( [great_outdoors_company].[Products].[Products] ) )

Result
Returns the Product line set and an empty set for the Products set.

Result data
Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories

generate

Evaluates "set_expression2" for each member of "set_expression1" and joins the resulting sets by union. The result retains duplicates only when the optional keyword "all" is supplied as the third argument.

Syntax
generate ( set_expression1 , set_expression2 [ , all ] )

Example
generate ( [Product line] , topCount ( descendants ( currentMember ( [great_outdoors_company].[Products].[Products] ) , [great_outdoors_company].[Products].[Products].[Product name] ) , 2 , [Revenue] ) )

Result
Returns the top two products by revenue for each product line.

head
Returns the first "index_expression" elements of "set_expression". The default for "index_expression" is 1.

Syntax
head ( set_expression [ , index_expression ] )

Example 1
head ( members ( [great_outdoors_company].[Products].[Products].[Product line] ) )

Result
Camping Equipment

Example 2
head ( members ( [great_outdoors_company].[Products].[Products].[Product line] ) , 2 )

Result
Returns the top two members of the Product line set.

Result data
Camping Equipment
Mountaineering Equipment

hierarchize
Orders the members of "set_expression" in a hierarchy. Members in a level are sorted in their natural order. This is the default ordering of the members along a dimension when no other sort conditions are specified.

Syntax
hierarchize ( set_expression )

Example

Result
Returns Camping Equipment, Golf Equipment, Mountaineering Equipment.
**hierarchy**

Returns the hierarchy that contains "level", "member", or "set_expression".

**Syntax**

hierarchy ( level|member|set_expression )

**Example 1**

hierarchy ( [Cooking Gear] )

**Result**

Returns every member in the hierarchy that contains Cooking Gear.

**Result data**

Products
- Camping Equipment
- Cooking Gear
- TrailChef Water Bag
- TrailChef Canteen
- Mountain Man Extreme
- Mountain Man Deluxe

**Example 2**

hierarchy ( [great_outdoors_company].[Products].[Products].[Product line] )

**Result**

Returns every member in the hierarchy that contains the Product line.

**Result data**

Products
- Camping Equipment
- Cooking Gear
- TrailChef Water Bag
- TrailChef Canteen
- Mountain Man Extreme
- Mountain Man Deluxe

**item**

Returns a member from the "index" location within "set_expression". The index into the set is zero based.

**Syntax**

item ( set_expression , index )

**Example**

item ( children ( [Camping Equipment] ) , 2 )

**Result**

Sleeping Bags
**intersect**

Returns the intersection of "set_expression1" and "set_expression2". The result retains duplicates only when the optional keyword "all" is supplied as the third argument.

**Syntax**

```plaintext
intersect ( set_expression1 , set_expression2 [, all ])
```

**Example**

```plaintext
intersect ( set ( [Camping Equipment] , [Mountaineering Equipment] ) ,
            set ( [Camping Equipment] , [Outdoor Protection] , ) , all )
```

**Result**

Camping Equipment

**lag**

Returns the sibling member that is "index_expression" number of positions prior to "member".

**Syntax**

```plaintext
lag ( member , index_expression )
```

**Example 1**

```plaintext
lag ( [Tents] , 1 )
```

**Result**

Cooking Gear

**Example 2**

```plaintext
lag ( [Tents] , -2 )
```

**Result**

Packs

**lastChild**

Returns the last child of a specified member.

**Syntax**

```plaintext
lastChild ( member )
```

**Example 1**

```plaintext
lastChild ( [Cooking Gear] )
```

**Result**

TrailChef Utensils

**Example 2**

```plaintext
lastChild ( [By Product Line] )
```
Result

Golf Equipment

**lastPeriods**

Returns the set of members from the same level that ends with "member". The number of members returned is the absolute value of "integer_expression". If "integer_expression" is negative, members following and including the specified member are returned. Typically used with a time dimension. This function appears in the Rolling and Moving Averages interactive sample report.

**Syntax**

\[
\text{lastPeriods} \ (integer\_expression, \ member)
\]

**Example 1**

\[
\text{lastPeriods} \ (2, \ [2006 \ Q \ 4])
\]

**Result**

Returns the last two members from the level that ends with 2006 Q 4.

**Result data**

2006 Q 3
2006 Q 4

**Example 2**

\[
\text{lastPeriods} \ (-3, \ [2006 \ Q \ 4])
\]

**Result**

Returns the last three members from the level that starts with 2006 Q 4.

**Result data**

2006 Q 4
2007 Q 1
2007 Q 2

**lastSibling**

Returns the last child of the parent of a specified member.

**Syntax**

\[
\text{lastSibling} \ (member)
\]

**Example**

\[
\text{lastSibling} \ (\text{[Camping Equipment]})
\]

**Result**

Golf Equipment

**lead**

Returns the sibling member that is "index_expression" number of positions after "member". If "index_expression" is negative, returns the sibling member that is "index_expression" number of positions before "member".
Syntax
lead ( member , index_expression )

Example 1
lead ( [Outdoor Protection] , 1 )

Result
Personal Accessories

Example 2
lead ( [Outdoor Protection] , -2 )

Result
Golf Equipment

level
Returns the level of "member".

Syntax
level ( member )

Example
level ( [Golf Equipment] )

Result
Returns the members on the Golf Equipment level.

Result data
Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment

levels
Returns the level in "hierarchy" whose distance from the root is specified by "index".

Syntax
levels ( hierarchy , index )

Example 1
levels ( [great_outdoors_company].[Products].[Products] , 2 )

Result
Returns the members two levels from the root Products hierarchy.

Result data
Cooking Gear
Sleeping Bags
Packs
Tents
...
Irons
Putters
Woods
Golf Accessories

Example 2
levels ( [great_outdoors_company].[Products].[Products] , 1 )

Result

Returns the members one level from the root Products hierarchy.

Result data
Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment

linkMember
Returns the corresponding member in "level" or "hierarchy" (of the same dimension). For level-based hierarchies, a level must be specified as the second argument, and for parent-child hierarchies, a hierarchy must be specified. An exception is thrown when the second parameter does not resolve to a hierarchy of the member's dimension. Note that calculated members are not supported as the first argument.

Syntax
linkMember ( member , level|hierarchy )

members
Returns the set of members in "hierarchy" or "level". In the case of a hierarchy, the order of the members in the result is not guaranteed. If a predictable order is required, an explicit ordering function (such as hierarchize) must be used.

Syntax
members ( hierarchy|level )

Example 1
members ( [great_outdoors_company].[Years].[Years] )

Result

Returns the members in Years.

Example 2
members ( [great_outdoors_company].[Products].[Products].[Product line] )

Result
Returns the members in Product line.

**nextMember**

Returns the next member in the "member" level.

**Syntax**

`nextMember ( member )`

**Example**

`nextMember ( [Outdoor Protection] )`

**Result**

Golf Equipment

**openingPeriod**

Returns the first sibling member among the descendants of a member at "level". This function is typically used with a time dimension.

**Syntax**

`openingPeriod ( level [ , member ] )`

**Example 1**

`openingPeriod ( [great_outdoors_company].[Years].[Years].[Month] )`

**Result**

2004/Jan

**Example 2**

`openingPeriod ( [great_outdoors_company].[Years].[Years].[Year] )`

**Result**

2004

**Example 3**

`openingPeriod ( [great_outdoors_company].[Years].[Years].[Month] , [2006 Q 4] )`

**Result**

2006/Oct

**order**

Arranges the members of "set_expression" according to their "value_expression" and the third parameter. ASC and DESC arrange members in ascending or descending order, respectively, according to their position in the set hierarchy. Then the children of each member are arranged according to "value_expression". BASC and BDESC arrange members in the set without regard to the hierarchy. In the absence of an explicit specification, ASC is the default.

**Syntax**
order ( set_expression, value_expression [ , ASC|DESC|BASC|BDESC ] )

Example 1
order ( members ( [Great Outdoors Company],[Product],[Product].
[Product type] ) , [Quantity sold] , BASC )

Result
Returns the quantity sold for each product type in no particular order.

Result data

<table>
<thead>
<tr>
<th>Product Line</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woods</td>
<td>13,924</td>
</tr>
<tr>
<td>Irons</td>
<td>14,244</td>
</tr>
<tr>
<td>Safety</td>
<td>22,332</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Sunscreen</td>
<td>215,432</td>
</tr>
<tr>
<td>Insect Repellents</td>
<td>270,874</td>
</tr>
<tr>
<td>Lanterns</td>
<td>345,096</td>
</tr>
</tbody>
</table>

Example 2
order ( members ( [Great Outdoors Company],[Product],[Product].
[Product type] ) , [Quantity sold] , ASC )

Result
Returns the quantity sold for each product type in ascending order.

Result data

<table>
<thead>
<tr>
<th>Product Line</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woods</td>
<td>13,924</td>
</tr>
<tr>
<td>Irons</td>
<td>14,244</td>
</tr>
<tr>
<td>Putters</td>
<td>23,244</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Tents</td>
<td>130,664</td>
</tr>
<tr>
<td>Cooking Gear</td>
<td>198,676</td>
</tr>
<tr>
<td>Lanterns</td>
<td>345,096</td>
</tr>
</tbody>
</table>

ordinal
Returns the zero-based ordinal value (distance from the root level) of "level".

Syntax
ordinal ( level )

Example 1
ordinal ( [great_outdoors_company].[Products].[Products].[Product line] )

Result
1
Example 2

\[
\text{ordinal ( [great_outdoors_company].[Products].[Products].[Product type] )}
\]

Result

2

**parallelPeriod**

Returns a member from a prior period in the same relative position as "member". This function is similar to the cousin function, but is more closely related to time series. It takes the ancestor of "member" at "level" (called "ancestor") and the sibling of "ancestor" that lags by "integer_expression" positions, and returns the parallel period of "member" among the descendants of that sibling. When unspecified, "integer_expression" defaults to 1 and "member" defaults to the current member.

**Syntax**

\[
\text{parallelPeriod ( level [, integer_expression [, member ] ] )}
\]

**Example 1**

\[
\text{parallelPeriod ( [great_outdoors_company].[Years].[Years].[Quarter] , -1 , [2006/Aug] )}
\]

Result

2006/Nov

**Example 2**

\[
\text{parallelPeriod ( [great_outdoors_company].[Years].[Years].[Quarter] , 1 , [2006/Aug] )}
\]

Result

2006/May

**Example 3**

\[
\text{parallelPeriod ( [great_outdoors_company].[Years].[Years].[Year] , 2 , [2006/Aug] )}
\]

Result

2004/Aug

**parent**

Returns the member that is the parent of "member" or "measure".

**Syntax**

\[
\text{parent ( member|measure )}
\]

**Example**

\[
\text{parent ( [Cooking Gear] )}
\]

Result
Camping Equipment

**periodsToDate**

Returns a set of sibling members from the same level as "member", as constrained by "level". It locates the ancestor of "member" at "level" and returns that ancestor's descendants at the same level as "member" (up to and including "member"). Typically used with a time dimension. This function appears in the Rolling and Moving Averages interactive sample report.

**Syntax**

```
periodsToDate ( level, member )
```  

**Example**

```
periodsToDate ( [great_outdoors_company].[Years].[Years].[Year], [2004/Mar] )
```  

**Result**

Returns values for [2004/Jan], [2004/Feb], [2004/Mar]

**prevMember**

Returns the member that immediately precedes "member" in the same level. This function appears in the Sales Growth Year Over Year sample report in the GO Data Warehouse (analysis) package.

**Syntax**

```
prevMember ( member )
```  

**Example 1**

```
prevMember ( [Outdoor Protection] )
```  

**Result**

Personal Accessories

**Example 2**

```
prevMember ( [2005] )
```  

**Result**

2004

**member**

Defines a member based on "value_expression" in "hierarchy". "String1" identifies the member created by this function. It must be unique in the query and different from any other member in the same hierarchy. "String2" is the caption of the member; if it is absent, the caption is empty. To ensure predictable results, it is recommended that you supply the "hierarchy". Note: All calculations used as grouping items whose sibling items are other calculations or member sets should be explicitly assigned to a hierarchy using this function. The results are not predictable otherwise. The only exception is where the calculation involves only members of the same hierarchy as the siblings. In this case, the calculation is assumed to belong to that hierarchy.
Syntax

member ( value_expression [ , string1 [ , string2 [ , hierarchy ] ] ] )

Example

member ( total ( currentMeasure within set filter ( [great_outdoors_company].[Products].[Products].[Product name] , caption ( [great_outdoors_company].[Products].[Products].[Product name] ) starts with 'B') ) , 'BProducts' , 'B Products' , [great_outdoors_company].[Products].[Products] )

Result

Returns the quantity sold and revenue for all products that start with the letter B.

nestedSet

This function is intended for use only by Analysis Studio

Syntax

nestedSet ( set_expression1 , set_expression2 )

set

Returns the list of members defined in the expression. The members must belong to the same hierarchy.

Syntax

set ( member { , member } )

Example

set ( [Golf Equipment] , [Irons] , [TrailChef Cup] )

Result

Returns Golf Equipment, Irons, and TrailChef Cup.

siblings

Returns the children of the parent of the specified member.

Syntax

siblings ( member )

Example

siblings ( [Golf Equipment] )

Result

Returns the siblings of Golf Equipment.

Result data

Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories
**tail**

Returns the last “index_expression” elements of "set expression". The default for "index_expression" is 1.

**Syntax**

tail ( set_expression [ , index_expression ] )

**Example 1**

tail (members ( [great_outdoors_company].[Products].[Products].
[Product line] ) )

**Result**

Returns the last member of the Product line set.

**Result data**

Personal Accessories

**Example 2**

tail ( members ( [great_outdoors_company].[Products].[Products].
[Product line] ), 2 )

**Result**

Returns the last two members of the Product line set.

**Result data**

Outdoor Protection
Personal Accessories

**topCount**

Sorts a set according to the values of "numeric_expression" evaluated at each of the members of "set_expression" and returns the top "index_expression" members.

**Syntax**

topCount ( set_expression , index_expression , numeric_expression )

**Example**

topCount ( [great_outdoors_company].[Products].[Products].[Product
line] , 2 , [Revenue] )

**Result**

Returns the top two revenues for the Product line set.

**Result data**

<table>
<thead>
<tr>
<th>Product line</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Equipment</td>
<td>$89,713,990.92</td>
</tr>
<tr>
<td>Personal Accessories</td>
<td>$31,894,465.86</td>
</tr>
</tbody>
</table>
**topPercent**

Sorts the set specified in "set_expression" in descending order and returns the topmost elements from the sorted set whose cumulative percentage of the total is greater than or equal to "percentage".

**Syntax**

topPercent ( set_expression , percentage , numeric_expression2 )

**Example**


**Result**

For the set of Camping Equipment, Golf Equipment, and Mountaineering Equipment, returns the members with the largest Gross profit whose total for the year 2006 is at least 40% of the overall total.

**topSum**

Sorts the set specified in "set_expression" in descending order and returns the topmost elements from the sorted set whose cumulative total is greater than or equal to "value".

**Syntax**

topSum ( set_expression , value , numeric_expression2 )

**Example**


**Result**

For the Product line members, returns the members with the largest Gross profit whose total for the year 2006 is at least $6,000,000.

**tuple**

Identifies a cell location (intersection) based on the specified members, each of which must be from a different dimension. This function implicitly includes the current member from all dimensions that are not otherwise specified in the arguments. The current member of any dimension not specified in the evaluating context is assumed to be the default member of that dimension. The value of this cell can be obtained with the "value" function.

**Syntax**

tuple ( member { , member } )

**Example**

tuple ( [Mountaineering Equipment] , [Fax] )

**Result**

Returns the Mountaineering Equipment sales by fax.
union

Returns data for "set_expression1" and "set_expression2". The result retains duplicates only when the optional keyword "all" is supplied as the third argument.

Syntax

union ( set_expression1 , set_expression2 [ , all ] )

Example 1


Result

Returns data for both sets as one new set, showing the Golf Equipment column only once.

Example 2


Result

Returns data for both sets as one new set, showing the Golf Equipment column twice.

roleValue

Returns the value of the attribute that is associated with the role whose name is specified by "string" within the specified context. "Member" or "set_expression" is optional only in a number of limited circumstances, where it can be derived from another context. Applications can be made portable across different data sources and models by accessing attributes by role rather than by query item ID. For dimensionally-modeled relational (DMR) data sources, assignment of roles is the modeler's responsibility. Intrinsic roles that are defined for members of all data source types include: '_businessKey', '_memberCaption', '_memberDescription', '_memberUniqueName'. Additional roles can be defined in Framework Manager for each level in a hierarchy. For example, a Product type level may have an attribute column called "Type Shipping Container", and the Product level may have a "Product Shipping Container" attribute. Each of these could be assigned a custom role in Framework Manager called "Container". The property could then be referenced independently of the actual column name by using the roleValue function.

Syntax

roleValue ( string [ , member|set_expression ] )

Example 1

roleValue ( 'memberCaption' , [Sales].[Product].[Product].[Product line] -> [all].[1] )

Result

Camping Equipment

Example 2
roleValue ( '_businessKey' , [great_outdoors_company].[Years].[Years].[Year] )

Result

Returns the value of the attribute that is associated with the business key role.

Result data

("2004-01-01","2004-12-31")
("2005-01-01","2005-12-31")
("2006-01-01","2006-12-31")

Example 3
roleValue ( '_memberUniqueName' , [great_outdoors_company].[Years].[Years].[Year] )

Result

Returns the value of the attribute that is associated with the MUN role.

Result data

[great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (Root)].[20040101-20041231]
[great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (Root)].[20050101-20051231]
[great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (Root)].[20060101-20061231]

rootMember

Returns the root member of a single-root hierarchy. This function appears in the Promotion Success sample report in the GO Data Warehouse (analysis) package.

Syntax

rootMember ( hierarchy )

rootMembers

Returns the root members of a hierarchy.

Syntax

rootMembers ( hierarchy )

Example

rootMembers ( [great_outdoors_company].[Years].[Years] )

Result

By Time

subset

Returns a subset of members in "set_expression" starting at "index_expression1" from the beginning. If the count "index_expression2" is specified, that many members are returned (if available). Otherwise, all remaining members are returned.

Syntax
subset ( set_expression, index_expression1 [ , index_expression2 ] )

Example 1
subset ( members ( [great_outdoors_company].[Products].[Products].[Product line] ), 2 )

Result
Returns the members of the Product line set starting at the second member.

Result data
Mountaineering Equipment
Outdoor Protection
Personal Accessories

Example 2
subset ( members ( [great_outdoors_company].[Products].[Products].[Product line] ), 2 , 2 )

Result
Returns two members of the Product line set starting at the second member.

Result data
Mountaineering Equipment
Outdoor Protection

unique
Removes all duplicates from "set_expression". The remaining members retain their original order.

Syntax
unique ( set_expression )

value
Returns the value of the cell identified by "tuple". Note that the default member of the Measures dimension is the Default Measure.

Syntax
value ( tuple )

Example 1
value ( tuple ( [great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (Root)].[20040101-20041231] , [great_outdoors_company].[Measures].[Revenue] ) )

Result
$34,750,563.50

Example 2
value ( tuple ( [2004] , [Camping Equipment] , [Revenue] ) )

Result
Report Functions

Today
Returns the current system date.

Syntax
Today ()

Now
Returns the current system time.

Syntax
Now ()

AsOfDate
Returns the date value of the AsOfDate expression, if it is defined. Otherwise, AsOfDate returns the report execution date.

Syntax
AsOfDate ()

AsOfTime
Returns the time value of the AsOfTime expression, if it is defined. Otherwise, AsOfTime returns the report execution time.

Syntax
AsOfTime ()

ReportDate
Returns the report execution date and time.

Syntax
ReportDate ()

ReportName
Returns the report name. This function works only when the report is run from IBM® Cognos® Connection.

Syntax
ReportName ()

ReportPath
Returns the report path. This function works only when the report is run from IBM® Cognos® Connection.

Syntax
ReportPath ()
**ReportDescription**

Returns the report description. This function works only when the report is run from IBM® Cognos® Connection.

**Syntax**

```
ReportDescription()
```

**ReportLocale**

Returns the run locale.

**Syntax**

```
ReportLocale()
```

**GetLocale**

Returns the run locale (deprecated).

**Syntax**

```
GetLocale()
```

**Locale**

Returns the run locale.

**Syntax**

```
Locale()
```

**ReportProductLocale**

Returns the product locale.

**Syntax**

```
ReportProductLocale()
```

**ReportAuthorLocale**

Returns the author locale.

**Syntax**

```
ReportAuthorLocale()
```

**ReportSaveDate**

Returns the date when the report was last saved.

**Syntax**

```
ReportSaveDate()
```

**ReportCreateDate**

Returns the date when the report was created.

**Syntax**

```
ReportCreateDate()
```
ReportID

Returns the report ID.

Syntax
ReportID ()

ReportOutput

Returns the name of the output format, such as CSV, HTML, layoutDataXML, MHT, PDF, rawXML, singleXLS, spreadsheetML, XLS, XML, or XLWA.

Syntax
ReportOutput ()

ReportOption

Returns the value of the run option variable identified by "optionName", such as attachmentEncoding, burst, cssURL, email, emailAsAttachment, emailAsURL, emailBody, emailSubject, emailTo, emailToAddress, metadataModel, outputEncapsulation, outputFormat, outputLocale, outputPageDefinition, outputPageOrientation, primaryWaitThreshold, print, printer, printerAddress, prompt, promptFormat, saveAs, saveOutput, secondaryWaitThreshold, verticalElements, or xslURL.

Syntax
ReportOption ('optionName')

ServerName

Returns the name of the web server where the run request originated from. The value may be empty if the request is executed from the scheduler.

Syntax
ServerName ()

ServerLocale

Returns the locale of the server that runs the report.

Syntax
ServerLocale ()

ModelPath

Returns the model path.

Syntax
ModelPath ()

BurstKey

Returns the burst key.

Syntax
BurstKey ()
**BurstRecipients**
Returns the distribution list of burst recipients.

Syntax
BurstRecipients ()

**IsBursting**
Returns Boolean 1 (true) when the report will be distributed to the recipient; otherwise, returns Boolean 0 (false).

Syntax
IsBursting ('recipientName')

**ParamNames**
Returns all parameter names.

Syntax
ParamNames ()

**ParamName**
Returns the parameter name of "parameterName".

Syntax
ParamName ('parameterName')

**ParamDisplayValue**
Returns a string that is the parameter display value of "parameterName". This function appears in the Recruitment Report sample report in the GO Data Warehouse (analysis) package.

Syntax
ParamDisplayValue ('parameterName')

**ParamValue**
Returns the parameter value of "parameterName".

Syntax
ParamValue ('parameterName')

**ParamCount**
Returns the parameter count of "parameterName".

Syntax
ParamCount ('parameterName')

**RowNumber**
Returns the current row.

Syntax
RowNumber ()
**PageNumber**

Returns the current page number.

**Syntax**

PageNumber ()

---

**PageCount**

Returns the current page count. This function works only when the report output is Adobe® PDF or Microsoft® Excel. If you save the report output, this function works for all formats.

**Syntax**

PageCount ()

---

**IsPageCountAvailable**

Returns Boolean 1 (true) if the page count is available for the current execution of the report; otherwise, returns Boolean 0 (false).

**Syntax**

IsPageCountAvailable ()

---

**HorizontalPageNumber**

Returns the current horizontal page number.

**Syntax**

HorizontalPageNumber ()

---

**HorizontalPageCount**

Returns the current horizontal page count.

**Syntax**

HorizontalPageCount ()

---

**PageName**

Returns the current page name.

**Syntax**

PageName ()

---

**URLEncode**

Returns the URL encoded value of the input text.

**Syntax**

URLEncode ('text')

---

**TOCHeadingCount**

Returns the table of contents heading count for a specified heading level.

**Syntax**

TOCHeadingCount ( headingLevel )
**IsAccessible**

Returns Boolean 1 (true) if the report is run with the accessibility features enabled. Use this function as a variable expression with a conditional block to make your reports accessible. For example, you can add a list or crosstab equivalent to a chart in reports that are run with accessibility features enabled.

**Syntax**

IsAccessible()

---

**ColumnNumber**

Returns the current column number.

**Syntax**

ColumnNumber()

---

**IsCrosstabRowNodeMember**

Returns Boolean 1 (true) if the current node is a crosstab row node member.

**Syntax**

IsCrosstabRowNodeMember()

---

**IsCrosstabColumnNodeMember**

Returns Boolean 1 (true) if the current node is a crosstab column node member.

**Syntax**

IsCrosstabColumnNodeMember()

---

**IsInnerMostCrosstabRowNodeMember**

Returns Boolean 1 (true) if the current node is an innermost crosstab row node member.

**Syntax**

IsInnerMostCrosstabRowNodeMember()

---

**IsInnerMostCrosstabColumnNodeMember**

Returns Boolean 1 (true) if the current node is an innermost crosstab column node member.

**Syntax**

IsInnerMostCrosstabColumnNodeMember()

---

**IsOuterMostCrosstabRowNodeMember**

Returns Boolean 1 (true) if the current node is an outermost crosstab row node member.

**Syntax**

IsOuterMostCrosstabRowNodeMember()

---

**IsOuterMostCrosstabColumnNodeMember**

Returns Boolean 1 (true) if the current node is an outermost crosstab column node member.
Syntax
IsOuterMostCrosstabColumnNodeMember ()

IsFirstColumn
Returns Boolean 1 (true) if the current column is the first column.

Syntax
IsFirstColumn ()

IsLastColumn
Returns Boolean 1 (true) if the current column is the last column.

Syntax
IsLastColumn ()

IsLastInnerMostCrosstabColumnNodeMember
Returns Boolean 1 (true) if the current node is the last innermost crosstab column node member.

Syntax
IsLastInnerMostCrosstabColumnNodeMember ()

IsLastInnerMostCrosstabRowNodeMember
Returns Boolean 1 (true) if the current node is the last innermost crosstab row node member.

Syntax
IsLastInnerMostCrosstabRowNodeMember ()

CubeName
Returns the name of the cube. "Dimension" specifies from which cube to retrieve the metadata.

Syntax
CubeName ( dimension )

CubeDescription
Returns the description of the cube. "Dimension" specifies from which cube to retrieve the metadata.

Syntax
CubeDescription ( dimension )

CubeCreatedOn
Returns the date and time when the cube was created. "Dimension" specifies from which cube to retrieve the metadata. If the dimension source is an IBM® Cognos® PowerCube (.mdc), the function returns a blank string because the initial creation date of a PowerCube is not maintained.

Syntax
CubeCreatedOn ( dimension )
**CubeDataUpdatedOn**
Returns the date time that data in the cube was last updated. "Dimension" specifies from which cube to retrieve the metadata.

**Syntax**
```
CubeDataUpdatedOn ( dimension )
```

**CubeSchemaUpdatedOn**
Returns the date time that the cube schema was last updated. "Dimension" specifies from which cube to retrieve the metadata.

**Syntax**
```
CubeSchemaUpdatedOn ( dimension )
```

**CubeIsOptimized**
Returns "true" if the cube is optimized. "Dimension" specifies from which cube to retrieve the metadata.

**Syntax**
```
CubeIsOptimized ( dimension )
```

**CubeDefaultMeasure**
Returns the name of the default measure for the cube. "Dimension" specifies from which cube to retrieve the metadata.

**Syntax**
```
CubeDefaultMeasure ( dimension )
```

**CubeCurrentPeriod**
Returns the current period for the cube. "Dimension" specifies from which cube to retrieve the metadata.

**Syntax**
```
CubeCurrentPeriod ( dimension )
```

**CellValue**
Returns the value of the current crosstab cell.

**Syntax**
```
CellValue ()
```

**InScope**
Returns Boolean 1 (true) when the cell is in the scope of the data items and MUNs; otherwise, returns Boolean 0 (false).

**Syntax**
```
InScope ( dataItem, MUN, ... )
```
Data Type Casting Functions

_add_days
Returns the datetime resulting from adding "integer_expression" days to "timestamp_expression".

Syntax
_add_days ( timestamp_expression, integer_expression )

Example
_add_days ( 2007-01-14 00:00:00.000, 3 )

Result
2007-01-17 00:00:00.000

_add_months
Returns the datetime resulting from adding "integer_expression" months to "timestamp_expression".

Syntax
_add_months ( timestamp_expression, integer_expression )

_add_years
Returns the datetime resulting from adding "integer_expression" years to "timestamp_expression".

Syntax
_add_years ( timestamp_expression, integer_expression )

_age
Returns a number by subtracting "timestamp_expression" from today's date.

Syntax
_age ( timestamp_expression )

Example
_age ([Query1].[Date]), where [Query1].[Date] is March 2, 2004, and today is July 8, 2009

Result
50,406, where 5 is the number of years, 04 is the number of months, and 06 is the number of days.

_day_of_week
Returns the day of the week (between 1 and 7) for "timestamp_expression" where "integer_expression" indicates which day of that week is day 1. To determine "integer_expression", choose the day of the week and count from Monday; for example, if you choose Wednesday, "integer_expression" would be 3 because Wednesday is the third day from Monday.
**Syntax**
_day_of_week ( timestamp_expression , integer_expression )

**Example**
_day_of_week ( 2009-01-01 , 7 ), where 7 means that Sunday is the first day of the week.

**Result**
5

**_day_of_year**
Returns the ordinal for the day of the year in "timestamp_expression" (1 to 366). Also known as Julian day.

**Syntax**
_day_of_year ( timestamp_expression )

**_days_between**
Returns a positive or negative number representing the number of days between "timestamp_expression1" and "timestamp_expression2". If "timestamp_expression1" < "timestamp_expression2", the result will be a negative number.

**Syntax**
_days_between ( timestamp_expression1 , timestamp_expression2 )

**_days_to_end_of_month**
Returns a number representing the number of days remaining in the month represented by "timestamp_expression".

**Syntax**
_days_to_end_of_month ( timestamp_expression )

**_first_of_month**
Returns a datetime that is the first day of the month represented by "timestamp_expression".

**Syntax**
_first_of_month ( timestamp_expression )

**Example 1**
_first_of_month ( 2009-05-04 00:00:00.000 )

**Result**
Returns 2009-05-01 00:00:00.000

**Example 2**
_first_of_month ( current_date )

**Result**
Returns Jul 1, 2009 if the current date is July 30, 2009.
_last_of_month

Returns a datetime that is the last day of the month represented by "timestamp_expression".

Syntax

_last_of_month ( timestamp_expression )

_make_timestamp

Returns a timestamp constructed from "integer_expression1" (the year), "integer_expression2" (the month), and "integer_expression3" (the day). The time portion defaults to 00:00:00.000.

Syntax

_make_timestamp ( integer_expression1 , integer_expression2 , integer_expression3 )

_months_between

Returns a positive or negative number representing the number of months between "timestamp_expression1" and "timestamp_expression2". If "timestamp_expression1" < "timestamp_expression2", the result will be a negative number.

Syntax

_months_between ( timestamp_expression1 , timestamp_expression2 )

_week_of_year

Returns the week number (1-53) of the year for "timestamp_expression". According to the ISO 8601, week 1 of the year is the first week to contain a Thursday, which is equivalent to the first week containing January 4th. A week starts on a Monday (day 1) and ends on a Sunday (day 7).

Syntax

_week_of_year ( timestamp_expression )

_years_between

Returns a positive or negative integer representing the number of years between "timestamp_expression1" and "timestamp_expression2". If "timestamp_expression1" < "timestamp_expression2", a negative value is returned.

Syntax

_years_between ( timestamp_expression1 , timestamp_expression2 )

_ymdint_between

Returns a number representing the difference between "timestamp_expression1" and "timestamp_expression2". This value has the form YYMMDD, where YY represents the number of years, MM represents the number of months, and DD represents the number of days.

Syntax

_ymdint_between ( timestamp_expression1 , timestamp_expression2 )

Example
_ymdint_between ( [Query1].[Date (close date)] , [Query1].[Date (ship date)] ), where [Query1].[Date (close date)] is February 20, 2004, and [Query1].[Date (ship date)] is January 19, 2004.

Result
101, where 1 is the number of months and 01 is the number of days.

abs
Returns the absolute value of "numeric_expression". If "numeric_expression" is negative, a positive value is returned.

Syntax
abs ( numeric_expression )

ceiling
Returns the smallest integer that is greater than or equal to "numeric_expression".

Syntax
ceiling ( numeric_expression )

character_length
Returns the number of characters in "string_expression".

Syntax
character_length ( string_expression )

date2string
Returns a date as a string in YYYY-MM-DD format.

Syntax
date2string ( date_expression )

date2timestamp
Converts "date_expression" to a timestamp. The time part of the timestamp will equal zero.

Syntax
date2timestamp ( date_expression )

date2timestampTZ
Converts "date_expression" to a timestamp with a time zone. The time and time zone parts of the timestamp will equal zero.

Syntax
date2timestampTZ ( date_expression )

DTinterval2string
Returns a date time interval as a string in DDDD HH:MM:SS.FFFFFFF or -DDDD HH:MM:SS.FFF format.

Syntax
DTInterval2string (date_time_interval_expression)

**DTInterval2stringAsTime**

Returns a date time interval as a string in HHHH:MM:SS.FFFFFFF or HH:MM:SS.FFF format. Days are converted to hours.

**Syntax**

```
DTInterval2stringAsTime (date_time_interval_expression)
```

**exp**

Returns the constant 'e' raised to the power of "numeric_expression". The constant 'e' is the base of the natural logarithm.

**Syntax**

```
exp (numeric_expression)
```

**Example**

```
exp (2)
```

**Result**

```
7.389056
```

**extract**

Returns an integer representing the value of "date_part_expression" in "datetime_expression". "Date_part_expression" could be the year, month, day, hour, minute, or second.

**Syntax**

```
extract (date_part_expression, datetime_expression)
```

**Example 1**

```
extract (year, 2003-03-03 16:40:15.535)
```

**Result**

```
2003
```

**Example 2**

```
extract (hour, 2003-03-03 16:40:15.535)
```

**Result**

```
16
```

**floor**

Returns the largest integer that is less than or equal to "numeric_expression".

**Syntax**

```
floor (numeric_expression)
```
**int2DTinterval**

Converts an integer to a date time interval. "String_expression" specifies what "integer_expression" represents: "ns" = nanoseconds, "s" = seconds (default), "m" = minutes, "h" = hours, "d" = days.

**Syntax**

\texttt{int2DTinterval ( integer_expression , string_expression )}

**Example 1**

\texttt{int2DTinterval (1020,"h")}

**Result**

42 days 12 hours

**Example 2**

\texttt{int2DTinterval (1020,"s")}

**Result**

17 minutes

**int2YMinterval**

Converts "integer_expression" to a year month interval. "String_expression" specifies what "integer_expression" represents: "y" = years, "m" = months (default).

**Syntax**

\texttt{int2YMinterval ( integer_expression , string_expression )}

**In**

Returns the natural logarithm of "numeric_expression".

**Syntax**

\texttt{ln ( numeric_expression )}

**lower**

Returns "string_expression" with all uppercase characters converted to lowercase.

This function appears in the Bursted Sales Performance Report sample report in the GO Data Warehouse (query) package.

**Syntax**

\texttt{lower ( string_expression )}

**mapNumberToLetter**

Adds "integer_expression" to "string_expression".

**Syntax**

\texttt{mapNumberToLetter ( string_expression , integer_expression )}

**Example**

\texttt{mapNumberToLetter ( 'a' , 1 )}
**Result**

b

**mod**

Returns an integer value representing the remainder (modulo) of "integer_expression1" / "integer_expression2".

**Syntax**

mod ( integer_expression1 , integer_expression2 )

**nullif**

Returns null if "string_expression1" equals "string_expression2" (case-insensitive), otherwise returns "string_expression1".

**Syntax**

nullif ( string_expression1 , string_expression2 )

**number2string**

Converts "numeric_expression" to a string, using the %g format specifier (C/C++ syntax).

**Syntax**

number2string ( numeric_expression )

**octet_length**

Returns the number of bytes in "string_expression".

**Syntax**

octet_length ( string_expression )

**position**

Returns the integer value representing the starting position of "string_expression1" in "string_expression2". Returns 0 if "string_expression1" is not found.

**Syntax**

position ( string_expression1 , string_expression2 )

**power**

Returns "numeric_expression1" raised to the power of "numeric_expression2".

**Syntax**

power ( numeric_expression1 , numeric_expression2 )

**Example**

power ( 3 , 2 )

**Result**

9
round
Returns "numeric_expression" rounded to the nearest value with
"integer_expression" significant digits to the right of the decimal point. If
"integer_expression" is negative, "numeric_expression" is rounded to the nearest
absolute value with "integer_expression" significant digits to the left of the decimal
point. Rounding takes place before data formatting is applied.

Syntax
round ( numeric_expression , integer_expression )

Example
round (125, -1)

Result
130

sqrt
Returns the square root of "numeric_expression". "Numeric_expression" must not
be a negative value.

Syntax
sqrt ( numeric_expression )

status
Returns the status of "expression". Possible values are: 0 - OK, 1 - null, 2 - not
available, 4 - divide by zero, 8 - overflow, 16 - security, 32 - error, 64 - new, 128 -
sample, 256 - pending.

Syntax
status ( expression )

string2date
Returns "string_expression" as a date in YYYY-MM-DD format.

Syntax
string2date ( string_expression )

string2double
Returns a floating point number. "String_expression" has the following form:
"[whitespace] [sign] [digits] [digits] [ (d|D|e|E )[sign]digits]"

Syntax
string2double ( string_expression )

string2DTinterval
Returns "string_expression" as a date time interval in [-]DD HH:MM:[SS][.FFF]]
format.

Syntax
string2DTinterval ( string_expression )
**string2int32**
Returns an integer. "String_expression" has the following form: "[whitespace] [+|-] [digits]"

Syntax
string2int32 ( string_expression )

**string2int64**
Returns a long integer. "String_expression" has the following form: "[whitespace] [+|-] [digits]"

Syntax
string2int64 ( string_expression )

**string2time**
Returns "string_expression" as a time in HH:MM:SS.FFFFFFF format.

Syntax
string2time ( string_expression )

**string2timestamp**
Returns "string_expression" as a timestamp in YYYY-MM-DD [T|t][white space]+ HH:MM:SS.FFFFFFF format.

Syntax
string2timestamp ( string_expression )

**string2timestampTZ**

Syntax
string2timestampTZ ( string_expression )

**string2YMinterval**
Returns "string_expression" as a Year Month Interval in [-]YY MM format.

Syntax
string2YMinterval ( string_expression )

**substring**
Returns the substring of "string_expression" that starts at position "integer_expression1" for "integer_expression2" characters or to the end of "string_expression" if "integer_expression2" is -1. The first character in "string_expression" is at position 1.

Syntax
substring ( string_expression , integer_expression1 , integer_expression2 )

Example
substring ( [Sales (analysis)].[Sales staff].[Sales staff].[Sales staff].[Position code], 3, 5 )

Result

Returns characters 3 to 7 of the position codes.

time2string

Returns a time as a string in HH:MM:SS.FFF format.

Syntax
time2string ( time_expression )

timestamp2date

Converts "timestamp_expression" to a date. The time part of the timestamp will be ignored.

Syntax
timestamp2date ( timestamp_expression )

timestamp2string

Returns a timestamp as a string in YYYY-MM-DD HH:MM:SS.FFFFFFF format.

Syntax
timestamp2string ( timestamp_expression )

timestamp2timestampTZ

Converts "timestamp_expression" to a timestamp with a time zone. The displacement part of the timestamp with the time zone will be zero.

Syntax
timestamp2timestampTZ ( timestamp_expression )

timestampTZ2date

Converts "timestamp_time_zone_expression" to a date. The time and time zone parts of the timestamp will be ignored.

Syntax
timestampTZ2date ( timestamp_time_zone_expression )

timestampTZ2string

Returns a timestamp with the time zone as a string in YYYY-MM-DD HH:MM:SS.FFFFFFF +HHMM or YYYY-MM-DD HH:MM:SS.FFF -HHMM format.

Syntax
timestampTZ2string ( timestamp_time_zone_expression )

timestampTZ2timestamp

Converts "timestamp_time_zone_expression" to a timestamp. The displacement part of the timestamp with the time zone will be ignored.

Syntax
timestampTZ2timestamp ( timestamp_time_zone_expression )

timeTZ2string
- Returns a time with the time zone as a string in HH:MM:SS.FFF +HHMM or HH:MM:SS.FFFFFFF -HHMM format. For example, -05:30 means a timezone of GMT minus 5 hours and 30 minutes

Syntax
- timeTZ2string ( timeTZ_expression )

trim
- Returns "string_expression" trimmed of any leading and trailing blanks or trimmed of the character specified by "match_character_expression". "Trim_what_expression" may be "leading", "trailing", or "both" (default). "Match_character_expression" can be an empty string to trim blanks or can specify a character to be trimmed.

Syntax
- trim ( trim_what_expression , match_character_expression , string_expression )

upper
- Returns "string_expression" with all lowercase characters converted to uppercase.

Syntax
- upper ( string_expression )

YMinterval2string
- Returns "year_month_interval_expression" as a string in (YY MM) or -(YY MM) format.

Syntax
- YMinterval2string ( year_month_interval_expression )
Appendix G. Report Studio Object and Property Reference

This appendix contains definitions of the objects and properties found in IBM Cognos Report Studio. They are available contextually, by pressing F1 when an object or property is active in the Report Studio authoring environment.

Objects

Objects are visible in the Report Studio work area. They can be inserted from the Toolbox tab.

Properties

Properties can be set on Report Studio objects. They are available in the lower-left pane of Report Studio.

Data formatting properties

Data formatting properties can be set on data values by selecting Default Data Format from the Data menu, or by editing the Data Format property for Report Studio objects.

Report Studio Objects

The following is a list of objects available in IBM Cognos Report Studio.

3-D Area

A chart in which members of a data series are represented by three-dimensional areas of varying size and color. The three-dimensional area chart is one of three chart types that can be included in a three-dimensional combination chart.

Properties of 3-D Area

“Border Color” on page 858, “Borders” on page 858, “Chart Type” on page 864, “Values” on page 964

3-D Bar

A chart in which members of a data series are represented by three-dimensional bars of varying length and color. The three-dimensional bar chart is one of three chart types that can be included in a three-dimensional combination chart.

Properties of 3-D Bar

“Border Color” on page 858, “Borders” on page 858, “Chart Type” on page 864, “Values” on page 964
3-D Combination Chart

A chart that includes any number and combination of three-dimensional bar charts, line charts, and area charts. This chart plots any number of data series against one common measure on the vertical numeric axis and one or two common data series on ordinal axes.

Properties of 3-D Combination Chart

- "3-D Viewing Angle" on page 846
- "Alternate Text" on page 848
- "Application Drill-Through" on page 849
- "Background Color" on page 854
- "Background Effects" on page 855
- "Baselines" on page 856
- "Border" on page 857
- "Box Type" on page 859
- "Classes" on page 865
- "Clickable Regions" on page 866
- "Conditional Palette" on page 868
- "Container Select" on page 870
- "Display for Orientation" on page 881
- "Drill-Through Definitions" on page 883
- "Margin" on page 904
- "Master Detail Relationships" on page 907
- "Maximum Hotspots" on page 908
- "Name" on page 914
- "No Data Contents" on page 916
- "Notes" on page 917
- "Numerical Axis" on page 919
- "Padding" on page 921
- "Pagination" on page 921
- "Palette" on page 922
- "Query" on page 928
- "Relative Alignment" on page 930
- "Render Page when Empty" on page 931
- "Render Variable" on page 932
- "Series Color" on page 939
- "Set Variable Values" on page 941
- "Show Hover" on page 942
- "Show Pointer Cursor" on page 943
- "Size & Overflow" on page 945
- "Suppression" on page 951
- "Title" on page 955
- "Tooltips" on page 957
- "Visible" on page 966
- "X Axis" on page 967
- "Y Axis" on page 968

3-D Line

A chart in which members of a data series are represented by three-dimensional lines of varying colors. The three-dimensional line chart is one of three chart types that can be included in a three-dimensional combination chart.

Properties of 3-D Line

- "Border Color" on page 858
- "Borders" on page 858
- "Chart Type" on page 864
- "Values" on page 964

3-D Scatter Chart

A chart that plots three measures against one or more data series on a three-dimensional X-Y-Z graph.

Properties of 3-D Scatter Chart

- "3-D Viewing Angle" on page 846
- "Alternate Text" on page 848
- "Application Drill-Through" on page 849
- "Background Color" on page 854
- "Background Effects" on page 855
- "Baselines" on page 856
- "Border" on page 857
- "Box Type" on page 859
- "Classes" on page 865
- "Clickable Regions" on page 866
- "Conditional Palette" on page 868
- "Container Select" on page 870
- "Display for Orientation" on page 881
- "Drill-Through Definitions" on page 883
- "Legend" on page 900
- "Margin" on page 904
- "Marker Size (pt)" on page 906
- "Master Detail Relationships" on page 907
- "Maximum Hotspots" on page 908
- "Name" on page 914
- "No Data Contents" on page 916
- "Notes" on page 917
- "Padding" on page 921
- "Pagination" on page 921
- "Palette" on page 922
- "Point Shape" on page 924
- "Query" on page 928
- "Relative Alignment" on page 930
- "Render Page when Empty" on page 931
- "Render Variable" on page 932
- "Set Variable Values" on page 941
- "Show Feelers" on page 942
- "Show Hover" on page 942
- "Show Pointer Cursor" on page 943
- "Size & Overflow" on page 945
- "Suppression" on page 951
- "Title" on page 955
- "Tooltips" on page 957
- "Visible" on page 966
Angular Axis

The angular numeric axis for a polar chart, including labels, titles, range, and scale.

Properties of Angular Axis

Angular Measure

The measure that determines the angular position of each data marker on a polar chart.

Properties of Angular Measure

Area

The data marker used to represent data series in an area chart.

Properties of Area

Area

The data marker used to represent data series in an area chart. This object applies only to legacy charts.

Properties of Area

As of Time Expression

An expression that produces a Date-Time value. This expression can be used to show report results for a specific time period that is defined by an expression that you create.
Properties of As of Time Expression

“Report Expression” on page 934

Axis Labels

The labels that appear on an axis.

Properties of Axis Labels


Axis Title

The title for an axis of the chart. This object applies only to legacy charts.

Properties of Axis Title


Axis Title

The title for an axis of the chart.

Properties of Axis Title


Bar

A chart in which members of a data series are represented by bars of varying length and color.

Properties of Bar


Bar

A chart in which members of a data series are represented by bars of varying length and color. This object applies only to legacy charts.
Properties of Bar


Baseline

A baseline to be rendered on a chart.

Properties of Baseline


Baseline

A baseline for a polar chart, scatter chart, or bubble chart.

Properties of Baseline


Baseline

A baseline for a three-dimensional combination chart.

Properties of Baseline


Baseline

A baseline for a three-dimensional scatter chart.

Properties of Baseline

Block
A container into which you can insert other objects.

Properties of Block

Bookmark
A link to another area within the same report. The link can be defined as a static value, a query item, or as the result of a report expression.

Properties of Bookmark
"Data Item Label" on page 875, "Data Item Value" on page 876, "Label" on page 898, "Report Expression" on page 933, "Source Type" on page 948

Bubble Chart
A point chart that plots one or more data series against three measures: a Y-value, an X-value, and a bubble whose relative size represents the third measure. Multiple points are plotted for each category. This object applies only to legacy charts.

To help distinguish values, set the Tooltips property of this object to Yes.

Properties of Bubble Chart
**Bubble Chart**

A point chart that plots one or more data series against three measures: a Y-value, an X-value, and a bubble whose relative size represents the third measure. Multiple points are plotted for each category.

To help distinguish values, set the Tooltips property of this object to Yes.

**Properties of Bubble Chart**

- "Alternate Text" on page 848
- "Application Drill-Through" on page 849
- "Background Effects" on page 855
- "Border" on page 857
- "Box Type" on page 859
- "Bubble Size" on page 860
- "Classes" on page 865
- "Clickable Regions" on page 866
- "Color by Value" on page 866
- "Colored Regions" on page 867
- "Conditional Palette" on page 868
- "Contained Text Direction" on page 870
- "Container Select" on page 870
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "Drill-Through Definitions" on page 883
- "Drop Shadow" on page 884
- "Legend" on page 901
- "Margin" on page 904
- "Markers" on page 906
- "Master Detail Relationships" on page 907
- "Material Effects" on page 907
- "Matrix Rows and Columns" on page 908
- "Maximum Hotspots" on page 908
- "Name" on page 914
- "No Data Contents" on page 916
- "Notes" on page 917
- "Numeric Baselines" on page 919
- "Padding" on page 921
- "Pagination" on page 921
- "Palette" on page 922
- "Plot Area Fill" on page 924
- "Query" on page 928
- "Relative Alignment" on page 930
- "Render Page when Empty" on page 931
- "Render Variable" on page 932
- "Set Variable Values" on page 941
- "Show Hover" on page 942
- "Show Pointer Cursor" on page 943
- "Show Values" on page 945
- "Size & Overflow" on page 945
- "Suppression" on page 951
- "Title" on page 955
- "Tooltips" on page 956
- "Trendlines" on page 958
- "Visible" on page 966

**Bubble Measure**

The measure that determines the size of each bubble on a bubble chart.

**Properties of Bubble Measure**

- "Conditional Styles" on page 868
- "Custom Label" on page 873
- "Data Format" on page 874

**Bubble Measure**

The measure that determines the size of each bubble on a bubble chart. This object applies only to legacy charts.

**Properties of Bubble Measure**

- "Conditional Styles" on page 868
- "Custom Label" on page 873
- "Data Format" on page 874
- "Style Variable" on page 950

**Bullet Chart**

A chart that compares a measure to a target. Use the Colored Regions property to relate the compared measures against additional qualitative measurements, such as colored regions for good, satisfactory, and poor.

**Properties of Bullet Chart**

- "Alternate Text" on page 848
- "Application Drill-Through" on page 849
- "Background Effects" on page 855
- "Bevel" on page 857
- "Border" on page 857
Bullet Measure

The measure that determines the size of the bullet for a bullet chart.

Properties of Bullet Measure

- "Custom Label" on page 873
- "Data Format" on page 874

Button

A button in a static button bar control.

Properties of Button

- "Icon" on page 896
- "Label" on page 898

Button

A single static button control. The button requires a row in the data table of the control.

Properties of Button

- "Application Drill-Through" on page 849
- "Button Definition" on page 861
- "Contained Text Direction" on page 870
- "Control Enable" on page 872
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "Icon" on page 896
- "Label" on page 898
- "Name" on page 914
- "No Data Contents" on page 916
- "Notes" on page 917
- "Padding" on page 921
- "Pagination" on page 921
- "Query" on page 928
- "Relative Alignment" on page 930
- "Render Page when Empty" on page 931
- "Render Variable" on page 932
- "Set Variable Values" on page 941
- "Size" on page 945

Button Bar

A static button bar control. Each button requires a corresponding row in the data table of the control.

Properties of Button Bar

- "Application Drill-Through" on page 849
- "Button Gap" on page 861
- "Button Definition" on page 861
- "Button Width" on page 861
- "Contained Text Direction" on page 870
- "Control Enable" on page 872
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "Name" on page 914
- "Orientation" on page 920
- "Render Variable" on page 932
- "Set Variable Values" on page 941
**Button Extra Item**

Data items that are used to determine what appears as the button labels in a data button bar control.

**Properties of Button Extra Item**

"Data Item" on page 874

**Button Icon**

The icon used to identify each button in a data button bar control. The icons are obtained from a data item inserted into the control.

**Properties of Button Icon**

"Data Item" on page 874

**Button Label**

The label used to identify each button in a data button bar control. Each label is determined by the data items inserted into the control.

**Properties of Button Label**

"Data Item" on page 874

**Calculated Measure**

A data item that is a calculated member.

**Properties of Calculated Measure**


**Calculated Member**

An item, within a dimension, that represents an occurrence of schema data defined as a calculation of two or more members.

**Properties of Calculated Member**

"Caption” on page 862, “Data Item” on page 875, “Name” on page 916

**Calculated Member**

A data item that is a calculated member.

**Properties of Calculated Member**


**Caption**

The caption on a Field Set object.
Properties of Caption

Card
A card in a static deck control.

Properties of Card
“Current Card” on page 873

Card Extra Item
A data item that is used to determine the data that appears in the cards of a data deck control.

Properties of Card Extra Item
“Data Item” on page 874

Card Label
The label used to identify a card in a deck control.

Properties of Card Label
“Data Item” on page 874

Category
A category that is used to plot data in a visualization.

Properties of Category
“Category Range” on page 863, “Label” on page 899

Category Axis
The axis line for an ordinal, or non-numeric, axis. This object applies only to legacy charts.

Properties of Category Axis

Category axis
The axis line for an ordinal, or non-numeric, axis.
Properties of Category axis

<table>
<thead>
<tr>
<th>“Axis Labels” on page 852</th>
<th>“Axis Line” on page 853</th>
<th>“Axis Title” on page 854</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Axis Title” on page 853</td>
<td>“Gridlines” on page 892</td>
<td>“Minor Gridlines” on page 912</td>
</tr>
<tr>
<td>“Number of Minor Intervals” on page 918</td>
<td>“Reverse Category Order” on page 935</td>
<td></td>
</tr>
</tbody>
</table>

Category Baseline
A baseline for the category in a chart.

Properties of Category Baseline

<table>
<thead>
<tr>
<th>“Aggregate Function” on page 847</th>
<th>“Category Baselines” on page 863</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Expression” on page 886</td>
<td>“Hierarchy” on page 894</td>
</tr>
<tr>
<td>“Level” on page 902</td>
<td>“Level Unique Name” on page 902</td>
</tr>
<tr>
<td>“Master Detail Relationships” on page 907</td>
<td>“Member” on page 910</td>
</tr>
<tr>
<td>“Member Offset (%)” on page 910</td>
<td>“Member Unique Name” on page 911</td>
</tr>
<tr>
<td>“Properties” on page 927</td>
<td>“Property Unique Name” on page 928</td>
</tr>
<tr>
<td>“Render Variable” on page 932</td>
<td>“Rollup Aggregate Function” on page 936</td>
</tr>
<tr>
<td>“Root Members Only” on page 936</td>
<td>“Set Definition” on page 940</td>
</tr>
</tbody>
</table>

Chart Body
Defines the body style of the chart itself. The style of the body can be set independently of the chart itself. This object applies only to legacy charts.

Properties of Chart Body

<table>
<thead>
<tr>
<th>“Background Color” on page 854</th>
<th>“Conditional Styles” on page 868</th>
<th>“Fill Effects” on page 887</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Font” on page 888</td>
<td>“Foreground Color” on page 889</td>
<td>“Horizontal Alignment” on page 895</td>
</tr>
<tr>
<td>“Style Variable” on page 950</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chart Body
Defines the body style of the chart itself. The style of the body can be set independently of the chart itself.

Properties of Chart Body

<table>
<thead>
<tr>
<th>“Background Color” on page 854</th>
<th>“Background Effects” on page 855</th>
<th>“Font” on page 888</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Foreground Color” on page 889</td>
<td>“Horizontal Alignment” on page 895</td>
<td></td>
</tr>
</tbody>
</table>

Chart Footer
A footer for the chart. This object applies only to legacy charts.

Properties of Chart Footer

<table>
<thead>
<tr>
<th>“Background Effects” on page 855</th>
<th>“Box Type” on page 860</th>
<th>“Classes” on page 865</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Conditional Styles” on page 868</td>
<td>“Horizontal Alignment” on page 895</td>
<td>“Master Detail Relationships” on page 907</td>
</tr>
<tr>
<td>“Properties” on page 927</td>
<td>“Query” on page 928</td>
<td></td>
</tr>
</tbody>
</table>

Chart Footer
A footer for the chart.
Properties of Chart Footer

“Background Effects” on page 855, “Box Type” on page 860, “Classes” on page 865,
“Conditional Styles” on page 868, “Horizontal Alignment” on page 895, “Master
Detail Relationships” on page 907, “Properties” on page 927, “Query” on page 928,
“Render Variable” on page 932

Chart Measure
The measure for the chart.

Properties of Chart Measure

“Conditional Styles” on page 868, “Custom Label” on page 873, “Data Format” on
page 874

Chart Node Member
A data item, and its accompanying text, to render on the chart.

Properties of Chart Node Member

“Aggregate Function” on page 847, “Custom Label” on page 873, “Data Format”
on page 874, “Expression” on page 886, “Hierarchy” on page 894, “Hierarchy
Unique Name” on page 894, “Level” on page 902, “Level Unique Name” on page
902, “Member” on page 910, “Member Unique Name” on page 911, “Properties”
on page 927, “Property Unique Name” on page 928, “Rollup Aggregate Function”
on page 936, “Root Members Only” on page 936, “Set Definition” on page 940,
“Set Sorting” on page 940, “Sorting” on page 947

Chart Subtitle
The subtitle for a chart. This object applies only to legacy charts.

Properties of Chart Subtitle

“Background Effects” on page 855, “Box Type” on page 860, “Classes” on page 865,
“Conditional Styles” on page 868, “Horizontal Alignment” on page 895, “Master
Detail Relationships” on page 907, “Properties” on page 927, “Query” on page 928,
“Style Variable” on page 950

Chart Subtitle
The subtitle for a chart.

Properties of Chart Subtitle

“Background Effects” on page 855, “Box Type” on page 860, “Classes” on page 865,
“Conditional Styles” on page 868, “Horizontal Alignment” on page 895, “Master
Detail Relationships” on page 907, “Properties” on page 927, “Query” on page 928,
“Render Variable” on page 932

Chart Text Item
The data source and format for a text item, such as a legend item, legend title, axis
label, or axis title. This object applies only to legacy charts.
Properties of Chart Text Item

Chart Text Item

The data source and format for a text item, such as a legend item, legend title, axis label, or axis title.

Properties of Chart Text Item

Chart Title

The title text that appears at the top of the chart. This object applies only to legacy charts.

Properties of Chart Title

Chart Title

The title text that appears at the top of the chart.

Properties of Chart Title

Check Box

A check box in a static check box group control.
Properties of Check Box

“Icon” on page 896, “Label” on page 898

Check Box Extra Item
Data items that are used to determine what appears as the check box labels in a data check box group control.

Properties of Check Box Extra Item

“Data Item” on page 874

Check Box Group
A static check box group control. Each check box requires a corresponding row in the data table of the control.

Properties of Check Box Group


Check Box Icon
The icon used to identify each check box in a data check box group control. The icons are obtained from a data item inserted into the control.

Properties of Check Box Icon

“Data Item” on page 874

Check Box Label
The label used to identify a check box in a data check box group control. The label is determined by the data items inserted into the control.

Properties of Check Box Label

“Data Item” on page 874

Class
The HTML class name for a layout object. Use this attribute to indicate the type of styling to apply to the object when the report is rendered.

Properties of Class

Color by Value Measure
The measure or value used to determine the color of the points in a scatter or bubble chart. The points are colored according to the range of the data item instead of using the palette.

Properties of Color by Value Measure

Colored Region
Defines a collection of colored regions, which are rectangles drawn in the chart body to highlight parts of the chart. The regions are drawn behind the data markers and in the same order that they appear in this element. The first region is drawn first, on the bottom, and the last region is drawn on top of the other regions.

Properties of Colored Region

Colored Region
Defines a collection of colored regions, which are rectangles drawn in the chart body to highlight parts of the chart. The regions are drawn behind the data markers and in the same order that they appear in this element. The first region is drawn first, on the bottom, and the last region is drawn on top of the other regions.

Properties of Colored Region

Combination Chart
A chart that uses combinations of column charts, area charts, and line charts as data markers to plot multiple data series. This object applies only to legacy charts.
Properties of Combination Chart

A chart that uses combinations of column charts, area charts, and line charts as data markers to plot multiple data series.

Properties of Combination Chart

Component Override

Overrides a child object of the Layout Component Reference object.

Properties of Component Override

"Component Reference" on page 868
Conditional Block

A block that can be used for conditional display.

Properties of Conditional Block

- "Background Color" on page 854
- "Background Effects" on page 855
- "Block Variable" on page 857
- "Border" on page 857
- "Box Type" on page 859
- "Classes" on page 865
- "Conditional Styles" on page 868
- "Current Block" on page 873
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "Font" on page 888
- "Foreground Color" on page 889
- "Horizontal Alignment" on page 895
- "Margin" on page 904
- "Padding" on page 921
- "Size & Overflow" on page 945
- "Spacing & Breaking" on page 948
- "Style Variable" on page 950
- "Visible" on page 966
- "WhiteSpace" on page 966

Conditional Block

Contains the default set of layout objects to render based on a report variable.

Properties of Conditional Block

- "Background Color" on page 854
- "Background Effects" on page 855
- "Block Variable" on page 857
- "Border" on page 857
- "Box Type" on page 859
- "Classes" on page 865
- "Conditional Styles" on page 868
- "Current Block" on page 873
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "Font" on page 888
- "Foreground Color" on page 889
- "Horizontal Alignment" on page 895
- "Margin" on page 904
- "Padding" on page 921
- "Size & Overflow" on page 945
- "Spacing & Breaking" on page 948
- "Style Variable" on page 950
- "Visible" on page 966
- "WhiteSpace" on page 966

Context Item Text

The text associated with a context area item.

Properties of Context Item Text

- "Background Color" on page 854
- "Border" on page 857
- "Box Type" on page 859
- "Classes" on page 865
- "Conditional Styles" on page 868
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "Font" on page 888
- "Foreground Color" on page 889
- "Horizontal Alignment" on page 895
- "Label Color" on page 899
- "Maximum Value" on page 910
- "Name" on page 914
- "Orientation" on page 920
- "Range Type" on page 930
- "Render Variable" on page 932
- "Set Variable Values" on page 941
- "Separator" on page 939
- "Size & Overflow" on page 945
- "Spacing & Breaking" on page 948
- "Style Variable" on page 950
- "Visible" on page 966

Continuous Values Slider

A slider control that allows users to slide through numeric values between minimum and maximum values that you define, such as 0 and 100.

Properties of Continuous Values Slider

- "Animate" on page 849
- "Application Drill-Through" on page 849
- "Contained Text Direction" on page 870
- "Control Enable" on page 872
- "Data Format" on page 874
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "Font" on page 888
- "Foreground Color" on page 889
- "Horizontal Alignment" on page 895
- "Label Color" on page 899
- "Maximum Value" on page 910
- "Minimum Value" on page 912
- "Name" on page 914
- "Orientation" on page 920
- "Range Type" on page 930
- "Render Variable" on page 932
- "Set Variable Values" on page 941
- "Step" on page 944
- "Tick Interval" on page 955
- "Tick Label"
Crosstab

A layout object used to render the results of a query that aggregates data, and then arranges it in a two-dimensional grid.

Properties of Crosstab

- Application Drill-Through
- Background Color
- Border
- Box Type
- Classes
- Clickable Regions
- Conditional Styles
- Container Filter
- Container Select
- Default Measure
- Direction & Justification
- Display for Orientation
- Drop Shadow
- Fact Cells Precedence
- Font
- Foreground Color
- Has Fact Cells
- Horizontal Alignment
- Margin
- Master Detail Relationships
- Name
- No Data Contents
- Pagination
- Query
- Render Page when Empty
- Render Variable
- Rows Per Page
- Set Variable Values
- Show Hover
- Show Pointer Cursor
- Size & Overflow
- Summary Text
- Suppression
- Table Properties
- Visible

Crosstab Columns

A list of columns in a crosstab.

Properties of Crosstab Columns

- Background Color
- Background Effects
- Border
- Box Type
- Conditional Styles
- Data Format
- Direction & Justification
- Font
- Foreground Color
- Horizontal Alignment
- Padding
- Size & Overflow
- Spacing & Breaking
- Style Variable
- Vertical Alignment
- White Space
Crosstab Corner
The top-left corner of a crosstab, on top of the row labels and to the left of the column labels. It is generally used to represent crosstab members.

Properties of Crosstab Corner

- Aggregate Function
- Background Color
- Background Effects
- Border
- Box Type
- Classes
- Conditional Styles
- Data Format
- Direction & Justification
- Expression
- Font
- Foreground Color
- Hierarchy
- Level
- Level Unique Name
- Member
- Member Unique Name
- Padding
- Property Unique Name
- Rollup Aggregate Function
- Root Members Only
- Set Definition
- Set Sorting
- Size & Overflow
- Spacing & Breaking
- Vertical Alignment
- White Space

Crosstab Fact Cells
The contents of the fact cells of the crosstab. There is only one fact cell definition for the crosstab, regardless of the number of measures.

Properties of Crosstab Fact Cells

- Aggregate Function
- Apply Single Class
- Background Color
- Background Effects
- Border
- Box Type
- Classes
- Clickable Region
- Conditional Styles
- Data Format
- Direction & Justification
- Expression
- Font
- Foreground Color
- Hierarchy
- Level
- Level Unique Name
- Member
- Member Unique Name
- Property Unique Name
- Rollup Aggregate Function
- Root Members Only
- Set Definition
- Set Sorting
- Size & Overflow
- Spacing & Breaking
- Style Variable
- Vertical Alignment
- White Space

Crosstab Intersection
An intersection in a crosstab. The cell contents of a specific intersection can be overridden and the style defined.

Properties of Crosstab Intersection

- Aggregate Function
- Background Color
- Background Effects
- Border
- Box Type
- Classes
- Clickable Region
- Column Coordinate
- Conditional Styles
- Data Format
- Define Contents
- Direction & Justification
- Drill-Through Definitions
- Expression
- Font
- Foreground Color
- Hierarchy
- Hierarchy Unique Name
- Horizontal Alignment
- Level
- Level Unique Name
- Member
- Member Unique Name
- Padding
- Property Unique Name
- Rollup Aggregate Function
- Root Members Only
- Set Definition
- Set Sorting
- Size & Overflow
- Spacing & Breaking
- Style Variable
- Vertical Alignment
- White Space
Crosstab Member Fact Cells

The contents of the fact cells of a crosstab node member.

Properties of Crosstab Member Fact Cells

Crosstab Node Member

A member in the crosstab node.

Properties of Crosstab Node Member

Crosstab Rows

Overrides the style for Crosstab Row Member objects that is defined in the GlobalReportStyles.css file.
Crosstab Rows
A list of rows in a crosstab.

Properties of Crosstab Rows
- “Background Color” on page 854
- “Background Effects” on page 855
- “Border” on page 857
- “Box Type” on page 860
- “Conditional Styles” on page 868
- “Data Format” on page 874
- “Direction & Justification” on page 880
- “Font” on page 888
- “Foreground Color” on page 889
- “Horizontal Alignment” on page 895
- “Padding” on page 921
- “Size & Overflow” on page 945
- “Spacing & Breaking” on page 948
- “Style Variable” on page 950
- “Vertical Alignment” on page 965
- “White Space” on page 966

Crosstab Space
Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge.

Properties of Crosstab Space
- “Background Color” on page 854
- “Background Effects” on page 855
- “Border” on page 857
- “Box Type” on page 860
- “Conditional Styles” on page 868
- “Data Format” on page 874
- “Direction & Justification” on page 880
- “Font” on page 888
- “Foreground Color” on page 889
- “Horizontal Alignment” on page 895
- “Padding” on page 921
- “Size & Overflow” on page 945
- “Spacing & Breaking” on page 948
- “Style Variable” on page 950
- “Visible” on page 965
- “White Space” on page 966

Crosstab Summary
The crosstab summary that appears in executed active reports.

Properties of Crosstab Summary
- “Background Color” on page 854
- “Background Effects” on page 855
- “Border” on page 857
- “Box Type” on page 860
- “Classes” on page 865
- “Clickable Region” on page 866
- “Conditional Styles” on page 868
- “Data Format” on page 874
- “Direction & Justification” on page 880
- “Font” on page 888
- “Foreground Color” on page 889
- “Horizontal Alignment” on page 895
- “Node Coordinate” on page 917
- “Padding” on page 921
- “Pagination” on page 921
- “Render Fact Cells” on page 931
- “Size & Overflow” on page 945
- “Spacing & Breaking” on page 948
- “Style Variable” on page 950
- “Visible” on page 965
- “White Space” on page 966

Appendix G. Report Studio Object and Property Reference  791
Cumulation Label
A label that is rendered for the cumulation line in a Pareto chart.

**Properties of Cumulation Label**
- "Conditional Styles" on page 868
- "Data Format" on page 874
- "Properties" on page 927
- "Query" on page 928

Cumulation Line
The cumulation line in a Pareto chart.

**Properties of Cumulation Line**
- "Axis Labels" on page 852
- "Axis Line" on page 853
- "Axis Range" on page 853
- "Axis Title" on page 854
- "Cumulation Axis" on page 872
- "Cumulation Label" on page 872
- "Data Points" on page 877
- "Dashes" on page 903
- "Line Styles" on page 903
- "Marker Shape" on page 906
- "Marker Size (pt)" on page 906
- "Number of Minor Intervals" on page 918
- "Properties" on page 927
- "Scale Interval" on page 938
- "Show Values" on page 944
- "Style Variable" on page 950
- "Visible" on page 966
- "Value Location:" on page 963
- "Values" on page 964

Cumulation Line
A line that shows the cumulative effect of multiple series members on a measure in a Pareto chart. This object applies only to legacy charts.

**Properties of Cumulation Line**
- "Cumulation Axis" on page 872
- "Cumulation Label" on page 872
- "Line Styles" on page 903
- "Marker Shape" on page 906
- "Marker Size (pt)" on page 906
- "Properties" on page 927
- "Scale Interval" on page 938
- "Show Values" on page 944
- "Style Variable" on page 950
- "Visible" on page 966
- "Value Location:" on page 963
- "Values" on page 964

Cumulation Line Axis
The axis for the cumulation line in a Pareto chart.

**Properties of Cumulation Line Axis**
- "Axis Labels" on page 852
- "Axis Line" on page 852
- "Axis Title" on page 853
- "Classes" on page 865
- "Conditional Styles" on page 868
- "Data Format" on page 874
- "Font" on page 888
- "Foreground Color" on page 889
- "Gridlines" on page 892
- "Minor Gridlines" on page 912
- "Number of Minor Intervals" on page 918
- "Scale Interval" on page 938
- "Style Variable" on page 950
- "Visible" on page 966

Cumulation Line Label
A label that is rendered with the cumulation line in a Pareto chart. This object applies only to legacy charts.
Properties of Cumulation Line Label


Custom Groups
A group of data items defined by the report author.

Properties of Custom Groups

“Define Custom Grouping” on page 878, “Label” on page 899, “Name” on page 913, “Type” on page 959

Data Button Bar
A data-driven button bar control. Each button label is determined by the data items inserted into the control.

Properties of Data Button Bar


Data Check Box Group
A data-driven check box group control. Each check box label is determined by the data items inserted into the control.

Properties of Data Check Box Group


Data Container
A set of data that is used to populate a visualization.
Properties of Data Container


Data Deck

A data-driven deck control. Each card in the deck is determined by the data items inserted into the control.

Properties of Data Deck


Data Discrete Values Slider

A slider control that allows users to slide through discrete values. The values on the slider are determined by the data items inserted in the control.

Properties of Data Discrete Values Slider


Data Drop-Down List

A data-driven drop-down list control. The items that appear in the list are determined by the data items inserted into the control.

Properties of Data Drop-Down List

Data Item
A set of data values or members.

Properties of Data Item

Data Iterator
A data-driven iterator control that allows users to navigate through values by using buttons such as previous and next. The values on the iterator are determined by the data items inserted in the control.

Properties of Data Iterator

Data List Box
A data-driven list box control. The items that appear in the list are determined by the data items inserted into the control.

Properties of Data List Box

Data Radio Button Group
A data-driven radio button group control. Each radio button label is determined by the data items inserted into the control.

Properties of Data Radio Button Group
Data Tab Control
A data-driven tab control. The label of each tab is determined by the data items inserted into the control.

Properties of Data Tab Control
- "Application Drill-Through" on page 849
- "Contained Text Direction" on page 870
- "Container Select" on page 870
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "Master Detail Relationships" on page 907
- "Name" on page 914
- "Query" on page 928
- "Render Variable" on page 932
- "Set Variable Values" on page 941
- "Size" on page 945
- "Sorting" on page 947
- "Suppression" on page 951
- "Tab Orientation" on page 952
- "Tab Width" on page 953

Data Toggle Button Bar
A data-driven toggle button bar control. Each toggle button label is determined by the data items inserted into the control.

Properties of Data Toggle Button Bar
- "Application Drill-Through" on page 849
- "Button Gap" on page 861
- "Button Width" on page 861
- "Contained Text Direction" on page 870
- "Container Select" on page 870
- "Control Enable" on page 872
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "Master Detail Relationships" on page 907
- "Name" on page 914
- "Orientation" on page 920
- "Query" on page 928
- "Render Variable" on page 932
- "Set Variable Values" on page 941
- "Sorting" on page 947
- "Suppression" on page 951

Date
The date when the report runs.

Properties of Date
- "Background Color" on page 854
- "Classes" on page 865
- "Data Format" on page 874
- "Display for Orientation" on page 881
- "Font" on page 888
- "Foreground Color" on page 889
- "Margin" on page 904
- "Relative Alignment" on page 930

Date & Time Prompt
A prompt control with which you can select a date and time value.

Properties of Date & Time Prompt
- "Box Type" on page 860
- "Calendar Type" on page 862
- "Choices Deselect All Text" on page 864
- "Choices Select All Text" on page 864
- "Choices Text" on page 864
- "Clock Mode" on page 866
- "Conditional Styles" on page 868
- "Containing Text Direction" on page 870
- "Default Selections" on page 878
- "Deselect Text" on page 879
- "Direction & Justification" on page 880
- "Display Milliseconds" on page 882
- "Display Seconds" on page 882
- "Display Time Zone" on page 887
- "First Date" on page 887
- "From Text" on page 887
- "Hide Adornments" on page 893
- "Highest Value Text" on page 894
- "Insert Text" on page 896
- "Last Date" on page 900
- "Lowest Value Text" on page 904
- "Multi-Select" on page 913
- "Name" on page 914
- "Parameter" on page 922
- "Range" on page 929
- "Remove Text" on page 931
- "Render Variable" on page 932
- "Required" on page 934
- "Select UI" on page 939
- "Style Variable" on page 950
- "To Text" on page 958
- "Visible" on page 966
Date Prompt
A prompt control with which you can to select a date value.

Properties of Date Prompt

Deck
A static deck control. Each card in the deck requires a corresponding row in the data table of the control.

Properties of Deck

Default Measure
A default measure for the chart. If the chart measure cannot be determined by the data series rendered on the chart edges, the default measure is used. This object applies only to legacy charts.

Properties of Default Measure

Default Measure
A default measure for the chart. If the chart measure cannot be determined by the data series rendered on the chart edges, the default measure is used.

Properties of Default Measure
“Conditional Styles” on page 868, “Custom Label” on page 873, “Data Format” on page 874

Default Measure
When using a dimensional data source, the default measure for the visualization. If the visualization measure cannot be determined by the data series that is rendered on the visualization edges, the default measure is used.
**Detail Filter**
A set of conditions in a query that narrow the scope of the data returned. A detail filter is applied before aggregation is complete.

**Properties of Detail Filter**

“Application” on page 849, “Definition” on page 879, “Usage” on page 960

**Dimension**
A grouping of descriptive information about an aspect of a business. Dimensions contain levels, whose order defines the hierarchy of organizational structures and data. Dimensions and levels are values by which measures can be viewed, filtered, or aggregated.

**Properties of Dimension**

“Name” on page 915

**Dimensional Edge Summary**
A summary that appears on the edge of a crosstab.

**Properties of Dimensional Edge Summary**


**Discrete Values Slider**
A static slider control that allows users to slide through discrete values. Each value in the slider comes from a row in the data table of the control.

**Properties of Discrete Values Slider**


**Display Layer**
A map layer that is there for appearance only. Display layers do not correspond to data series or measures.

**Properties of Display Layer**

Drop-Down List
A static drop-down list control. Each item in the list requires a corresponding row in the data table of the control.

**Properties of Drop-Down List**

- "Application Drill-Through" on page 849
- "Contained Text Direction" on page 870
- "Container Filter" on page 870
- "Container Select" on page 870
- "Control Enable" on page 872
- "Direction & Justification" on page 880
- "Display for Orientation" on page 881
- "List Items Definition" on page 904
- "Name" on page 914
- "No Value List Item" on page 918
- "Render Variable" on page 932
- "Set Variable Values" on page 941
- "Show Icon" on page 943
- "Size" on page 945

Edge Summary
A summary that appears on the edge of a crosstab.

**Properties of Edge Summary**

- "Aggregate Data Item" on page 847
- "Aggregation Method" on page 847
- "Calculation Intersection" on page 861
- "Label" on page 899
- "Name" on page 913
- "Solve Order" on page 947
- "Type" on page 959
- "Use Set Aggregation" on page 962

Equation Label
An equation label for a trendline that is displayed on a chart.

**Properties of Equation Label**

- "Conditional Styles" on page 868
- "Data Format" on page 874

Equation Label
An equation label for a trendline displayed on a chart.

**Properties of Equation Label**

- "Background Effects" on page 855
- "Conditional Styles" on page 868
- "Font" on page 888
- "Foreground Color" on page 889
- "Horizontal Alignment" on page 895
- "Position" on page 925

Explicit Member Set
A set of data items that define an explicit set of members.

**Properties of Explicit Member Set**

- "Detail" on page 879
- "Dimension" on page 880
- "Hierarchy" on page 894
- "Label" on page 899
- "Members" on page 911
- "Name" on page 913
- "Set Sorting" on page 940
- "Type" on page 950

Extra Category
A data item that is used for filtering or setting variables. The item is not visible in the visualization.
Extra Value
A measure or single value that is used for filtering or setting variables. The item is not visible in the visualization.

Fact
The central values that are aggregated and analyzed. Also known as measures, they are special business measurement values, such as sales or inventory levels.

Properties of Fact
“Data Item” on page 875, “Name” on page 915

Field Set
A container with a caption, into which you can insert other objects. It is similar to a block object, except that it also has a caption.

Properties of Field Set

Gauge Axis
The axis on the face of the gauge that contains gauge indicators, such as data ranges, color ranges, and interval markers.

Properties of Gauge Axis

Gauge Chart
A chart that plots a data series against a measure using a dial or gauge for the measure, and needles or indicators for the series members.

Properties of Gauge Chart
“Alternate Text” on page 848, “Application Drill-Through” on page 849, “Background Effects” on page 855, “Border” on page 857, “Box Type” on page 859
**Gauge Chart**

A chart that plots a data series against a measure using a dial or gauge for the measure, and needles or indicators for the series members. This object applies only to legacy charts.

**Properties of Gauge Chart**

- Alternate Text
- Application Drill-Through
- Axis Title
- Background Color
- Background Effects
- Border
- Border Color
- Borders
- Box Type
- Classes
- Clickable Regions
- Conditional Palette
- Container Select
- Dial Outline Color
- Display for Orientation
- Drill-Through Definitions
- Face Color
- Gauge Labels
- Gauge Palette
- Legend
- Margin
- Master Detail Relationships
- Name
- No Data Contents
- Notes
- Numerical Axis
- Padding
- Pagination
- Query
- Relative Alignment
- Render Page when Empty
- Render Variable
- Set Variable Values
- Show Hover
- Show Pointer Cursor
- Size & Overflow
- Suppression
- Title
- Tooltips
- Visible

**Gauge Labels**

A label for each gauge in a multiple gauge chart.

**Properties of Gauge Labels**

- Classes
- Conditional Styles
- Drill-Through Definitions
- Font
- Foreground Color
- Horizontal Alignment
- Maximum Truncation Characters
- Style Variable
- Truncation
- Truncation Text
- Visible

**Gauge Numerical Axis**

The numeric axis for the gauge chart, including labels, titles, range, and scale.
Properties of Gauge Numerical Axis


Generated Prompt

A control that acts as a placeholder. The report server will replace this control with an appropriate generated prompt control, as if it was on a generated prompt page.

Properties of Generated Prompt

“Hide Adornments” on page 893, “Name” on page 914, “Parameter” on page 922, “Render Variable” on page 932, “Required” on page 934

Hierarchy Set

A set of data items that define the members of a hierarchy.

Properties of Hierarchy Set


HTML Item

A container into which you can add HTML, such as a link to a multimedia file. HTML items will only appear when you run the report in HTML format.

When you upgrade to the next version of IBM® Cognos® Business Intelligence, the report upgrade processes do not account for the use of undocumented and unsupported mechanisms or features such as JavaScript that refers to IBM Cognos HTML items.

Properties of HTML Item


Hyperlink

A hyperlink that can be defined as a static value, a query item, or as the result of a report expression. If a report expression is used, then the other values are ignored.
Properties of Hyperlink

Hyperlink Button
A hyperlink that is formatted as a button. The hyperlink can be defined as a static value, a query item, or as the result of a report expression. If a report expression is used, then the other values are ignored.

Properties of Hyperlink Button

Image
A link to an image file. The link can be a static value, or it can come from a report expression or query item. Use the URL source properties of the image object to define the link.

Properties of Image
Intersection (Tuple)

A data item obtained from the combination of two or more members that you specify. An intersection appears as a single, unified member instead of its component members. Each member must be from a different dimension.

Properties of Intersection (Tuple)

Interval Prompt

An advanced prompt control that allows you to enter time duration values.

Properties of Interval Prompt

Iterator

A static iterator control that allows users to navigate through values by using buttons such as previous and next. Each value in the iterator comes from a row in the data table of the control.

Properties of Iterator

Iterator Extra Item

A data item that is used to determine the data that appears in an iterator control.

Properties of Iterator Extra Item
Iterator First
The button in the iterator that moves the control to the first value.

Properties of Iterator First

Iterator Label Area
The area in the iterator that shows the label of the current value.

Properties of Iterator Label Area

Iterator Last
The button in the iterator that moves the control to the last value.

Properties of Iterator Last

Iterator Next
The button in the iterator that moves the control to the next value.

Properties of Iterator Next

Iterator Previous
The button in the iterator that moves the control to the previous value.

Properties of Iterator Previous

Iterator Value Icon
The icon used to identify each value in an iterator control. The icons are obtained from a data item inserted into the control.

Properties of Iterator Value Icon
“Data Item” on page 874

Iterator Value Label
The label used to identify each value in a data iterator control. Each label is determined by the data items inserted into the control.
Properties of Iterator Value Label

“Data Item” on page 874

Join

A relationship between a field in one table or query and a field of the same data type in another table or query.

Properties of Join

“Join Relationships” on page 897

Key

An object that uniquely identifies members of a level.

If the unique identifier is a primary or alternate key, you need only one key object.
If the unique identifier is a composite key, you need one key object for every data item that participates in making the members of a level unique.

Properties of Key

“Data Item” on page 875, “Name” on page 916

Label

Specifies the text, or label, for the object in a chart.

Properties of Label


Layout Component Reference

A reference to another layout object. Before you can reference an object, its ID property must be set.

Properties of Layout Component Reference


Legend

A key to the patterns or colors assigned to the data series in a chart. This object applies only to legacy charts.

Properties of Legend

Legend

A key to the patterns or colors assigned to the data series in a chart.

Properties of Legend

Legend Title

The title for the legend, including the title text and text style. If this object is empty, a default title is rendered, if available. This object applies only to legacy charts.

Properties of Legend Title

Legend Title

The title for the legend, including the title text and text style. If this object is empty, a default title is rendered, if available.

Properties of Legend Title

Level

A set of members with a predefined set of similar characteristics. For example, the members Year 2011 and Year 2012 in the Time dimension form a year level, while the corresponding quarters form a quarter level.

Properties of Level

Level Hierarchy

Specifies how the levels in a dimension are logically ordered.
**Properties of Level Hierarchy**

“Name” on page 916

**Level Set**

A set of data items that define the members of a level.

**Properties of Level Set**


**Line**

The data marker used to represent a data series in a line chart.

**Properties of Line**


**List**

A layout object that is used to present query results in a list fashion.

**Properties of List**

List Box
A static list box control. Each item in the list requires a corresponding row in the data table of the control.

Properties of List Box

List Cell
A cell in a row, you can use for a list header or footer.

Properties of List Cell

List Column
A column in a list. Generally, the column will contain a query item, but it may also contain any number of layout objects. For example, the To column in a statement list may present the address in a table format. If the column only contains a query items, then the column will automatically span the group if the query item is grouped.

Properties of List Column
List Column Body
The contents of a column in a list report.

Properties of List Column Body

List Columns
A set of columns in a list.

Properties of List Columns

List Columns
Overrides the style for List Column objects that is defined in the GlobalReportStyles.css file.

Properties of List Columns
List Columns Body Style

Overrides the style for List Column Body objects that is defined in the GlobalReportStyles.css file.

Properties of List Columns Body Style

- "BackgroundColor" on page 854
- "Background Effects" on page 855
- "Border" on page 857
- "Box Type" on page 860
- "Conditional Styles" on page 868
- "Data Format" on page 874
- "Direction & Justification" on page 880
- "Font" on page 888
- "Foreground Color" on page 889
- "Horizontal Alignment" on page 895
- "Padding" on page 921
- "Size & Overflow" on page 945
- "Spacing & Breaking" on page 948
- "Style Variable" on page 950
- "Vertical Alignment" on page 965
- "White Space" on page 966

List Columns Title Style

Overrides the style for List Column Title objects that is defined in the GlobalReportStyles.css file.

Properties of List Columns Title Style

- "BackgroundColor" on page 854
- "Background Effects" on page 855
- "Border" on page 857
- "Box Type" on page 860
- "Conditional Styles" on page 868
- "Data Format" on page 874
- "Direction & Justification" on page 880
- "Font" on page 888
- "Foreground Color" on page 889
- "Horizontal Alignment" on page 895
- "Padding" on page 921
- "Size & Overflow" on page 945
- "Spacing & Breaking" on page 948
- "Style Variable" on page 950
- "Vertical Alignment" on page 965
- "White Space" on page 966

List Column Title

The title of a list column.

Properties of List Column Title

- "Aggregate Function" on page 847
- "Allow Sorting" on page 848
- "BackgroundColor" on page 854
- "Background Effects" on page 855
- "Border" on page 857
- "Box Type" on page 860
- "Classes" on page 865
- "Clickable Region" on page 865
- "Column Visibility" on page 867
- "Conditional Styles" on page 868
- "Data Format" on page 874
- "Direction & Justification" on page 880
- "Drill-Through Definitions" on page 883
- "Expression" on page 886
- "Font" on page 888
- "Foreground Color" on page 889
- "Hierarchy" on page 894
- "Hierarchy Unique Name" on page 894
- "Horizontal Alignment" on page 895
- "Level" on page 902
- "Level Unique Name" on page 910
- "Member" on page 910
- "Member Unique Name" on page 911
- "Padding" on page 921
- "Property Unique Name" on page 928
- "Rollup Aggregate Function" on page 936
- "Root Members Only" on page 936
- "Set Definition" on page 940
- "Set Sorting" on page 940
- "Size & Overflow" on page 945
- "Spacing & Breaking" on page 948
- "Style Variable" on page 950
- "Vertical Alignment" on page 965
- "White Space" on page 966

List Footer

Overall footer that appears once at the end of lists. Footers are containers in which you can insert text, images, data items, and report expressions such as the page number.
Properties of List Footer


List Header

Overall header that appears once at the top of lists. Headers are containers in which you can insert text, images, data items, and report expressions such as the current date.

Properties of List Header


List Item

A list item in a static list box control.

Properties of List Item

“Icon” on page 896, “Label” on page 898

List Item Extra Item

Data items that are used to determine what appears as the item labels in a data list box control.

Properties of List Item Extra Item

“Data Item” on page 874

List Item Icon

The icon used to identify each item in a data list box control. The icons are obtained from a data item inserted into the control.

Properties of List Item Icon

“Data Item” on page 874

List Item Label

The label used to identify each item in a data list box control. Each label is determined by the data items inserted into the control.
Properties of List Item Label

“Data Item” on page 874

List Page Footer

The footer that appears at the bottom of every page of a list report. This object is rendered after the list details and other list footers. It is useful for presenting page totals.

Properties of List Page Footer

“Push To Bottom” on page 928

List Page Header

The header in the list that will appear on every page rendered. It occurs after the column titles and before the overall group header of the list. It is useful for presenting carry forward totals.

Properties of List Page Header

“Display After Overall Header” on page 881

List Row

A row in a list.

Properties of List Row


List Row Cells Style

Overrides the style for Row Cells Style objects that is defined in the GlobalReportStyles.css file.

Properties of List Row Cells Style


List Summary

A summary that is automatically added as the footer row in a list.
Properties of List Summary

"Aggregate Data Item" on page 847, "Aggregation Method" on page 847, "Label" on page 899, "Name" on page 913, "Type" on page 959

List Summary

The list summary that appears in executed active reports.

Properties of List Summary


Map

A chart that uses a map to show data.

Properties of Map


Map Location

Associates a data series with regions on the region layer of the map.

Properties of Map Location

"Aggregate Function" on page 847, "Expression" on page 886, "Hierarchy" on page 894, "Hierarchy Unique Name" on page 894, "Level" on page 902, "Level Unique Name" on page 902, "Member" on page 910, "Member Unique Name" on page 911, "Property Unique Name" on page 928, "Rollup Aggregate Function" on page 936, "Root Members Only" on page 936, "Set Definition" on page 940, "Set Sorting" on page 940
Map Location
Associates a data series with points on the point layer of the map.

Properties of Map Location

- Aggregate Function on page 847
- Expression on page 886
- Hierarchy on page 894
- Hierarchy Unique Name on page 894
- Level on page 902
- Level Unique Name on page 902
- Member on page 910
- Member Unique Name on page 911
- Property Unique Name on page 928
- Rollup Aggregate Function on page 936
- Root Members Only on page 936
- Set Definition on page 940
- Set Sorting on page 940

Map Refinement Location
Qualifies the data series members that are associated with regions on the map.

Properties of Map Refinement Location

- Aggregate Function on page 847
- Expression on page 886
- Hierarchy on page 894
- Hierarchy Unique Name on page 894
- Level on page 902
- Level Unique Name on page 902
- Member on page 910
- Member Unique Name on page 911
- Property Unique Name on page 928
- Rollup Aggregate Function on page 936
- Root Members Only on page 936
- Set Definition on page 940
- Set Sorting on page 940

Map Refinement Location
Qualifies the data series members that are associated with points on the map.

Properties of Map Refinement Location

- Aggregate Function on page 847
- Expression on page 886
- Hierarchy on page 894
- Hierarchy Unique Name on page 894
- Level on page 902
- Level Unique Name on page 902
- Member on page 910
- Member Unique Name on page 911
- Property Unique Name on page 928
- Rollup Aggregate Function on page 936
- Root Members Only on page 936
- Set Definition on page 940
- Set Sorting on page 940

Marimekko Chart
A type of bar chart that can show three levels of data.

Properties of Marimekko Chart

- Alternate Text on page 848
- Application Drill-Through on page 849
- Background Color on page 854
- Background Effects on page 855
- Baselines on page 856
- Border on page 857
- Border Color on page 858
- Borders on page 858
- Box Type on page 859
- Category Axis on page 863
- Classes on page 865
- Clickable Regions on page 866
- Conditional Palette on page 868
- Container Select on page 870
- Depth on page 872
- Display for Orientation on page 881
- Drill-Through Definitions on page 883
- Legend on page 900
- Margin on page 904
- Marimekko Totals on page 905
- Markers on page 905
- Marker Text Location on page 906
- Master Detail Relationships on page 907
- Maximum Hotspots on page 908
- Name on page 914
- No Data Contents on page 916
- Notes on page 917
- Padding on page 921
- Pagination on page 921
- Palette on page 922
- Query on page 928
- Relative Alignment on page 930
- Render Page when Empty on page 931
- Render Variable on page 932
Marker

A marker, or symbol, that you can place at a static place in a chart. Markers can designate a point of significance that may help you analyze or understand the data.

Properties of Marker

Marker

A marker, or symbol, that you can place at a static place in a chart. Markers can designate a point of significance that may help you analyze or understand the data. This object applies only to legacy charts.

Properties of Marker

Marker

A marker on a point chart. Point charts include scatter charts, polar charts, radar charts, and bubble charts. This object applies only to legacy charts.

Properties of Marker
**MDX**

A multidimensional expression (MDX) query against an OLAP data source.

**Properties of MDX**

- "Catalog" on page 862, "Data Source" on page 877, "MDX" on page 910, "Name" on page 915

**Measure**

A data item that defines a measure.

**Properties of Measure**

- "Dimension" on page 880, "Label" on page 899, "Member" on page 910, "Member Unique Name" on page 911, "Name" on page 913, "Type" on page 959

**Member**

A data item that defines a member.

**Properties of Member**

- "Detail" on page 879, "Dimension" on page 880, "Hierarchy" on page 894, "Label" on page 899, "Member" on page 910, "Member Unique Name" on page 911, "Name" on page 913, "Type" on page 959

**Member Children Set**

A set of data items that define the children set of a member.

**Properties of Member Children Set**

- "Detail" on page 879, "Dimension" on page 880, "Hierarchy" on page 894, "Label" on page 899, "Member" on page 910, "Member Unique Name" on page 911, "Name" on page 913, "Set Sorting" on page 940, "Type" on page 959

**Member Hierarchy**

The organization of a dimension’s members into a logical tree structure, with each member having one or more ‘parent’ members and an arbitrary number of ‘child’ members.

**Properties of Member Hierarchy**

- "Name" on page 916

**Member Property**

A property that is associated with members of a level. Attributes can be used to refine a search within level members, or to provide additional information about members.

**Properties of Member Property**

- "Data Item" on page 875, "Name" on page 916
Member Property
A member property from either a level or a hierarchy.

Properties of Member Property
- "Dimension" on page 880
- "Hierarchy" on page 894
- "Label" on page 899
- "Level" on page 902
- "Name" on page 913
- "Property" on page 928
- "Property Unique Name" on page 928
- "Type" on page 959

Member Set
A data item that represents a named set. Member sets supply the context for expressions that require evaluation for multiple dimensions. For example, they determine the contexts for which you can apply analytical functions such as rank.

Properties of Member Set
- "Data Item" on page 874
- "Name" on page 915

Metrics Range Chart
A chart that superimposes target value markers, target range markers, and tolerance range markers over any number or combination of bar, line, and area charts.

Properties of Metrics Range Chart
- "Alternate Text" on page 848
- "Application Drill-Through" on page 849
- "Background Color" on page 854
- "Background Effects" on page 855
- "Baselines" on page 856
- "Border" on page 857
- "Box Type" on page 859
- "Category Axis" on page 863
- "Classes" on page 865
- "Clickable Regions" on page 866
- "Conditional Palette" on page 868
- "Container Select" on page 870
- "Depth" on page 879
- "Display for Orientation" on page 881
- "Drill-Through Definitions" on page 883
- "Legend" on page 900
- "Margin" on page 904
- "Marker Color" on page 905
- "Marker Label" on page 905
- "Markers" on page 905
- "Marker Text Location" on page 906
- "Master Detail Relationships" on page 907
- "Maximum Hotspots" on page 908
- "Name" on page 914
- "No Data Contents" on page 916
- "Notes" on page 917
- "Padding" on page 921
- "Pagination" on page 921
- "Palette" on page 922
- "Performance Pattern" on page 924
- "Query" on page 928
- "Range Label" on page 930
- "Relative Alignment" on page 930
- "Render Page when Empty" on page 931
- "Render Variable" on page 932
- "Set Variable Values" on page 941
- "Show Hover" on page 942
- "Show Pointer Cursor" on page 943
- "Size & Overflow" on page 945
- "Suppression" on page 951
- "Target Color" on page 953
- "Target Marker" on page 953
- "Target Marker Border Color" on page 953
- "Target Marker Position" on page 953
- "Target Range (%)" on page 953
- "Title" on page 955
- "Tolerance Color" on page 956
- "Tolerance Label" on page 956
- "Tooltips" on page 957
- "Upper Range Skew (%)" on page 960
- "Value Location:" on page 963
- "Visible" on page 966
- "Visual Angle" on page 966
- "Y1 Axis" on page 968

Metric Studio Diagram
A Metric Studio diagram rendered as a static image.

Properties of Metric Studio Diagram
- "Alternate Text" on page 848
- "Description" on page 879
- "Diagram Identifier" on page 880
Named Set
A reference to a named set from the model.

Properties of Named Set

Node
Data item that represents categorical data in the visualization. For example, the bubbles in a packed bubble visualization.

Properties of Node

Note
A note on a chart. The source of the note text can be static text, a query item, or a report expression.

Properties of Note

Note
A note on a chart. The source of the note text can be static text, a query item, or a report expression. This object applies only to legacy charts.

Properties of Note

Note Content
The content and style of a note.

Properties of Note Content

No Value List Item
The default item in a list control. When the default item is selected when the report is run, it sets the value of the data item in the control to null. By default, the no value item is the first item in the list.
Properties of No Value List Item

“Label” on page 896, “Position in List” on page 925

Numerical Axis

The numeric axis for a progressive chart, including labels, titles, range, scale, and gridlines. This object applies only to legacy charts.

Properties of Numerical Axis


Numerical Axis

The numeric axis for a Pareto chart, including labels, titles, and gridlines. This object applies only to legacy charts.

Properties of Numerical Axis


Numeric Axis

The axis that displays numbers.

Properties of Numeric Axis


Numeric Baseline

The baseline on a numeric axis.

Properties of Numeric Baseline

Page
A page in a layout.

Properties of Page

Page Body
The main body of a page.

Properties of Page Body
“Classes” on page 865, “Size & Overflow” on page 945

Page Break Text
The text associated with a page break.

Properties of Page Break Text

Page Footer
The footer of a page.

Properties of Page Footer
“Classes” on page 865, “Size & Overflow” on page 945

Page Header
The header of a page.

Properties of Page Header
“Classes” on page 865, “Size & Overflow” on page 945

Page Number
The number of a page.
Properties of Page Number


Page Set

The set of pages to render according to a grouping structure.

Properties of Page Set


Pareto Bars

The bars in a Pareto chart.

Properties of Pareto Bars


Pareto Chart

A chart in which data series appear as colored sections stacked in columns or bars. The maximum of each column or bar represents the series total as a percentage of the overall total of all data series in the chart.

Negative values are not supported in Pareto charts.

Properties of Pareto Chart

Pareto Chart
A chart in which data series appear as colored sections stacked in columns or bars. The maximum of each column or bar represents the series total as a percentage of the overall total of all data series in the chart. This object applies only to legacy charts.

Negative values are not supported in Pareto charts.

Properties of Pareto Chart

Percentaged Measure
A measure that shows values as a percentage of a summary.

Properties of Percentaged Measure

Pie Chart
A chart that uses sections of a circle as data markers to plot one or more data series. The size of each section is proportional to the value of each data series for a given category value. Each pie corresponds to a category value. This object applies only to legacy charts.

Properties of Pie Chart
Pie Chart

A chart that uses sections of a circle as data markers to plot one or more data series. The size of each section is proportional to the value of each data series for a given category value. Each pie corresponds to a category value.

Properties of Pie Chart

- Alternate Text
- Application Drill-Through
- Axis Title
- Background Effects
- Bevel
- Border
- Border Color
- Borders
- Box Type
- Classes
- Clickable Regions
- Conditional Palette
- Contained Text Direction
- Container Select
- Depth
- Direction & Justification
- Display for Orientation
- Drill-Through Definitions
- Drop Shadow
- Exploded Slices
- First Slice Angle
- Hole Size (%)
- Legend
- Margin
- Master Detail Relationships
- Material Effects
- Matrix Rows and Columns
- Maximum Hotspots
- Name
- No Data Contents
- Notes
- Padding
- Pagination
- Palette
- Relative Alignment
- Render Page when Empty
- Render Variable
- Set Variable Values
- Show Hover
- Show Pointer Cursor
- Show Values
- Slice Direction
- Summarize Small Slices
- Title
- Tooltips
- Value Representation
- Values
- Visible

Pie Labels

The labels that will be drawn if multiple pie charts are rendered. If this object does not exist, no labels will be rendered.

Properties of Pie Labels

- Classes
- Conditional Styles
- Drill-Through Definitions
- Font
- Foreground Color
- Horizontal Alignment
- Maximum Truncation Characters
- Style Variable
- Truncation
- Truncation Text
- Visible

Point Color Legend Title

The title for the point color legend. If this object is empty, a default title is rendered, if available.
Properties of Point Color Legend Title

Point Layer
A map layer that includes points, such as cities. The color and size of each point is determined by their respective measure.

Properties of Point Layer

Point Measure
The measure that determines the colors of points on a map chart.

Properties of Point Measure

Point Size Legend Title
The title for the point size legend. If this object is empty, a default title is rendered, if available.

Properties of Point Size Legend Title

Point Size Measure
The measure that determines the size of the points on a map chart.

Properties of Point Size Measure

Polar Chart
A point chart that plots one or more data series against two measures. The radius of a data series is determined by one measure and the arc is determined by the other measure. Multiple points are plotted for each category. They can be distinguished with the help of tool tips if the Tool Tip property is set to Yes.
Properties of Polar Chart

Primary Axis
The generic numeric axis for the combination chart. This axis is normally shown as the top-left axis but can be in a different position depending on the chart orientation, as specified in the Chart Orientation property of the chart.

Properties of Primary Axis

Primary Bottom Axis
The numeric axis for a combination chart. This axis is normally rendered as the bottom-left axis, but can change depending on the orientation of the chart, as specified in the Chart Orientation property of the chart.

Properties of Primary Bottom Axis

Progressive Chart
A chart that uses columns as data markers to plot one category across a single measure. The top of the first column represents the starting value for the second column. This chart emphasizes the positive or negative contribution of each value to the total.

Properties of Progressive Chart
Progressive Chart

A chart that uses columns as data markers to plot one category across a single measure. The top of the first column represents the starting value for the second column. This chart emphasizes the positive or negative contribution of each value to the total. This object applies only to legacy charts.

Properties of Progressive Chart

Prompt Button

A predefined button used in prompt pages. Its usage changes according to its Type property, which can be set to Cancel, Back, Next, Finish, or Reprompt.

Properties of Prompt Button
Query

The data that is to be retrieved from the database. The query consists of a source, a selection, detailed and summary filters, and dimension information.

Properties of Query


Query Operation

Union, Intersect, Except (minus) operations on one or more queries that result in a projection list upon which other queries can be based.

Properties of Query Operation


Query Reference

A reference to another query defined in the same query set.

Properties of Query Reference

“Cardinality” on page 862.

Radar Chart

A chart that integrates multiple axes into a single radial figure as lines or stacked areas.

Properties of Radar Chart

Radial Axis

The radial numeric axis for a polar chart or radar chart, including labels, titles, range, and scale.

Properties of Radial Axis


Radial Measure

The measure that determines the distance between the center of the chart and each data marker.

Properties of Radial Measure


Radio Button

A radio button in a static radio button group control.

Properties of Radio Button

“Icon” on page 896, “Label” on page 898

Radio Button Extra Item

Data items that are used to determine what appears as the radio button labels in a data radio button group control.

Properties of Radio Button Extra Item

“Data Item” on page 874

Radio Button Group

A static radio button group control. Each radio button requires a corresponding row in the data table of the control.
Properties of Radio Button Group


Radio Button Icon

The icon used to identify each radio button in a data radio button group control. The icons are obtained from a data item inserted into the control.

Properties of Radio Button Icon

“Data Item” on page 874

Radio Button Label

The label used to identify each radio button in a data radio button group control. Each label is determined by the data items inserted into the control.

Properties of Radio Button Label

“Data Item” on page 874

Region Color Legend Title

The title for the region color legend. If this object is empty, a default title is rendered, if available.

Properties of Region Color Legend Title

“Master Detail Relationships” on page 907, “Properties” on page 927, “Query” on page 928

Region Layer

A map layer that includes regions, such as provinces.

Properties of Region Layer


Region Measure

The measure that determines the colors of regions on a map chart.

Properties of Region Measure

Regression Line
A regression line for a bubble chart or scatter chart.

Properties of Regression Line
“Box Type” on page 860, “Line Styles” on page 903, “Number of Regression Lines” on page 918, “Polynomial Exponent” on page 925, “Properties” on page 927, “Regression Type” on page 930

Regression Line Label
A label that will be shown in the legend for the regression line.

Repeater
A table into which you can insert items that will be repeated.

Properties of Repeater

Repeater Table
Renders query data in a table.

Properties of Repeater Table

Repeater Table Cell
The contents of a repeater table object.

Properties of Repeater Table Cell
Repeater Table Cell
A cell in an active report repeater table.

Properties of Repeater Table Cell

Rich Text Item
Inserts an object that is used to render a subset of HTML in the layout. The HTML may come from either a static or dynamic source, and the object will also render in PDF output. For information about what elements are allowed in rich text items, see Elements Supported in Rich Text Items.

Properties of Rich Text Item

Row Number
Shows the row numbers in a column.

Properties of Row Number

Row Number
Shows the row numbers in a column in an active report. The active report row number object will always reflect the visible row number in the report. The report row number object reflects the server calculated row number.
Properties of Row Number


Scatter Chart
A point chart that plots one or more data series against two measures. Multiple points are plotted for each category. This object applies only to legacy charts.

Properties of Scatter Chart


Scatter Chart
A point chart that plots one or more data series against two measures. Multiple points are plotted for each category.

Properties of Scatter Chart

Scatter Marker

A marker, or symbol, that you can place at a static place in a chart. Markers can designate a point of significance that may help you analyze or understand the data.

Properties of Scatter Marker

Secondary Axis

The generic numeric axis for the combination chart. This axis is normally shown as the top-right axis but can be in a different position depending on the chart orientation, as specified in the Chart Orientation property of the chart.

Properties of Secondary Axis

Secondary Bottom Axis

The numeric axis for a combination chart. This axis is normally shown as the bottom-right axis, but can change depending on the orientation of the chart, as specified in the Chart Orientation property of the chart.

Properties of Secondary Bottom Axis

Select & Search Prompt

An advanced prompt control that allows you to search for values. You cannot use this prompt control with SAP BW data sources.

Properties of Select & Search Prompt
Set Expression
A set of members.

Properties of Set Expression
- "Detail" on page 879
- "Dimension" on page 880
- "Expression" on page 886
- "Hierarchy" on page 894
- "Label" on page 899
- "Name" on page 913
- "Set Sorting" on page 940
- "Type" on page 959

Singleton
A query item that you can insert anywhere in the layout of your report where there is no query associated. When the report is run, the singleton object retrieves only the first row value for that query. Singletons are useful for adding boilerplate text to a report, such as a company name or address, to add overall calculations in a report, or to add multidimensional cell calculations in a report.

Properties of Singleton
- "Name" on page 914
- "Properties" on page 927
- "Query" on page 928
- "Render Variable" on page 932

Slicer Member Set
A set expression that returns members from a single hierarchy of a single dimension.

Properties of Slicer Member Set
- "Expression" on page 885

SQL
An SQL query against a relational data source.

Properties of SQL
- "Data Source" on page 877
- "Name" on page 915
- "SQL" on page 948
- "SQL Syntax" on page 949

Static Repeater Table
An active report layout object that is used to present data in a repeating structure. Repeater table controls allow you to create button bars or grids of custom content that control the content that appears in other controls.
Properties of Static Repeater Table


Summary Filter
A set of conditions in a query that narrow the scope of the data returned. A summary filter is applied after aggregation is complete.

Properties of Summary Filter

“Definition” on page 879, “Scope” on page 938, “Usage” on page 960

Tab
A tab in a static tab control.

Properties of Tab

“Icon” on page 896, “Label” on page 898

Tab Control
A static tab control. Each tab requires a corresponding row in the data table of the control.

Properties of Tab Control


Tab Extra Item
Data items that are used to determine what appears as the tab labels in a data tab control.

Properties of Tab Extra Item

“Data Item” on page 874
**Tab Icon**
The icon used to identify each tab in a data tab control. The icons are obtained from a data item inserted into the control.

**Properties of Tab Icon**

- “Data Item” on page 874

**Tab Label**
The label used to identify each tab in a data tab control. Each label is determined by the data items inserted into the control.

**Properties of Tab Label**

- “Data Item” on page 874

**Table**
A collection of cells in which objects can be organized in a grid fashion.

**Properties of Table**


**Table Cell**
The cells within a row.

**Properties of Table Cell**


**Table of Contents**
A collection of Table of Contents Entry objects that refer the same table of contents. The table of contents is generated in the rendered output.

**Properties of Table of Contents**

Table of Contents Entry
An entry in the table of contents.

Properties of Table of Contents Entry

Table Row
The rows in a table.

Properties of Table Row

Target Measure
The target measure for a bullet chart.

Properties of Target Measure
“Custom Label” on page 873, “Data Format” on page 874

Target Measure
The target measure for a metrics range chart.

Properties of Target Measure

Text Box Prompt
A prompt control that allows you to type in a value.

Properties of Text Box Prompt
Text Item

A text item in a report. The content can be static text, or it can come from a query item or report expression.

Properties of Text Item

- Aggregate Function
- Background Color
- Border
- Box Type
- Classes
- Conditional Styles
- Control Data Item Value
- Data Format
- Data Item Label
- Data Item Value
- Direction & Justification
- Display for Orientation
- Drill-Through Definitions
- Expression
- Font
- Foreground Color
- Hierarchy
- Hierarchy Unique Name
- Label For
- Level
- Level Unique Name
- Margin
- Maximum Characters
- Member
- Member Unique Name
- Name
- Padding
- Property Unique Name
- Relative Alignment
- Render Variable
- Report Expression
- Rollup Aggregate Function
- Root Members Only
- Set Definition
- Set Sorting
- Size & Overflow
- Source Type
- Spacing & Breaking
- Style Variable
- Text
- Text Source Variable
- Use Detail Value on Page
- Visible

Time

Returns the time on the report server when the report server started rendering the report.

Properties of Time

- Background Color
- Classes
- Data Format
- Display for Orientation
- Font
- Foreground Color
- Margin
- Relative Alignment

Time Prompt

An advanced prompt control that allows you to select a time value.

Properties of Time Prompt

- Box Type
- Choices Deselect All Text
- Choices Select All Text
- Choices Text
- Clock Mode
- Contained Text Direction
- Default Selections
- Deselect Text
- Direction & Justification
- Display Milliseconds
- Display Seconds
Toggle Button Bar

A static toggle button bar control. Each toggle button requires a corresponding row in the data table of the control.

Properties of Toggle Button Bar

- Application Drill-Through
- Button Gap
- Buttons Definition
- Button Width
- Contained Text Direction
- Container Select
- Control Enable
- Direction & Justification
- Display for Orientation
- Name
- Orientation
- Render Variable
- Set Variable Values
- To Text
- Visible

Tolerance Measure

The tolerance measure for a metrics chart.

Properties of Tolerance Measure

- Conditional Styles
- Custom Label
- Data Format
- Style Variable

Total Column

A column or bar representing the total cumulative value of all other columns or bars in a progressive chart.

Properties of Total Column

- Aggregate Function
- Data Format
- Data Item Value
- Expression
- Hierarchy
- Hierarchy Unique Name
- Level
- Level Unique Name
- Member
- Member Unique Name
- Properties
- Rollup Aggregate Function
- Root Members Only
- Set Definition
- Set Sorting
- Text
- Total Column Color

Total Column Label

The label to be rendered for the total column.

Properties of Total Column Label

- Properties

Tree Prompt

A data-driven prompt control that shows hierarchical information and allows you to select one or more members.
Properties of Tree Prompt


Trendline

A line or curve that indicates the general direction of the data over time.

Properties of Trendline

“Based on” on page 856, “Line Style” on page 903, “Render Variable” on page 932, “Trendline Label” on page 958, “Type” on page 959

Trendline Label

A label that will be shown in the legend for the trend line.

Properties of Trendline Label


Value

The return value from the evaluation of the report variable.

Value

A group of two or more variable values.

Value

A measure or single value that is used to plot data in a visualization.

Properties of Value


Value Prompt

A prompt control that allows you to select one or more values from a list.

Properties of Value Prompt

Variable
A report variable.

Properties of Variable

Variable Text Item
A text item that renders the value of an active report variable.

Properties of Variable Text Item

Visualization
A visual representation of data.

Properties of Visualization

Win-Loss Chart
A microchart in which the value of each column is either 1 or -1, often denoting a win or a loss.

Properties of Win-Loss Chart
Win-Loss Measure

The measure for a win-loss chart.

Properties of Win-Loss Measure

X Axis

The horizontal numeric axis for the chart, including labels, titles, range, and scale.

Properties of X Axis

X Axis

The x-axis on a scatter or bubble chart.

Properties of X Axis

X Axis

The axis line for an ordinal, or non-numeric, axis.

Properties of X Axis
X Axis Measure
The measure for the horizontal axis of a scatter chart or bubble chart.

Properties of X Axis Measure

Y1 Axis
The numeric axis of a metrics chart, or the primary numeric axis of a combination chart.

Properties of Y1 Axis

Y2 Axis
The secondary numeric axis of a combination chart.

Properties of Y2 Axis

Y Axis
The vertical numeric axis for the chart, including labels, titles, range, and scale.

Properties of Y Axis

Y Axis
The y-axis on a scatter or bubble chart.

Properties of Y Axis
Y Axis
The axis line for an ordinal, or non-numeric, axis.

Properties of Y Axis

- Allow 45° Rotation on page 847
- Allow 90° Rotation on page 848
- Allow Skip on page 848
- Allow Stagger on page 848
- Axis Line on page 852
- Axis Title on page 853
- Axis Title on page 853
- Classes on page 865
- Display Frequency on page 882
- Drill-Through Definitions on page 883
- First Label Index on page 887
- Gridlines on page 892
- Label Control on page 899
- Maximum Truncation Characters on page 909
- Minor Gridlines on page 912
- Truncation on page 958
- Truncation Text on page 959

Y Axis Measure
The measure for the vertical axis of a scatter chart or bubble chart.

Properties of Y Axis Measure

- Conditional Styles on page 868
- Custom Label on page 873
- Data Format on page 874
- Style Variable on page 950

Z Axis
The third numeric axis for a three-dimensional scatter chart, including labels, titles, range, and scale.

Properties of Z Axis

- Axis Labels on page 852
- Axis Line on page 852
- Axis Title on page 853
- Axis Title on page 853
- Classes on page 865
- Data Format on page 874
- Gridlines on page 892
- Include Zero For Auto Scale on page 896
- Maximum Value on page 910
- Minimum Value on page 912
- Minor Gridlines on page 912
- Scale on page 937
- Scale Interval on page 938
- Use Same Range For All Instances on page 962

Z Axis
The vertical numeric axis for a three-dimensional combination chart, including labels, titles, range, and scale.

Properties of Z Axis

- Axis Labels on page 852
- Axis Line on page 852
- Axis Title on page 853
- Axis Title on page 853
- Classes on page 865
- Data Format on page 874
- Gridlines on page 892
- Include Zero For Auto Scale on page 896
- Maximum Value on page 910
- Minimum Value on page 912
- Minor Gridlines on page 912
- Scale on page 937
- Scale Interval on page 938
- Use Same Range For All Instances on page 962

Z Axis Measure
The measure for the third axis of a three-dimensional scatter chart.
Properties of Z Axis Measure


Report Studio Properties

The following is a list of properties available in the lower left pane of IBM® Cognos® Report Studio.

3-D Viewing Angle

Specifies the 3-D viewing angle of the chart.

Applies to

“3-D Combination Chart” on page 772, “3-D Scatter Chart” on page 772

Absolute Position

Specifies that the legend will be positioned absolutely, by setting its pixel position.

Applies to

“Legend” on page 806

Across

Sets the number of times across, or horizontally, that the contents of the object may be rendered.

The default value depends on the setting of the Repeater Direction property. If it is set to Left to right, top to bottom, the default is one. If it is set to Top to bottom, left to right, the default is 20.

Applies to

“Repeater Table” on page 831, “Static Repeater Table” on page 835

Active Dot Color

Specifies the color of the dot of the selected value in the iterator.

Applies to

“Iterator Label Area” on page 805

Active Report Variable

Specifies the variable to use in the control.

Applies to

“Variable Text Item” on page 842

Actual Measure

The measure that is used to show values as a percentage of a summary.
Applies to

“Percentaged Measure” on page 823

Aggregate Data Item
Specifies the data item that is used to calculate the summary or aggregation.

Applies to

“Dimensional Edge Summary” on page 798, “Edge Summary” on page 799, “List Summary” on page 813

Aggregate Function
Specifies the type of aggregation to apply. The Automatic setting means that the application groups or summarizes based on the data type. The Summarize setting means that any setting found in the model will be used to determine the type of aggregation. The default setting is Automatic.

Applies to


Aggregation Method
Specifies the type of summary or aggregation that is applied.

Applies to

“Dimensional Edge Summary” on page 798, “Edge Summary” on page 799

Aggregation Method
Specifies the type of summary or aggregation that is applied.

Applies to

“List Summary” on page 813

Allow 45° Rotation
Specifies whether the labels can be rotated 45 degrees if the labels are long.
Applies to

“Category Axis” on page 780, “X Axis” on page 843, “Y Axis” on page 845

Allow 90° Rotation
Specifies whether the labels can be rotated 90 degrees if the labels are long.

Applies to

“Category Axis” on page 780, “X Axis” on page 843, “Y Axis” on page 845

Allow Skip
Specifies whether some labels can be skipped if they are long.

Applies to

“Category Axis” on page 780, “X Axis” on page 843, “Y Axis” on page 845

Allow Sorting
Specifies whether sorting is allowed for the column when the active report is executed. You can set the value of this property for multiple objects simultaneously by first control+clicking the items in the report.

Applies to

“List Column” on page 809, “List Column Body” on page 810, “List Column Title” on page 811

Allow Stagger
Specifies whether the labels can be staggered if they are long.

Applies to

“Category Axis” on page 780, “X Axis” on page 843, “Y Axis” on page 845

Allow Ties
Specifies whether data values that equal the Win Loss Threshold property are converted to zero and mapped on the zero line.

Applies to

“Win-Loss Chart” on page 842

Alternate Text
Specifies a text alternative for non-text objects such as images and charts. Use to make reports accessible for people who use screen readers.

Applies to

Angular Axis
Specifies whether the axis is rendered.

Applies to

“Polar Chart” on page 825

Animate
Specifies whether to animate the movement of the slider.

Applies to

“Continuous Values Slider” on page 787, “Data Discrete Values Slider” on page 794, “Discrete Values Slider” on page 798

Animation Effect
Specifies the effect that is used to animate data changes in the visualization.

Applies to

“Visualization” on page 842

Application
Specifies if the condition will be applied before or after aggregation and summarization. When true, the condition will apply to the summarized rows, and a fact, or aggregate, in the expression will be interpreted as the aggregated value of the summarized rows. When false, the condition will apply to the detail database rows from the tabular result set prior to aggregation or summarization, and a fact, or aggregate, in the expression will be interpreted as the individual database value before it has been summarized. This property has no effect on OLAP data sources, on references to non-aggregate items, or when automatic summarization is disabled.

Applies to

“Detail Filter” on page 798

Application Drill-Through
Applies to

Apply Single Class
Specifies whether to apply all of the classes defined or only one class. When set to Yes, the last class that was applied is used.

For example, if you applied a class to a crosstab intersection and another class to the member fact cells of a crosstab row, the class applied to the intersection is the last class applied to the fact cells. For information about the order in which classes are applied, see the topic Create and Modify Report and Object Styles.

This property is set to Yes when you are upgrading a report so that the upgraded report will look the same as the original report. This is because objects in reports created using earlier versions of IBM® Cognos® Business Intelligence support only one class.

Applies to
“Crosstab Fact Cells” on page 789

Area Shape
Specifies the shape of an area in an area chart.

Applies to
“Area” on page 773

Auto Font Size
Specifies whether to automatically resize the font to fit the labels.

Applies to
“Axis Labels” on page 774, “Axis Title” on page 774, “Legend” on page 807, “Legend Title” on page 807

Auto Group & Summarize
Specifies whether the application will apply suggested aggregate functions to aggregate data items and group all non-aggregate data items, producing groups and summary rows. If it is set to No, detail rows will be rendered.
Applies to

“Query” on page 828

Auto-Sort
When running the report, specifies whether to automatically sort based on data type.

Applies to

“Query” on page 828

Auto-Submit
Specifies whether the application submits the prompt page automatically, as soon as a value is changed.

Applies to

“Value Prompt” on page 841

Auto Truncation
Specifies whether to allow truncation of text.

Applies to

“Legend” on page 806

Avoid Division by Zero
Specifies whether the application will return a null value when it encounters a division by zero. This property applies only to relational data sources.

Applies to

“Query” on page 828

Avoid Label Collision
Controls how labels are arranged. If set to false, the chart uses the default positions. If set to true, the chart uses a different layout to avoid label collision. To keep existing reports unchanged, set this property to false.

Applies to

“Pie Chart” on page 823

Axis Angles
Specifies the start and end angle and the direction of the gauge axis. All angles are measured in degrees starting from the three o’clock position.

Applies to

“Gauge Axis” on page 800
Axis Assignment
Specifies the numeric axis to use. This property applies only to legacy charts.

Applies to

"Area" on page 773, "Bar" on page 774, "Baseline" on page 775, "Line" on page 808, "Marker" on page 816

Axis Assignment
Specifies which numeric axis to use.

Applies to

"Baseline" on page 775

Axis Justification
Specifies whether the data labels are aligned to the inside or outside of the gauge axis.

Applies to

"Gauge Axis" on page 800

Axis Labels
Specifies whether to show or hide axis labels.

Applies to


Axis Labels
Specifies whether to show or hide axis labels. This property applies only to legacy charts.

Applies to


Axis Line
Specifies the properties of the axis line in a chart. This property applies only to legacy charts.
Applies to

“Angular Axis” on page 773, “Category Axis” on page 780, “Cumulation Line Axis” on page 792, “Numerical Axis” on page 820, “Radial Axis” on page 829, “Win-Loss Chart” on page 842, ”X Axis” on page 843, ”Y1 Axis” on page 844, ”Y2 Axis” on page 844, ”Y Axis” on page 844, ”Y Axis” on page 845, ”Z Axis” on page 845,

Axis Line
Specifies the properties of the axis line in a chart.

Applies to

“Category axis” on page 780, “Cumulation Line” on page 792, “Numeric Axis” on page 820, ”Pareto Bars” on page 822, ”Primary Axis” on page 826, ”Primary Bottom Axis” on page 826, ”Secondary Axis” on page 834, ”Secondary Bottom Axis” on page 834, ”X Axis” on page 843, ”Y Axis” on page 844

Axis Line
Specifies the properties of the gauge axis line in a gauge chart.

Applies to

”Gauge Axis” on page 800

Axis Range
Specifies the appearance of the range of values on an axis.

Applies to

”Cumulation Line” on page 792, ”Gauge Axis” on page 800, ”Numeric Axis” on page 820, ”Pareto Bars” on page 822, ”Primary Axis” on page 826, ”Primary Bottom Axis” on page 826, ”Secondary Axis” on page 834, ”Secondary Bottom Axis” on page 834, ”X Axis” on page 843, ”Y Axis” on page 844

Axis Title
Specifies whether an axis title is rendered.

Applies to

”Angular Axis” on page 773, ”Category Axis” on page 780, ”Category axis” on page 780, ”Cumulation Line” on page 792, ”Numerical Axis” on page 820, ”Numerical Axis” on page 820, ”Pareto Bars” on page 822, ”Primary Axis” on page 826, ”Primary Bottom Axis” on page 826, ”Radial Axis” on page 829, ”Secondary Axis” on page 834, ”Secondary Bottom Axis” on page 834, ”X Axis” on page 843, ”X Axis” on page 843, ”X Axis” on page 843, ”X Axis” on page 843, ”X Axis” on page 843, ”X Axis” on page 843, ”Y1 Axis” on page 844, ”Y2 Axis” on page 844, ”Y Axis” on page 844, ”Y Axis” on page 844, ”Y Axis” on page 844, ”Y Axis” on page 844, ”Z Axis” on page 845, ”Z Axis” on page 845, ”Z Axis” on page 845

Axis Title
Specifies whether an axis title will be rendered. This property applies only to legacy charts.
Applies to


Axis Title

Specifies whether to show the axis titles in a chart.

Applies to


Background Color

Specifies the background color for the object.

Applies to

Background Effects

Specifies the characteristics of a background. You can add background effects only to objects that have a fixed height and width; if a percentage size is given, the effects are ignored.

Applies to


Background Image

Specifies an image to be used as the background for the object.

Applies to

Bar, Line, or Area Index

Specifies which combination object to use when calculating the position. When multiple series exist, the index of 0 represents the topmost bar, line, or area in the Series drop zone, the index of 1 represents the second one, and so on.

Applies to

“Baseline” on page 775, “Baseline” on page 775, “Marker” on page 816

Bar Shape

Specifies the shape of the bars in a bar chart.

Applies to

“Bar” on page 774

Bar Width

Specifies the width of the bars as a percentage of the space available. For example, if you specify 50 percent, the bar takes up half of the space available and the bars are separated from each other. If you specify 100 percent, there is no space between the bars. The default value is 80.

Applies to

“Bar” on page 774, “Pareto Chart” on page 822, “Progressive Chart” on page 826

Based on

Specifies the series data item on which the trendline is based.

Applies to

“Trendline” on page 841

Baselines

Adds reference lines to a chart based on numeric or statistical values, calculations, or layout calculations. This property applies only to legacy charts.

Applies to

Bevel

Specifies whether the chart appears with a beveled edge.

**Applies to**

- “Bar” on page 774
- “Bullet Chart” on page 777
- “Pareto Chart” on page 822
- “Pie Chart” on page 824
- “Progressive Chart” on page 826

Block Variable

Specifies a variable based on which the block can be conditionally rendered.

**Applies to**

- “Conditional Block” on page 787
- “Conditional Block” on page 787

Border

Specifies the width, style, and color for the border of the object.

**Applies to**

- “3-D Combination Chart” on page 772
- “3-D Scatter Chart” on page 772
- “Block” on page 776
- “Bubble Chart” on page 776
- “Bubble Chart” on page 777
- “Bullet Chart” on page 777
- “Caption” on page 779
- “Combination Chart” on page 785
- “Combination Chart” on page 786
- “Conditional Block” on page 787
- “Conditional Block” on page 787
- “Context Item Text” on page 787
- “Crosstab” on page 788
- “Crosstab Columns” on page 788
- “Crosstab Columns” on page 788
- “Crosstab Corner” on page 789
- “Crosstab Fact Cells” on page 789
- “Crosstab Intersection” on page 789
- “Crosstab Member Fact Cells” on page 790
- “Crosstab Node Member” on page 790
- “Crosstab Rows” on page 790
- “Crosstab Rows” on page 791
- “Crosstab Space” on page 791
- “Crosstab Summary” on page 791
- “Field Set” on page 800
- “Gauge Chart” on page 800
- “Gauge Chart” on page 801
- “Hyperlink” on page 802
- “Hyperlink Button” on page 803
- “Image” on page 803
- “List” on page 808
- “List Cell” on page 809
- “List Column” on page 809
- “List Column Body” on page 810
- “List Columns” on page 810
- “List Columns” on page 810
- “List Columns Body Style” on page 811
- “List Columns Title Style” on page 811
- “List Column Title” on page 811
- “List Footer” on page 811
- “List Header” on page 812
- “List Row” on page 813
- “List Row Cells Style” on page 813
- “List Summary” on page 814
- “Map” on page 814
- “Marimekko Chart” on page 815
- “Metrics Range Chart” on page 818
- “Page” on page 821
- “Page Break Text” on page 821
- “Pareto Chart” on page 822
- “Pareto Chart” on page 822
- “Pie Chart” on page 823
- “Pie Chart” on page 823
- “Pie Chart” on page 824
- “Polar Chart” on page 825
- “Progressive Chart” on page 826
- “Progressive Chart” on page 826
- “Prompt Button” on page 827
- “Radar Chart” on page 828
- “Repeater Table” on page 831
- “Repeater Table Cell” on page 831
- “Repeater Table Cell” on page 832
- “Scatter Chart” on page 833
- “Scatter Chart” on page 833
- “Static Repeater Table” on page 835
- “Table” on page 837
- “Table” on page 837
- “Table Cell” on page 837
- “Table of Contents” on page 837
- “Table of Contents Entry” on page 838
- “Text Box Prompt” on page 838
- “Text Item” on page 839
- “Variable Text Item” on page 842
- “Win-Loss Chart” on page 842
Border Color
Specifies the color of the border rendered around specific chart elements, such as bars, stacks, areas, points, or pie slices.

Applies to

Border Color
Specifies the color of the borders around the chart body in a pie chart.

Applies to
- "Pie Chart" on page 824

Borders
Specifies whether borders are rendered around specific chart elements, such as bars, stacks, areas, points, or pie slices.

Applies to

Borders
Specifies whether borders appear around the chart body.

Applies to
- "Area" on page 773, "Bar" on page 774, "Pareto Chart" on page 822, "Progressive Chart" on page 826

Borders
Specifies whether borders appear around the chart body in a pie chart.

Applies to
- "Pie Chart" on page 824
Bottom Position
Specifies the position of the bottom edge of the colored region.

Applies to
“Colored Region” on page 785

Bottom Position (px)
Specifies the pixel position of the bottom edge of the note measured from the bottom of the chart.

Applies to
“Note” on page 819

Bottom Position (px)
Specifies the pixel position of the bottom edge of the legend measured from the bottom of the chart.

Applies to
“Legend” on page 806

Box Type
Each report object has a container, or box, which can be set to display the contents (the default) or not. This property specifies whether to override the default box type for the object.

When set to None, the object is not rendered and its space is not reserved in the report. You could use this property to remove an object completely from a report when a specific condition is met. When set to Inline, you can insert other objects on the same line as the object. When set to Block, you can insert other objects only on the lines above and below the object.

Applies to
Box Type

Each report object has a container, or box, which can be set to display the contents (the default) or not. This property specifies whether to override the default box type for the object. When set to None, the object is not rendered and its space is not reserved in the report.

Applies to


Bubble Size

Specifies how the bubble size is computed. Minimum-Based assigns the smallest bubble to the minimum data value. Zero-Based computes the bubble size relative to 0. This option is compatible with Excel 2002. Zero-Based with Negatives shows negative bubbles as hollow, and the bubbles get larger as they get further from 0. This option is compatible with Excel 2007.

Applies to

"Bubble Chart" on page 776

Bubble Size

Specifies how bubble sizes are calculated on a bubble chart.

Applies to

"Bubble Chart" on page 777

Bullet Indicators

Specifies the size and shape of the bullet chart data marker that represents the actual value.

Applies to

"Bullet Chart" on page 777
Button Definition
Defines the button in the control.

Applies to
"Button" on page 778

Button Gap
Specifies the gap between buttons in the button bar.

Applies to
"Button Bar" on page 778, "Data Button Bar" on page 793, "Data Toggle Button Bar" on page 796, "Toggle Button Bar" on page 840

Buttons Definition
Defines the buttons in the button bar.

Applies to
"Button Bar" on page 778, "Toggle Button Bar" on page 840

Button Width
Specifies the width of buttons in the button bar.

Applies to
"Button Bar" on page 778, "Data Button Bar" on page 793, "Data Toggle Button Bar" on page 796, "Toggle Button Bar" on page 840

Button Width
Specifies the width of the button.

Applies to
"Iterator First" on page 805, "Iterator Last" on page 805, "Iterator Next" on page 805, "Iterator Previous" on page 805

Calculation Intersection
Specifies whether to suppress calculated values that occur at the intersection of a data source calculated member and a query-defined calculated member. When values are suppressed, the characters specified for the Not Applicable characters data format appear in the cells.

Applies to
"Calculated Measure" on page 779, "Calculated Member" on page 779, "Dimensional Edge Summary" on page 798, "Edge Summary" on page 799
Calendar Type
Specifies the type of calendar to show. The date values are mapped to the selected calendar before being formatted. The default value is inherited from the user’s content language.

**Applies to**

"Date & Time Prompt” on page 796, "Date Prompt” on page 797

Caption
Specifies the caption for the level.

**Applies to**

"Level” on page 807

Caption
Specifies the caption.

**Applies to**

“Calculated Member” on page 779

Cardinality
Specifies the cardinality for this join operand.

**Applies to**

"Query Reference” on page 828

Cascade Source
Specifies the parameter whose value is used to filter the values displayed in this control.

**Applies to**

"Select & Search Prompt” on page 834, "Tree Prompt” on page 840, "Value Prompt” on page 841

Case Insensitive
Specifies whether to perform a case insensitive search by default.

**Applies to**

“Select & Search Prompt” on page 834

Catalog
Specifies the OLAP catalog.

**Applies to**

“MDX” on page 817
Category Axis
Specifies whether the axis is rendered.

Applies to

Category Baselines
Adds reference lines on the category axis of a chart based on numeric or statistical values, calculations, or layout calculations.

Applies to
“Category Baseline” on page 781, “Combination Chart” on page 786, “Pareto Chart” on page 822, “Progressive Chart” on page 826

Category Range
Specifies whether to show all categorical data in the visualization or categorical data that is filtered when users select values in an Active Report control. For example, when all data is selected, all legend items are always shown in the visualization, and the color of each legend item stays the same in the visualization. When a category is inserted on an axis, space is reserved on the axis for all data. When filtered data is selected, only the filtered data is shown in the legend and on axes. The color of each legend item can change when users select different values in the control.

Applies to
“Category” on page 780

Chart Orientation
Specifies whether the chart is rendered vertically or horizontally.

Applies to

Chart Type
Specifies whether the data may be rendered as either a bar, line, or area. This property applies only to legacy charts.

Applies to
“Area” on page 773, “Bar” on page 774, “Line” on page 808

Chart Type
Specifies whether the data may be rendered as either a bar, line, or area.
Chart Type
Specifies whether the data may be rendered as either a bar, line or area.

Applies to
“Area” on page 773, “Bar” on page 774, “Line” on page 808

Check Boxes Definition
Defines the check boxes in the check box group.

Applies to
“Check Box Group” on page 784

Choices Deselect All Text
Specifies the text for the link below the choices box that deselects all the items in the box. This property applies to all prompts with either multiple selections and ranges or multiple selections and search. The default link text is Deselect All.

Applies to

Choices Select All Text
Specifies the text for the link below the choices box that selects all the items in the box. This property applies to all prompts with either multiple selections and ranges or multiple selections and search. The default link text is Select All.

Applies to

Choices Text
Specifies the title that appears above the choices box when multiple selections are enabled. This property applies to the following prompt types: value, text box, date, date & time, time, interval, and select & search. The default title text is Choices.

Applies to
Classes

Specifies a class to apply to the object. The class provides a default style. If you apply more than one class, the style properties from all classes are merged together when they are applied. However, if the classes have style properties in common, the style properties from the last class applied override those from previous classes.

Applies to


Clickable Region

Specifies whether to make the object clickable when the active report is executed. You can set the value of this property for multiple objects simultaneously by first control+clicking the items in the report.

Applies to

Clickable Region
Specifies whether to make the object clickable. When (Default) is selected, the value is inherited from the Clickable Regions property of the parent object.

Applies to
“Crosstab Intersection” on page 789, “Crosstab Member Fact Cells” on page 790, “Crosstab Node Member” on page 790, “Crosstab Space” on page 791, “Crosstab Summary” on page 791

Clickable Regions
Specifies which areas of the data container are clickable. For some data containers, if you select to make parts clickable, you can further refine which areas are clickable with the Clickable Region property.

Applies to

Clickable Regions
Specifies which areas of the visualization are clickable. Visualization as a Whole makes the entire visualization the clickable region. Visualization Intersections makes each element in the visualization clickable.

Applies to
“Visualization” on page 842

Clock Mode
Specifies whether the arms of the clock move.

Applies to
“Date & Time Prompt” on page 796, “Time Prompt” on page 839

Color by Value
Specifies how color by value is rendered in a chart.

Applies to
“Bubble Chart” on page 777, “Color by Value Measure” on page 785, “Scatter Chart” on page 833
Colored Regions
Specifies rectangular colored regions drawn in the chart body to highlight parts of the chart. The regions are rendered behind the data markers. The regions are drawn in the same order that they appear in this property. The first region is drawn first, on the bottom, and the last region is drawn on top of the other regions.

Applies to

Color Legend Title
Specifies a title within the legend above the palette for the region color. If this object is not defined, no additional title is drawn. If no legend is drawn, this object is ignored. Styling for this object is inherited from the legend title.

Applies to
“Region Layer” on page 830

Color Legend Title
Specifies a title within the legend above the palette for the point color. If this object is not defined, no additional title is drawn. If no legend is drawn, this object is ignored. Styling for this object is inherited from the legend title.

Applies to
“Point Layer” on page 825

Column Coordinate
Uniquely identifies the column of a node member or spacer on an edge of the crosstab. You cannot modify this value.

Applies to
“Crosstab Intersection” on page 789

Column Titles
Specifies where or whether column titles may be rendered.

Applies to
“List” on page 808

Column Visibility
Specifies whether a column is visible when the active report is executed. You can set column visibility based on a condition.
Applies to
“List Column” on page 809, “List Column Body” on page 810, “List Column Title” on page 811

Combinations
Specifies which axes to show and the chart types to use for the series.

Applies to
“Combination Chart” on page 786

Component Reference
Specifies the layout object that is referenced. An object is a reusable component only if it has a name.

Applies to
“Layout Component Reference” on page 806

Component Reference
Specifies the layout object that is referenced. An object is a reusable component only if it has a name. You cannot modify this value.

Applies to
“Component Override” on page 786

Conditional Palette
Specifies a conditional palette for the chart.

Applies to

Conditional Palette
Specifies a conditional palette for the chart. This property applies only to legacy charts.

Applies to

Conditional Styles
Specifies the conditions and styles used to style the object.
Connecting Lines

Specifies the properties of the lines that connect the segments of a stacked bar. This property is ignored for clustered bars. This property applies only to legacy charts.

Applies to

“Bar” on page 774, “Pareto Chart” on page 823, “Progressive Chart” on page 827

Connecting Lines

Specifies the properties of the lines that connect the segments of a stacked bar in a progressive chart. This property is ignored for clustered bars.
Contains **Text Direction**

Specifies the direction of text contained in compound objects. Compound objects include charts, maps, and prompts. Contextual sets the text direction based on the first alphabetic character in the text. If the character belongs to a language that supports bidirectional script, the direction is right to left. Otherwise, the direction is left to right. Numbers and special characters do not influence the text direction. For example, if the text starts with a number followed by an Arabic character, the direction will be right to left. If the text starts with a number followed by a Latin character, the direction will be left to right.

**Applies to**

“Progressive Chart” on page 826

**Container Filter**

Filters items in the control or container based on the value of a variable.

**Applies to**


**Container Select**

Selects an item in the control or container based on the value of a variable. For example, selects a row in a list, selects a named list item from a drop-down list, or selects a named tab from a tab control.
Contents Height
Specifies the relative height of list rows. This property is used only when a list has a height defined in the Size and Overflow property.

Stretched means that the rows will be evenly sized to fit in the list's height. This is default HTML behavior.

Minimal means that rows will take up only as much space as they need, and be compressed at the top of the list. You can position a footer at the bottom of the list by setting the Push To Bottom property to Yes on a footer object inside the list.

Applies to

“List” on page 808

Control Data Item Value
Specifies the data item used to populate the text item in the control.

Applies to

“Chart Text Item” on page 782, “Chart Text Item” on page 783, “Cumulation Line Label” on page 792, “Hyperlink” on page 802, “Hyperlink Button” on page 803, “Text Item” on page 839

Control Data Item Value
Specifies the data item used to populate the HTML item in the control.

Applies to

“HTML Item” on page 802, “Rich Text Item” on page 832

Control Data Item Value
Specifies the data item used to populate the image in the control.
Applies to

“Hyperlink” on page 802, “Hyperlink Button” on page 803, “Image” on page 803

Control Enable
Enables the control or container based on the value of a variable.

Applies to


Cross Product Allowed
Specifies whether the query will be allowed to run if there is a cross join between database tables. This type of query generates a result set that includes all possible unique combinations of values from the first and second table. The default value is Deny.

Applies to

“Query” on page 828

Cumulation Axis
Specifies whether the axis for the cumulation line is rendered.

Applies to

“Cumulation Line” on page 792

Cumulation Label
Specifies whether a label for the cumulation line is rendered in the legend.

Applies to

“Cumulation Line” on page 792

Cumulation Label
Specifies the label that is shown with the cumulation line on a Pareto chart.

Applies to

“Cumulation Line” on page 792

Cumulative Line
Specifies whether the cumulation line is rendered.
Applies to

“Pareto Chart” on page 823

Cumulative Line
Specifies whether the cumulation line is rendered in a Pareto chart.

Applies to

“Pareto Chart” on page 822

Current Block
Specifies which block is currently being authored.

Applies to

“Conditional Block” on page 787, “Conditional Block” on page 787

Current Card
Specifies which card to show in the report.

Applies to

“Card” on page 780, “Deck” on page 797

Current Tab
Specifies which tab to show in the report.

Applies to

“Tab Control” on page 836

Custom Label
Overrides the default label for the data item.

Applies to


Custom Label
Specifies the data source and format for a text item, such as a legend item, legend title, axis label, or axis title.

Applies to

Data Format

Specifies the data format of the object.

Applies to


Data Format

Specifies the data format of the slider values.

Applies to

“Continuous Values Slider” on page 787, “Data Discrete Values Slider” on page 794, “Discrete Values Slider” on page 798

Data Item

Specifies a reference to a data item. You cannot modify this value.

Applies to

Data Item
Specifies a reference to a data item.

Applies to
- "Calculated Member" on page 779
- "Fact" on page 800
- "Key" on page 806
- "Member Property" on page 817

Data Item Label
Specifies the data item label that defines the text to render.

Applies to
- "Chart Text Item" on page 782
- "Chart Text Item" on page 783
- "Cumulation Line Label" on page 792
- "Hyperlink" on page 802
- "Hyperlink Button" on page 803
- "Text Item" on page 839

Data Item Label
Specifies the data item label that defines the HTML to render.

Applies to
- "HTML Item" on page 802
- "Rich Text Item" on page 832

Data Item Label
Specifies the data item label that defines the URL.

Applies to
- "Hyperlink" on page 802
- "Hyperlink Button" on page 803
- "Image" on page 803

Data Item Label
Specifies the data item label that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to
- "Bookmark" on page 776

Data Item Value
Specifies the data item value that defines the text to render.

Applies to
- "Chart Text Item" on page 782
- "Chart Text Item" on page 783
- "Cumulation Line Label" on page 792
- "Hyperlink" on page 802
- "Hyperlink Button" on page 803
- "Text Item" on page 839
- "Total Column" on page 840
Data Item Value
Specifies the data item value that defines the HTML to render.

Applies to
“HTML Item” on page 802, “Rich Text Item” on page 832

Data Item Value
Specifies the data item value that defines the URL.

Applies to
“Hyperlink” on page 802, “Hyperlink Button” on page 803, “Image” on page 803

Data Item Value
Specifies the data item value that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to
“Bookmark” on page 776

Data Item Value
Specifies the numeric position by using a data item value.

Applies to
“Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Marker” on page 816

Data Item Value
If the marker or baseline is based on a query calculation, allows you to switch to a different data item.

Applies to
“Marker” on page 816, “Numeric Baseline” on page 820

Data Item Value
Specifies the numeric position of a data item.

Applies to
“Marker” on page 816

Data Item Value
Specifies the numeric position of a data item.

Applies to
“Marker” on page 816
Data Language
Specifies the language of the data.

Applies to
“Map” on page 814

Data Points
Specifies whether to show data points on the chart and how they are formatted.

Applies to
“Cumulation Line” on page 792

Data Source
Specifies the query data source.

Applies to
“MDX” on page 817, “SQL” on page 835

Days Text
Specifies the title that appears above the days box in interval prompts. The default title text is Days.

Applies to
“Interval Prompt” on page 804

Deck Cards Definition
The cards in the deck. Each card must have a corresponding row in the static data table.

Applies to
“Deck” on page 797

Default Card
Card to display when no other card matches the current variable state.

Applies to
“Data Deck” on page 794

Default Card
Card to display when no other card matches the current variable state.

Applies to
“Deck” on page 797
Default Measure
Specifies the default measure to use for a crosstab or chart. If the measures of the
crosstab or chart cannot be determined by what is being rendered on the edges,
then the default measure will be rendered.

Applies to
"Crosstab” on page 788

Default Selections
Specifies the collection of default selections for a prompt control.

Applies to
"Date & Time Prompt” on page 796, "Date Prompt” on page 797, "Interval
Prompt” on page 804, "Text Box Prompt” on page 838, "Time Prompt” on page
839, "Tree Prompt” on page 840, "Value Prompt” on page 841

Default Title
Specifies whether the default title is generated.

Applies to
"Axis Title” on page 774, "Legend Title” on page 807

Default Title
Specifies whether the default title may be generated.

Applies to
"Axis Title” on page 774, "Legend Title” on page 807

Define Contents
Overrides the content of the selected crosstab intersection. Use this property to
hide measure values for individual cells or to define custom content.

Applies to
"Crosstab Intersection” on page 789, "Crosstab Member Fact Cells” on page 790

Define Custom Grouping
Specifies the details of the data items in custom groups. Custom groups create
groups of data items that are meaningful to you.

Applies to
"Custom Groups” on page 793

Define Member Sets
Specifies the set structure of a query. If it is not defined, it is assumed that each
data item defines an unrelated set.
Applies to
“Query” on page 828

Definition
Specifies the expression to evaluate when filtering the data.

Applies to
“Detail Filter” on page 798, “Summary Filter” on page 836

Depth
Specifies the three-dimensional depth effect of the chart. A value of zero indicates a flat chart.

Applies to

Description
Specifies a description for the object, that is used to assist authoring.

Applies to
“Class” on page 784, “HTML Item” on page 802, “Metric Studio Diagram” on page 818, “Rich Text Item” on page 832

Deselect Text
Specifies the text for the link that deselects the items when the selection is optional.
This property applies to the following prompt types: text box, date, date & time, time, interval, value, select & search, and tree. The default link text is Deselect.

Applies to

Detail
Specifies whether the data item is to be used for calculating aggregates or not.
When set to Yes, the data item is used to aggregate the lowest level details.

Applies to
Diagram Identifier
Identifies an impact or custom diagram in IBM® Cognos® Metric Studio. Copy the identifier from Metric Studio (Diagrams tab, View the Diagram Identifier button in the Actions column) and paste it in this property. IBM® Cognos® Report Studio decodes the identifier into an image URL. The diagram is imported as a static image.

Applies to
“Metric Studio Diagram” on page 818

Dial Face Fill
Specifies the fill color and effects for the dial face of a gauge chart.

Applies to
“Gauge Chart” on page 800

Dial Outline Color
Specifies the dial outline color in a gauge chart.

Applies to
“Gauge Chart” on page 801

Dictionary
Specifies the aliases to use when matching data values to feature names in the map.

Applies to
“Map” on page 814

Dimension
Specifies a reference to a dimension. You cannot modify this value.

Applies to

Direction & Justification
Specifies text flow properties, such as direction, writing mode, and justification.

Applies to
Display After Overall Header

Specifies whether the list page header is to be rendered after the overall header.

Applies to

“List Page Header” on page 813

Display for Orientation

Specifies the orientation of the mobile device required to display the object when the active report is viewed. For example, if Landscape is selected, the object is displayed only when the mobile device is held in a landscape orientation. The object is not displayed when the mobile device is rotated into a portrait orientation.

Applies to

Display Frequency

Specifies the frequency for which chart labels are to be rendered. If set to 3, for example, every third label will be rendered. This property applies only to legacy charts.

Applies to

“Category Axis” on page 780, “X Axis” on page 843, “Y Axis” on page 845

Display Milliseconds

Specifies whether to show the milliseconds. The format of the milliseconds can be controlled by selecting a specific format. This property is ignored if seconds are not rendered. The default value is inherited from the user’s content language.

Applies to

“Date & Time Prompt” on page 796, “Interval Prompt” on page 804, “Time Prompt” on page 839

Display Seconds

Specifies whether to show the seconds. The format of the seconds can be controlled by selecting a specific format. The default value is inherited from the user’s content language.

Applies to

“Date & Time Prompt” on page 796, “Interval Prompt” on page 804, “Time Prompt” on page 839

Display Value

Specifies the values rendered to the report user when the prompt is used. These values can be different than the ones that are actually used by the report.
Applies to

“Select & Search Prompt” on page 834, “Value Prompt” on page 841

Display Values

Specifies which value of the variable to show. If the variable returns multiple values, List shows the values separated by commas.

Applies to

“Variable Text Item” on page 842

Dot Color

Specifies the color of the dots of the values that are not selected in the iterator.

Applies to

“Iterator Label Area” on page 805

Down

Specifies the number of times down, or rows, that the frame contents may be rendered.

The default value depends on the setting of the Repeater Direction property. If it is set to Left to right, top to bottom, the default is 20. If it is set to Top to bottom, left to right, the default is one.

Applies to

“Repeater Table” on page 831, “Static Repeater Table” on page 835

Drill-Through Definitions

Specifies report-to-report drill-through definitions, bookmarks, and drill-through text for the object.

Applies to

**Drop-Down Width**

Specifies the width of the drop-down list box.

**Applies to**

“Iterator Label Area” on page 805

**Drop Shadow**

Defines a drop shadow that is rendered around a container.

**Applies to**

“Cross tab” on page 788, “List” on page 808, “Repeater Table” on page 831, “Table” on page 837

**Drop Shadow**

Specifies whether a drop shadow appears on a chart and how it appears.

**Applies to**


**Duplicates**

Specifies whether duplicate rows will be preserved or removed.

**Applies to**

“Query Operation” on page 828

**Embed**

Specifies how to embed the reference object. A reference, or pointer, to the object is stored, by default. Alternatively, a copy of the external object can be stored in the report.

**Applies to**

“Layout Component Reference” on page 806

**End Position**

Specifies the position of one edge of the colored region along the numeric axis. The region extends from the position defined in the Start Position to the position defined in this property. The value specified in this property must be greater than the value specified in the Start Position property.

**Applies to**

“Colored Region” on page 785
Execution Method
Specifies whether the query is a candidate to run concurrently. If set to Concurrent, the query may still execute sequentially based on other factors. If not explicitly set, the query will execute sequentially. Concurrent query execution may improve performance in some cases.

Applies to
“Query” on page 828

Execution Optimization
Specifies how much of the query processing is performed by the client and how much is performed by the database server. If the database server can perform all the query processing, it does.

If All Rows is selected, the optimizer adopts a plan that retrieves all rows of the result set in the least amount of time. This value is generally used in a batch environment.

If First Rows is selected, the optimizer adopts a plan that retrieves the first row as quickly as possible. This value is generally used in an interactive environment.

If Incremental is selected, the optimizer retrieves the first N rows, and then retrieves the next N rows.

Applies to
“Query” on page 828

Expand Features
Specifies whether to center and expand the feature in the map chart. When set to Yes, the map feature is centered and expanded to take up all available space in the chart. When set to No, the map feature is not expanded.

Applies to
“Map” on page 814

Exploded Slices
Specifies the slices that appear pulled out of a pie chart and their appearance.

Applies to
“Pie Chart” on page 824

Expression
Specifies the expression that defines the slicer member set.

Applies to
“Slicer Member Set” on page 835
Expression
Specifies the numeric position for a data marker in a scatter chart.

Applies to
“Marker” on page 816

Expression
Specifies the numeric position for a data marker in a scatter chart.

Applies to
“Marker” on page 816

Expression
Specifies the expression used to populate the data item.

Applies to

Extend Width
Specifies whether the area extends to the width of the chart. This property applies only to defined areas of area charts. When the area chart is converted to a matrix of charts, this property is not supported.

Applies to
“Area” on page 773

Face Color
Specifies the color to show on the face of each gauge in a gauge chart.

Applies to
“Gauge Chart” on page 801

Fact Cell Data Format
Specifies the data format of the object.
Applies to

“Node” on page 819

Fact Cells Precedence

Specifies which style property will override the other style property for intersecting cells in a crosstab, the row’s properties, or the column’s properties.

This property only applies to style properties that are both set, but to different values. For example, if the row’s background color is set to yellow and the column’s background color is set to red, you can select which of these properties will override the other. If only the row or the column has a set background color, then that color will be used on the intersecting cell, regardless of this setting.

Applies to

“Crosstab” on page 788

Fill Effects

 Specifies the fill effects for the object.

Applies to


First Column Color

Specifies the color, gradient, or pattern to be used for the first column in the progressive chart.

Applies to

“Progressive Chart” on page 827

First Date

Specifies the earliest date to render in the control, and the earliest date that can be selected.

Applies to

“Date & Time Prompt” on page 796, “Date Prompt” on page 797

First Label Index

Specifies which label will be rendered first. If set to 5, for example, the fifth label will be the first label rendered. Subsequent labels will be rendered as defined in the Display Frequency property. This property applies only to legacy charts.

Applies to

“Category Axis” on page 780, “X Axis” on page 843, “Y Axis” on page 845

First Slice Angle

Specifies the angle at which the first pie slice begins in a pie chart.
Applies to

“Pie Chart” on page 824

Floating

Specifies how objects flow around an object.

Float controls the way the content that follows the selected object will flow around or below it.

Clear controls where the selected object is positioned, relative to other floating objects.

Applies to


Font

Specifies the font family, size, weight, style, and effects used to display the object's text.

Applies to

Font Auto-Sizing

Specifies whether to allow automatic resizing of the font.

 Applies to

“Legend” on page 806

Footer

Specifies whether a chart footer is rendered. This property applies only to legacy charts.

 Applies to

“Win-Loss Chart” on page 842

Foreground Color

Specifies the color of the object’s text.

 Applies to

From Text

Specifies the label that appears beside the beginning of a range. This property applies to the following prompt types: date, date & time, time, and interval. The default label text is From.

Applies to

“Date & Time Prompt” on page 796, “Date Prompt” on page 797, “Interval Prompt” on page 804, “Text Box Prompt” on page 838, “Time Prompt” on page 839, “Value Prompt” on page 841

Gauge Axes

Specifies the start and end angles for the gauge axes.

Applies to

“Gauge Chart” on page 800

Gauge Axis Colors

Specifies the colors of the gauge axis.

Applies to

“Gauge Axis” on page 800

Gauge Axis Inner Radius

Specifies the inner radius of the gauge axis as a percentage of the maximum allowed.

Applies to

“Gauge Axis” on page 800

Gauge Axis Outer Radius

Specifies the outer radius of the gauge axis as a percentage of the maximum allowed.

Applies to

“Gauge Axis” on page 800
**Gauge Border**
Specifies whether borders appear around the chart body in a gauge chart.

**Applies to**
"Gauge Chart" on page 800

**Gauge Labels**
Specifies whether gauge labels are rendered.

**Applies to**
"Gauge Chart" on page 801

**Gauge Needle**
Specifies the size and style of the gauge needle. The chart contains one needle for each row in the data series. All needles on a gauge axis use the same specified styles.

**Applies to**
"Gauge Axis" on page 800

**Gauge Palette**
Specifies the palette that controls the look of the dial portion of a gauge.

**Applies to**
"Gauge Chart" on page 801

**Gauge Pivot**
Specifies whether to show the center pivot point and its color, size, and style.

**Applies to**
"Gauge Chart" on page 800

**Generated SQL**
Shows the generated SQL or MDX.

**Applies to**
"Query" on page 828

**Gradient**
Specifies a color gradient for the page.

**Applies to**
"Page" on page 821
Gridlines

Specifies the properties of the major gridlines in the gauge axis of a gauge chart.

**Applies to**

“Gauge Axis” on page 800

Gridlines

Specifies the properties of the gridlines in a chart.

**Applies to**


Gridlines

Specifies the properties of the major gridlines in a chart.

**Applies to**


Grouping & Sorting

Specifies the grouping and sorting structure.

**Applies to**

“List” on page 808, “Page Set” on page 822, “Repeater” on page 831, “Repeater Table” on page 831

Grouping Type

Specifies whether the absolute, stacked, or 100 percent stacked data may be drawn.

**Applies to**

“Area” on page 773, “Bar” on page 774, “Line” on page 808

Group Span

Specifies the group that this cell should visually span.

**Applies to**

“List Column Body” on page 810
Has Fact Cells
Specifies the contents of the fact cells of the crosstab. There is only one fact cell definition for the crosstab, regardless of the number of measures.

Applies to
“Crosstab” on page 788

Header Text
Specifies the title that appears above the list of choices in a value prompt. The default title text is the name of the level above the data items that are listed as choices; for example, Regions.

Applies to
“Value Prompt” on page 841

Heading Level
Specifies the heading level of the table of contents entry.

Applies to
“Table of Contents Entry” on page 838

Height
Specifies the height of the visualization, in pixels.

Applies to
“Visualization” on page 842

Height (px)
Specifies the height of the note, in pixels.

Applies to
“Note” on page 819

Hide Adornments
Specifies whether to hide the asterisk (*) on required prompts and arrow (->) on type-in prompts that are in an error state.

Applies to
“Date & Time Prompt” on page 796  “Date Prompt” on page 797  “Generated Prompt” on page 798  “Interval Prompt” on page 802  “Select & Search Prompt” on page 834  “Text Box Prompt” on page 838  “Time Prompt” on page 839  “Tree Prompt” on page 840  “Value Prompt” on page 841

Hide Text
Specifies whether to replace characters entered in the prompt control with asterisk (*) characters.
Applies to

"Text Box Prompt" on page 838

Hierarchy

Specifies a reference to a hierarchy. You cannot modify this value.

Applies to


Hierarchy Unique Name

Specifies the Hierarchy Unique Name (HUN) of the hierarchy to which the data item belongs.

Applies to


Highest Value Text

Specifies the label that appears beside the highest value option when ranges are enabled. This property applies to the following prompt types: date, date & time, time, value, text box, and interval. The default label text is Latest date, Latest time, or Highest interval.
Applies to

“Date & Time Prompt” on page 796, “Date Prompt” on page 797, “Interval Prompt” on page 804, “Text Box Prompt” on page 838, “Time Prompt” on page 839, “Value Prompt” on page 841

Hole Size (%)
Specifies the size of the hole in a donut chart. A value of zero indicates a pie chart.

Applies to

“Pie Chart” on page 823, “Pie Chart” on page 824

Horizontal Alignment
Specifies how the contents of the selected object are aligned horizontally.

Applies to


Hours Text
Specifies the title that appears above the hours box in interval prompts. The default title text is Hrs.

Applies to

“Interval Prompt” on page 804

HTML
Specifies the static text used as HTML.

Applies to

“HTML Item” on page 802, “Rich Text Item” on page 832
**HTML Source Variable**

Specifies a variable based on which the HTML source is chosen.

**Applies to**

- “HTML Item” on page 802, “Rich Text Item” on page 832

**Icon**

Specifies the image used to identify the object in the control.

**Applies to**

- “Button” on page 788, “Check Box” on page 783, “List Item” on page 812, “Radio Button” on page 829, “Tab” on page 836

**Ignore Data with No Features**

Specifies whether to allow data that has no corresponding features. When set to Yes, data with no corresponding features will be ignored. When set to No, the map will not run if it contains data with no corresponding features.

**Applies to**

- “Map” on page 814

**Include Zero For Auto Scale**

Specifies whether the value zero is included in the automatic calculation of the numeric scale. This property applies only to legacy charts.

**Applies to**


**Insert Text**

Specifies the label that appears on the button that is used to add items to the selected items box in all multiple selection prompts. The default label text is Insert.

**Applies to**


**Iterator First**

Specifies whether to show or hide the First button in the iterator.

**Applies to**

- “Data Iterator” on page 795, “Iterator” on page 804
Iterator Label Area
Specifies whether to show or hide the labels of iterator values.

Applies to
“Data Iterator” on page 795, “Iterator” on page 804

Iterator Last
Specifies whether to show or hide the Last button in the iterator.

Applies to
“Data Iterator” on page 795, “Iterator” on page 804

Iterator Next
Specifies whether to show or hide the Next button in the iterator.

Applies to
“Data Iterator” on page 795, “Iterator” on page 804

Iterator Previous
Specifies whether to show or hide the Previous button in the iterator.

Applies to
“Data Iterator” on page 795, “Iterator” on page 804

Iterator Values Definition
Defines the values of the iterator.

Applies to
“Iterator” on page 804

Join Relationships
Specifies how to join the two queries.

Applies to
“Join” on page 806

Keywords Text
Specifies the title that appears above the keyword search box in select & search prompts. The default title text is Keywords.

Applies to
“Select & Search Prompt” on page 834
Label
Specifies the static text that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to
“Bookmark” on page 776

Label
Specifies the class label for a local class.

Applies to
“Class” on page 784

Label
Specifies the class label for a global class. You cannot modify this label.

Applies to
“Class” on page 784

Label
Specifies the name of the no value list item.

Applies to
“No Value List Item” on page 819

Label
Specifies the name of the object in the control.

Applies to
“Button” on page 778, “Button” on page 778, “Check Box” on page 783, “List Item” on page 812, “Radio Button” on page 829, “Tab” on page 836

Label
Specifies whether a label is rendered for the baseline.

Applies to
“Baseline” on page 775, “Baseline” on page 775

Label
Specifies whether a label is rendered for the marker.

Applies to
“Marker” on page 816, “Marker” on page 816
Label
The label for the data item. Automatic text uses the data item name as the label. Specified text uses text that you specify as the label. With specified text, you can add translations for the label to support users in multiple languages.

Applies to
“Category” on page 780, “Value” on page 841

Label
Defines the button label in the iterator.

Applies to
“Iterator First” on page 805, “Iterator Last” on page 805, “Iterator Next” on page 805, “Iterator Previous” on page 805

Label
Specifies the label of the object.

Applies to

Label Color
Specifies the color of labels in the slider.

Applies to
“Continuous Values Slider” on page 787, “Data Discrete Values Slider” on page 794, “Discrete Values Slider” on page 798

Label Control
Controls how the labels in a chart are rendered.

Applies to
“Category Axis” on page 780, “X Axis” on page 843, “Y Axis” on page 845

Label For
Specifies the name of the prompt control to associate to the text item. This property is used to make a prompt control accessible by defining a label for the control.

Applies to
“Text Item” on page 839
**Label Location**
Specifies whether to show the label in the legend or in the chart.

**Applies to**
“Marker” on page 816, “Scatter Marker” on page 834

**Labels**
Specifies whether labels are rendered.

**Applies to**
“Pie Chart” on page 823

**Last Date**
Specifies the latest date rendered in the control, and the last date that can be selected.

**Applies to**
“Date & Time Prompt” on page 796, “Date Prompt” on page 797

**Left Position**
Specifies the position of the left edge of the colored region.

**Applies to**
“Colored Region” on page 785

**Left Position (px)**
Specifies the pixel position of the bottom edge of the note measured from the left edge of the chart.

**Applies to**
“Note” on page 819

**Left Position (px)**
Specifies the pixel position of the left edge of the legend measured from the left edge of the chart.

**Applies to**
“Legend” on page 806

**Legend**
Specifies whether the legend is rendered.

**Applies to**
Legend
Specifies whether a legend is rendered and where the legend is positioned in a chart.

Applies to

Legend Label
Specifies whether to render the baseline in the legend.

Applies to
“Baseline” on page 775, “Baseline” on page 775

Legend Separator
Specifies the separator to use between the legend entry and the value when you show values in the legend. The default is a comma (,) followed by a space.

Applies to
“Legend” on page 807

Legend Title
Specifies whether a legend title is rendered. This property applies only to legacy charts.

Applies to
“Legend” on page 806

Legend Title
Specifies whether a legend title is rendered in a chart.

Applies to
“Legend” on page 807

Legend Title
Specifies whether a legend title is rendered in a chart.

Applies to
“Legend” on page 806, “Legend” on page 807
Level
Specifies a reference to a level. You cannot modify this value.

Applies to

Level Indentation
Controls the indentation of the contents of node members in a crosstab.

Applies to
"Crosstab Node Member" on page 790

Level Unique Name
Specifies the Level Unique Name (LUN) of the level to which the data item belongs.

Applies to

Line and Markers
Specifies whether to display a line and whether to display markers.

Applies to
"Line" on page 808
Line Shape
Specifies the shape of a line in a line chart.

Applies to
"Line" on page 808

Line Style
Specifies the style, color, and weight of the line in a Pareto chart.

Applies to
"Cumulation Line" on page 792

Line Style
Specifies the style, color, and weight of the line.

Applies to
"Category Baseline" on page 781, "Numeric Baseline" on page 820, "Trendline" on page 841

Line Style
Specifies the style, color, and weight of the line. This property applies only to legacy charts.

Applies to
"Line" on page 808

Line Styles
Specifies the style, color, and weight of the line.

Applies to
"Baseline" on page 775, "Baseline" on page 775, "Baseline" on page 775, "Baseline" on page 775, "Cumulation Line" on page 792, "Regression Line" on page 831

Line Type
Specifies the type of line used to connect data values.

Applies to
"Line" on page 808

Line Weight (pt)
Specifies the line thickness in points. A value of zero indicates the thinnest possible line.

Applies to
"Line" on page 808
List Items Definition
Defines the list items in the control.

Applies to
"Drop-Down List" on page 799, "List Box" on page 809

Loss Color
Specifies a color, color gradient, or pattern to apply to the loss values.

Applies to
"Win-Loss Chart" on page 842

Lowest Value Text
Specifies the label that appears beside the lowest value option when ranges are enabled. This property applies to the following prompt types: date, date & time, time, value, text box, and interval. The default label text is Earliest date, Earliest time, or Lowest interval.

Applies to
"Date & Time Prompt" on page 796, "Date Prompt" on page 797, "Interval Prompt" on page 804, "Text Box Prompt" on page 838, "Time Prompt" on page 839, "Value Prompt" on page 841

Map & Layers
Sets the map and layers for a map chart.

Applies to
"Map" on page 814

Map Drills
Controls the drill definitions in a map. It is possible to define different drill targets for each region or point.

Applies to
"Point Layer" on page 825, "Region Layer" on page 830

Margin
Specifies the margin properties for the object.

Applies to
"3-D Combination Chart" on page 772, "3-D Scatter Chart" on page 772, "Block" on page 776, "Bubble Chart" on page 776, "Bullet Chart" on page 777, "Caption" on page 779, "Combination Chart" on page 785, "Caption" on page 786, "Conditional Block" on page 787, "Conditional Block" on page 787, "Context Item Text" on page 787, "Crosstab" on page 788, "Date" on page 796, "Field Set" on page 800, "Gauge Chart" on page 800, "Gauge Chart" on page 801, "Hyperlink" on page 802, "Hyperlink Button" on page 803,
Marimekko Totals
Specifies whether to show totals for each column at the top of the chart.

Applies to
"Marimekko Chart" on page 815

Marker Color
Specifies a color for the target value markers in a metrics chart.

Applies to
"Metrics Range Chart" on page 818

Marker Color
Specifies the color of the markers.

Applies to
"Marker" on page 816

Marker Label
Specifies whether the label for the Target Marker will appear in the legend.

Applies to
"Metrics Range Chart" on page 818

Markers
Adds reference points to a chart based on numeric or statistical values, calculations, or layout calculations. This property applies only to legacy charts.

Applies to
Markers
Adds reference points to a chart based on numeric or statistical values, calculations, or layout calculations.

Applies to
“Bubble Chart” on page 777, “Combination Chart” on page 786, “Pareto Chart” on page 822, “Progressive Chart” on page 826, “Scatter Chart” on page 833

Marker Shape
Specifies the shape of the markers.

Applies to
“Cumulation Line” on page 792, “Marker” on page 816, “Marker” on page 816

Marker Size (pt)
Specifies the size of markers in points. A value of zero means do not show markers.

Applies to

Marker Text Location
Specifies where the text of the marker is rendered.
## Master Detail Relationships

Specifies relationships between the master data container and the detail data container. Specifically, specifies how query items in the master query are linked to query items or parameters in the detail query.

### Applies to

- “3-D Combination Chart” on page 772
- “3-D Scatter Chart” on page 772
- “Axis Title” on page 774
- “Bubble Chart” on page 777
- “Bullet Chart” on page 777
- “Category Baseline” on page 781
- “Chart Footer” on page 781
- “Chart Subtitle” on page 782
- “Chart Subtitle” on page 782
- “Chart Title” on page 783
- “Chart Title” on page 783
- “Colored Region” on page 785
- “Colored Region” on page 785
- “Combination Chart” on page 785
- “Combination Chart” on page 785
- “Crosstab” on page 788
- “Data Button Bar” on page 793
- “Data Check Box Group” on page 793
- “Data Container” on page 793
- “Data Deck” on page 794
- “Data Discrete Values Slider” on page 794
- “Data Drop-Down List” on page 794
- “Data Iterator” on page 795
- “Data List Box” on page 795
- “Data Radio Button Group” on page 795
- “Data Tab Control” on page 796
- “Data Toggle Button Bar” on page 796
- “Gauge Chart” on page 800
- “Gauge Chart” on page 801
- “Label” on page 806
- “Legend Title” on page 807
- “Legend Title” on page 807
- “List” on page 808
- “Map” on page 814
- “Marimekko Chart” on page 815
- “Marker” on page 816
- “Markers” on page 816
- “Metrics Range Chart” on page 818
- “Note” on page 819
- “Note Content” on page 819
- “Numeric Baseline” on page 820
- “Page” on page 821
- “Page Set” on page 822
- “Pareto Chart” on page 822
- “Pareto Chart” on page 822
- “Pie Chart” on page 823
- “Pie Chart” on page 824
- “Point Color Legend Title” on page 824
- “Point Size Legend Title” on page 825
- “Polar Chart” on page 825
- “Progressive Chart” on page 826
- “Progressive Chart” on page 827
- “Radar Chart” on page 828
- “Region Color Legend Title” on page 830
- “Repeater” on page 831
- “Repeater Table” on page 831
- “Scatter Chart” on page 833
- “Scatter Chart” on page 833
- “Scatter Marker” on page 834
- “Win-Loss Chart” on page 842

### Material Effects

Specifies a material effect, such as plastic or metallic.

### Applies to

- “Bubble Chart” on page 777
- “Combination Chart” on page 786
- “Pareto Chart” on page 822
- “Progressive Chart” on page 826
- “Scatter Chart” on page 833

### Material Effects

Specifies a material effect, such as plastic or metallic, in a pie chart.
Applies to

“Pie Chart” on page 824

Matrix Rows and Columns
Specifies whether to render a matrix of charts in rows and columns.

Applies to


Maximum Characters
Specifies the maximum number of characters to show before the text is truncated.

Applies to

“Legend” on page 806, “Text Item” on page 839

Maximum Execution Time
Specifies the maximum period, in seconds, that the query can spend to open the database cursor and to retrieve the first row of data. An error is returned if the specified time is exceeded. Note that this property is not for the total time required to execute the query. If no value is specified, no error is returned and the query runs until complete.

Applies to

“Query” on page 828

Maximum Hotspots
Specifies the maximum number of hotspots generated in a chart. A hotspot in a chart appears when you pause a pointer over it. For example, a hotspot on a drill-down symbol or a tooltip gives details about the column, line, or pie slice. If specified, this value overrides the hotspot configuration settings in IBM® Cognos® Administration.

For more information, see the topic Changing the number of hotspots in a chart.

Applies to

Maximum Rows Retrieved

Specifies the maximum number of database rows that the query can retrieve. An error is returned if the number of database rows returned exceeds the specified value. If no value is specified, no error is returned and the query returns all rows.

Applies to

“Query” on page 828

Maximum Size (pt)

Specifies the maximum size used for map point features that have corresponding data. For example, if the minimum size is 2pt and the maximum size is 12pt, the size of each point is calculated using linear interpolation that is based on its measure value.

Applies to

“Point Size Measure” on page 825

Maximum Tables

Specifies the maximum number of tables that the query can retrieve. An error is returned if the number of tables in the generated IBM® Cognos® SQL exceeds the specified value. If no value is specified, no error is returned and there is no restriction on the number of tables that can be queried.

Applies to

“Query” on page 828

Maximum Text Blob Characters

Specifies the maximum number of characters that the query is allowed to retrieve for each text BLOB. An error is returned if the number of characters retrieved exceeds the specified value. If no value is specified, no error is returned and text BLOBs can be of any size.

Applies to

“Query” on page 828

Maximum Truncation Characters

Specifies the maximum number of characters to show before the label is truncated.

If the Allow Truncation property is set to Yes and no value is specified in the Maximum Characters property, the application automatically determines the optimum number of characters after which to truncate. Use this property only if you want explicit control over the truncation level. Note that regardless of this property’s setting, no truncation will occur if there is sufficient space.

Applies to

Maximum Value

Specifies the maximum value for the numeric scale. If no value is specified, one will be calculated based on the data.

Applies to


Maximum Value

Specifies the maximum value for the slider.

Applies to

“Continuous Values Slider” on page 787

MDX

The text of the typed-in MDX. It is assumed to be appropriate for the type and data source. If it is not the query may fail or produce unexpected results.

Applies to

“MDX” on page 817

Member

Specifies a reference to a member. You cannot modify this value.

Applies to


Member Offset (%)

Specifies a position relative to the next item in the chart. This is a percentage value. Negative values indicate before the member and positive values indicate after the member.
**Apply to**

"Baseline" on page 775, "Category Baseline" on page 781, "Marker" on page 816

**Members**

Specifies the members of the intersection (tuple).

**Apply to**

"Intersection (Tuple)" on page 804

**Members**

Specifies the members in the member set.

**Apply to**

"Explicit Member Set" on page 799

**Member Unique Name**

Specifies the Member Unique Name (MUN) of the member.

**Apply to**


**Milliseconds Text**

Specifies the title that appears above the milliseconds box in interval prompts. The default title text is ms.

**Apply to**

"Interval Prompt" on page 804

**Minimum Size (pt)**

Specifies the minimum size used for map point features that have corresponding data. For example, if the minimum size is 2pt and the maximum size is 12pt, the size of each point is calculated using linear interpolation that is based on its measure value.
Applies to

“Point Size Measure” on page 825

Minimum Value

Specifies the minimum value for the numeric scale. If no value is specified, one will be calculated based on the data.

Applies to


Minimum Value

Specifies the minimum value for the slider.

Applies to

“Continuous Values Slider” on page 787

Minor Gridlines

Specifies the properties of the minor gridlines in the gauge axis of a gauge chart.

Applies to

“Gauge Axis” on page 800

Minor Gridlines

Specifies the properties of the minor gridlines in a chart. This property applies only to legacy charts.

Applies to


Minor Gridlines

Specifies the properties of the minor gridlines in a chart.

Applies to

Minutes Text
Specifies the title that appears above the minutes box in interval prompts. The default title text is Mins.

Applies to
"Interval Prompt" on page 804

Multi-Line
Specifies whether to allow multi-line editing in the text control

Applies to
"Text Box Prompt" on page 838

Multi-Select
Specifies whether the control allows the selection of multiple values. Note that an associated parameterized filter expression will override this object's setting. If you edit this property but do not get the expected results, check the associated expression for the presence of an operator that specifies how many items can be selected. Examples of multiple selection operators are "in" and "not in"; examples of single selection operators are equal (=), less than (<) and greater than (>).

For example, if you used the prompt wizard to create a parameterized filter expression and selected one of the operators that specify selection rules, changing the value of this property is not sufficient to change this restriction. You must also edit the filter's expression to remove the offending operator.

To edit a filter expression, you must select the filter, which is accessible from the Query view, using the Explorer bar.

Applies to
"Data List Box" on page 795, "Date & Time Prompt" on page 796, "Date Prompt" on page 797, "Interval Prompt" on page 804, "List Box" on page 809, "Select & Search Prompt" on page 834, "Text Box Prompt" on page 838, "Time Prompt" on page 839, "Tree Prompt" on page 840, "Value Prompt" on page 841

Name
Specifies the name of the object.

Applies to
"Calculated Measure" on page 779, "Calculated Member" on page 779, "Custom Groups" on page 793, "Dimensional Edge Summary" on page 798, "Edge Summary" on page 799, "Explicit Member Set" on page 799, "Hierarchy Set" on page 802, "Intersection (tuple)" on page 804, "Level Set" on page 808, "List Summary" on page 813, "Measure" on page 817, "Member" on page 817, "Member Children Set" on page 817, "Member Property" on page 818, "Named Set" on page 819, "Percentaged Measure" on page 823, "Set Expression" on page 835

Name
Specifies the name of the object.
Applies to

“Page Set” on page 822

Name

Specifies the name of the object.

Applies to

“Page” on page 821

Name

Specifies a unique name that allows layout objects to be reused, usually to take advantage of any applied styling.

Applies to


Name

Specifies the name of the object.

Applies to

“Query” on page 828
**Name**

Specifies the unique name identifier for a query operation.

**Applies to**

“Query Operation” on page 828

**Name**

Specifies the unique name identifier for an SQL object.

**Applies to**

“SQL” on page 835

**Name**

Specifies the unique name identifier for an MDX object.

**Applies to**

“MDX” on page 817

**Name**

Specifies the name of the object.

**Applies to**

“Dimension” on page 798

**Name**

Specifies the name of the object.

**Applies to**

“Level” on page 807

**Name**

Specifies the name of the object.

**Applies to**

“Fact” on page 800

**Name**

Specifies the name of the object.

**Applies to**

“Member Set” on page 818
Name

Specifies a unique name that allows layout objects to be reused, usually to take advantage of any applied formatting.

Negative Column Color

Specifies the color, gradient, or pattern to be used for columns in the progressive chart that represent negative values.

Nested Label Display

Specifies how to display the axis labels for nested categories. Concatenated separates the labels with a comma.

Nesting Order

Specifies the nesting order of extra categories in the query. Change the nesting order of extra categories when items are from the same dimension. For example, a visualization has Year as categories, and Month and Quarter as extra categories. The order of categories must be Year, Quarter, and Month for the query to run.

No Data Contents

Specifies what appears in a data container when there is no data available from the database. When set to No Content, shows an empty data container, such as a blank list. When set to Content specified in the No data tab, shows the content that you specified in the No data tab. You can insert any object from the Toolbox tab. When set to Specified text (the default), shows the text that you specify.
No Data Features Size (pt)
Specifies the point size used for map point features that do not have corresponding data.

 Applies to
“Map” on page 814

Node Coordinate
Uniquely identifies a node member or spacer on an edge of a crosstab. Used by the Crosstab Intersection object to uniquely identify the intersection of elements from each edge. You cannot modify this value.

 Applies to
“Crosstab Node Member” on page 790, “Crosstab Space” on page 791, “Crosstab Summary” on page 791

Note Border
Specifies the properties for the border of a note.

 Applies to
“Note” on page 819

Notes
Specifies a block of text that you can position on a chart.

 Applies to

Notes
Specifies whether a note is rendered in a chart.

 Applies to
Note Text
Specifies the text that appears in a note within a chart. To edit the note text, you must switch to Page Design view (View > Page Design)

Applies to
“Note” on page 819

No Value List Item
Specifies whether to define a default list item. When the default item is selected, sets the value of the data item in the control to null. By default, the no value item is the first item in the list.

Applies to

Number of Minor Intervals
When minor gridlines are defined for the chart axis, specifies by how many minor intervals to divide the major intervals. The default value is 2, which shows one minor gridline between each major gridline.

Applies to

Number of Minor Intervals
When minor gridlines are defined for the chart axis, specifies by how many minor intervals to divide the major intervals. The default value is 2, which shows one minor gridline between each major gridline.

Applies to

Number of Regression Lines
Specifies whether there will be one regression line for all the data or one for each series.

Applies to
“Regression Line” on page 831

Numbers Only
Specifies whether the Text Box Prompt allows numbers only.

Applies to
“Text Box Prompt” on page 838
**Number Style**
Specifies the style to use for page numbers.

**Applies to**
- “Page Number” on page 821

**Number Style**
Specifies the number style to use for the iterator number.

**Applies to**
- “Iterator Label Area” on page 805

**Numerical Axis**
Specifies whether the axis is rendered.

**Applies to**
- “3-D Combination Chart” on page 772

**Numerical Axis**
Specifies whether the numeric axis of a gauge chart is rendered.

**Applies to**
- “Gauge Chart” on page 801

**Numeric Baselines**
Adds reference lines on the numeric axis of a chart based on numeric or statistical values, calculations, or layout calculations.

**Applies to**

**Numeric Value**
Specifies the numeric position by using a number.

**Applies to**
- “Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Marker” on page 816

**Numeric Value**
Specifies the value of the numeric position.

**Applies to**
- “Marker” on page 816
**Numeric Value**
Specifications the value of the numeric position.

**Applies to**

“Marker” on page 816

**Options Text**
Specifies the text for the additional prompt options link in select & search prompts. The default label text is Options.

**Applies to**

“Select & Search Prompt” on page 834

**Orientation**
Specifies the orientation of the control.

**Applies to**


**Outer Join Allowed**
Specifies whether outer joins are allowed on the object. This property applies to a single query and overrides the setting in IBM® Cognos® Framework Manager, the modeling tool.

**Applies to**

“Query” on page 828

**Output Aggregation Method**
Specifies the aggregation method.

**Applies to**

“Crosstab Summary” on page 791, “List Summary” on page 814

**Output Aggregation Method**
Specifies the aggregation method.

**Applies to**

“List Cell” on page 809
Override Dimension Info
Specifies dimension information for a query. Add dimension information when the data source has no dimension information available or when you want to override, extend, or restrict the information.

For more information, see the topic Add Dimension Information to a Query.

Applies to
“Query” on page 828

Overrides
Specifies whether to override child objects. Before you can override child objects, they must have a name.

Applies to
“Layout Component Reference” on page 806

Padding
Specifies the space between the object and the margin. If there is a border then it specifies the space between the object and the border.

Applies to

Pagination
Specifies pagination rules, such as page breaks, keep-with properties, page counts, and numbering.
Palette

Specifies the palette to use for the chart. This property applies only to legacy charts.

Applies to


Palette

Specifies the color palette for a chart.

Applies to


Palette

Map layers use the numeric palette, in which the color of a region or point is based on its numeric value.

Applies to

“Point Layer” on page 825, “Region Layer” on page 830

Parameter

Specifies the parameter that is satisfied by values chosen in the prompt control.

Applies to

**Pareto Axis**
Specifies whether the axis is rendered.

**Applies to**
“Pareto Chart” on page 823

**PDF Page Setup**
Specifies the orientation and paper size for PDF documents.

**Applies to**
“Page” on page 821

**Percentage Based On**
Specifies the information that is used to calculate the percentage values.

**Applies to**
“Percentaged Measure” on page 823

**Percentile**
Specifies a position based on a data percentile value. This value must be greater than zero.

**Applies to**
“Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Marker” on page 816

**Percentile**
Specifies a position based on a data percentile value. This value must be greater than zero.

**Applies to**
“Marker” on page 816

**Percentile**
Specifies a position based on a data percentile value. This value must be greater than zero.

**Applies to**
“Marker” on page 816

**Percent of Axis**
Specifies a position based on a percentage along the numeric axis. This value must be greater than zero.
Applies to

“Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Marker” on page 816

Percent of Axis
Specifies a position based on a percentage along the numeric axis. This value must be greater than zero.

Applies to

“Marker” on page 816

Percent of Axis
Specifies a position based on a percentage along the numeric axis. This value must be greater than zero.

Applies to

“Marker” on page 816

Performance Pattern
Controls what portions of the range markers for tolerance and target ranges are rendered on a metrics chart.

Applies to

“Metrics Range Chart” on page 818

Pie Labels
Specifies whether pie labels are rendered.

Applies to

“Pie Chart” on page 823

Plot Area Fill
Specifies the fill color and effects for the plot area of a chart.

Applies to

“Bubble Chart” on page 777, “Combination Chart” on page 786, “Pareto Chart” on page 822, “Progressive Chart” on page 826, “Scatter Chart” on page 833

Point Shape
Specifies the shape of the markers. If you choose a value of series or category, the marker shape varies accordingly.

Applies to

Polynomial Exponent
Specifies the highest exponential value to use in the regression calculation.

Applies to
“Regression Line” on page 831

Position
Specifies where to position the legend.

Applies to
“Legend” on page 806

Position
Specifies the position of a note in a chart.

Applies to
“Equation Label” on page 799, “Note” on page 819

Position in List
Specifies the position of the no list value item in the list.

Applies to
“No Value List Item” on page 819

Position type
Specifies the source type for the numeric position.

Applies to
“Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Marker” on page 816

Position type
Specifies the type of numeric position on the X-axis. This property applies only to legacy charts.

Applies to
“Marker” on page 816

Position type
Specifies a type of numeric position.

Applies to
“Marker” on page 816
Positive Column Color

Specifies the color, gradient, or pattern to be used for columns in the progressive chart that represent positive values.

Applies to

“Progressive Chart” on page 827

Pre-populate

Specifies whether to pre-populate the control with values, but only if the parent of this prompt control is optional. This only applies to prompt controls that have a parent in a cascade.

Applies to

“Tree Prompt” on page 840, “Value Prompt” on page 841

Pre-populate Levels

Specifies the number of levels to pre-populate the prompt with. The default value is 1, which will pre-populate the prompt with only the root members.

Applies to

“Tree Prompt” on page 840

Pre-Sort

Sorts the data that is used by the query to produce a temporary cube when needed, such as for a crosstab against a relational data source. This property affects the default order of members of a level populated from the data item.

Applies to

“Data Item” on page 795

Processing

Specifies whether the query engine will pick up a minimal amount of processing. Local processing only occurs if the database cannot handle the load. This property applies only to dimensional, or OLAP, data sources.

Specifies whether the IBM® Cognos® report server picks up a minimal amount of processing. This property overrides the corresponding governor in the model. If unspecified, the value of the governor in the model is used. This property applies only to relational data sources, unless you use the dynamic query mode.

Database only specifies that all query processing is performed on the database, and none is performed and cached by the IBM Cognos report server. The query executes successfully only if all query operations can be performed by the database.

Limited Local specifies that little, or possibly none, of the query processing is performed by the Cognos report server. Local processing only occurs if the database cannot handle the query.
When using the dynamic query mode, this property applies only to SAP BW data sources and the Limited Local option is not supported. Database Only specifies that little, or possibly none, of the query processing is performed and cached by the Cognos report server. Local processing only occurs if the database cannot handle the query. Consider using this option only if your report performance is unacceptable with the default setting, and becomes usable with this setting. Be aware that results may change with this setting; test carefully to confirm that the results are still correct.

**Applies to**

"Query“ on page 828

**Progressive Axis**

Specifies whether the axis is rendered.

**Applies to**

"Progressive Chart“ on page 827

**Progressive Palette**

Specifies the color palette in a progressive chart.

**Applies to**

"Progressive Chart“ on page 826

**Projection List**

Shows the list of projected data items for the set operation. You can automatically generate the list or manually add data items.

**Applies to**

"Query Operation“ on page 828

**Properties**

Specifies a list of data items from the query in scope to associate to the selected object. This is necessary when you want to reference a data item that is in the associated query, but not used in the layout.

**Applies to**

Property
Specifies a reference to a member property. You cannot modify this value.

Applies to
“Member Property” on page 818

Property Unique Name
Specifies the Member Property Unique Name (MPUN) of the member property.

Applies to

Push To Bottom
Specifies whether to position the footer as low as possible inside the parent object.

Applies to
“List Footer” on page 811, “List Page Footer” on page 813

Query
Specifies a reference to a query.

Applies to
Query

Specifies a reference to a query.

Applies to

“Page” on page 821, “Select & Search Prompt” on page 834, “Tree Prompt” on page 840, “Value Prompt” on page 841

Radar Type

Specifies how the radar chart is rendered.

Applies to

“Radar Chart” on page 828

Radial Axis

Specifies whether the axis is rendered.

Applies to

“Polar Chart” on page 825, “Radar Chart” on page 828

Radio Buttons Definition

Defines the buttons in the radio button group.

Applies to

“Radio Button Group” on page 829

Range

Specifies whether this control accepts ranges. The setting of the associated parameterized expression for this property will override the setting of this object. If you edit this property but do not get the expected results, check the associated expression for the presence or absence of an in_range operator.
For example, if you created this prompt control with the prompt wizard and set up an associated parameterized filter that accepts ranges, changing the value of this property is not sufficient to change this restriction. You must also edit the filter’s expression and remove the in_range operator.

To edit a filter expression, you must select the filter, which is accessible from the Query view, using the Explorer bar.

**Applies to**

“Date & Time Prompt” on page 796, “Date Prompt” on page 797, “Interval Prompt” on page 804, “Text Box Prompt” on page 838, “Time Prompt” on page 839, “Value Prompt” on page 841

**Range Label**

Specifies whether the label for the Target Range will appear in the legend.

**Applies to**

“Metrics Range Chart” on page 818

**Range Type**

Specifies whether to show one or two thumbs and what region of the track to highlight.

**Applies to**

“Continuous Values Slider” on page 787, “Data Discrete Values Slider” on page 794, “Discrete Values Slider” on page 798

**Regression Line**

Specifies whether a regression line is rendered. A regression line is a straight or curved line that best approximates the data points in the series.

**Applies to**

“Bubble Chart” on page 776, “Scatter Chart” on page 833

**Regression Type**

Specifies the type of regression used.

**Applies to**

“Regression Line” on page 831

**Relative Alignment**

Specifies how to vertically align this object, relative to its siblings.

**Applies to**

Remove Text

Specifies the label that appears on the button that is used to remove items from the selected items box in all multiple selection prompts. The default label text is Remove.

Applies to

“Date & Time Prompt” on page 796, “Date Prompt” on page 797, “Interval Prompt” on page 804, “Select & Search Prompt” on page 834, “Text Box Prompt” on page 838, “Value Prompt” on page 841

Render

For list reports, specifies whether to render the column in the report output. Whether set to Yes or No, the query for the column is always executed.

Applies to

“List Column” on page 809

Render Fact Cells

Specifies whether to render values in the fact cells of the spacer on an edge of the crosstab.

Applies to

“Crosstab Space” on page 791

Render Page when Empty

Specifies whether to render a page when data containers on a page contain no data. If all data containers on a page have this property set to No and do not have any data to render, the page is not rendered. If any data container on a page has data or has this property set to Yes, the page is rendered.

Applies to

**Render Variable**

Specifies a variable based on which the object can be conditionally rendered.

**Applies to**

- “3-D Combination Chart” on page 772
- “3-D Scatter Chart” on page 772
- “Axis Labels” on page 774
- “Axis Title” on page 774
- “Block” on page 776
- “Bubble Chart” on page 776
- “Bubble Chart” on page 777
- “Bullet Chart” on page 777
- “Button” on page 778
- “Button Bar” on page 778
- “Category Baseline” on page 781
- “Chart Footer” on page 781
- “Chart Subtitle” on page 781
- “Chart Text Item” on page 781
- “Chart Title” on page 781
- “Check Box Group” on page 784
- “Colored Region” on page 785
- “Colored Region” on page 785
- “Combination Chart” on page 785
- “Combination Chart” on page 785
- “Context Item Text” on page 787
- “Continuous Values Slider” on page 787
- “Crosstab” on page 788
- “Data Button Bar” on page 793
- “Data Check Box Group” on page 794
- “Data Deck” on page 794
- “Data Discrete Values Slider” on page 794
- “Data Drop-Down List” on page 794
- “Data Iterator” on page 795
- “Data List Box” on page 795
- “Data宗教 Button Group” on page 795
- “Data Tab Control” on page 796
- “Data Toggle Button Bar” on page 796
- “Date & Time Prompt” on page 796
- “Date Prompt” on page 797
- “Deck” on page 797
- “Discrete Values Slider” on page 798
- “Drop-Down List” on page 798
- “Field Set” on page 800
- “Gauge Chart” on page 800
- “Gauge Chart” on page 800
- “Generated Prompt” on page 802
- “HTML Item” on page 802
- “Hyperlink” on page 802
- “Hyperlink Button” on page 803
- “Image” on page 803
- “Interval Prompt” on page 804
- “Iterator” on page 804
- “Label” on page 806
- “Legend” on page 807
- “Legend Title” on page 807
- “List” on page 808
- “List Box” on page 809
- “List Column” on page 809
- “List Summary” on page 814
- “Map” on page 814
- “Marimekko Chart” on page 815
- “Marker” on page 816
- “Metrics Range Chart” on page 818
- “Note” on page 819
- “Numeric Baseline” on page 820
- “Page” on page 821
- “Page Break Text” on page 821
- “Pareto Chart” on page 822
- “Pareto Chart” on page 822
- “Pie Chart” on page 823
- “Pie Chart” on page 823
- “Polar Chart” on page 825
- “Progressive Chart” on page 826
- “Progressive Chart” on page 826
- “Radio Button Group” on page 829
- “Repeater” on page 831
- “Repeater Table” on page 831
- “Repeater Table Cell” on page 831
- “Rich Text Item” on page 832
- “Scatter Chart” on page 832
- “Scatter Marker” on page 834
- “Select & Search Prompt” on page 834
- “Singleton” on page 835
- “Static Repeater Table” on page 835
- “Tab Control” on page 836
- “Table” on page 837
- “Table of Contents” on page 837
- “Table of Contents Entry” on page 838
- “TextBox Prompt” on page 838
- “Text Item” on page 839
- “Time Prompt” on page 839
- “Toggle Button Bar” on page 840
- “Tree Prompt” on page 840
- “Trendline” on page 841
- “Value Prompt” on page 841
- “Variable Text Item” on page 842
- “Visualization” on page 842
- “Win-Loss Chart” on page 842

**Repeater Direction**

Specifies the direction in which to populate the rendered repeater cells.
Repeater Table Values Definition

Defines the data values that can appear in the repeater table. Values are organized by data item. In addition to the default data item (Label) and values provided, you can create your own data items and values.

Applies to

“Static Repeater Table” on page 835

Report Expression

Specifies the report expression that defines the text to render.

Applies to

“Chart Text Item” on page 782, “Chart Text Item” on page 783, “Cumulation Line Label” on page 792, “Hyperlink” on page 802, “Hyperlink Button” on page 803, “Text Item” on page 839, “Total Column” on page 840

Report Expression

Specifies the report expression used to define the HTML to render.

Applies to

“HTML Item” on page 802, “Rich Text Item” on page 832

Report Expression

Specifies the report expression that defines the URL.

Applies to

“Hyperlink” on page 802, “Hyperlink Button” on page 803, “Image” on page 803

Report Expression

Specifies the report expression that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to

“Bookmark” on page 776

Report Expression

Specifies the numeric position by using a report expression.

Applies to

“Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Marker” on page 816
**Report Expression**  
Specifies the report expression for the member position.

**Applies to**

“Baseline” on page 775, “Marker” on page 816

**Report Expression**  
Specifies a report expression.

**Applies to**

“As of Time Expression” on page 773

**Report Expression**  
Specifies the expression to evaluate when determining the value for this variable.

**Applies to**

“Variable” on page 842

**Report Expression**  
Specifies the numeric position from a report expression.

**Applies to**

“Marker” on page 816

**Report Expression**  
 Specifies the numeric position from a report expression.

**Applies to**

“Marker” on page 816

**Required**  
Specifies whether the prompt is required or optional. If this property is set to required, the prompt must have a value entered before the report can be run. The Usage setting of the associated parameterized filter for this property will override the setting of this object. If you edit this property, but do not get the expected results, verify the **Usage** setting of the associated filter.

For example, if you created this prompt control with the prompt wizard and set the associated parameterized filter to be optional, changing the value of this property is not sufficient to change this setting. You must also edit the filter's **Required** property to match the setting for this object's **Required** property.

To edit a filter expression, you must select the filter, which is accessible from the Query view, using the Explorer bar.
Applies to


Results Deselect All Text
Specifies the text for the link below the results box that deselects all the items in the box. This property applies to all prompts with multiple selections and search, tree prompts, and value prompts. The default link text is Deselect All.

Applies to

“Select & Search Prompt” on page 834, “Tree Prompt” on page 840, “Value Prompt” on page 841

Results Select All Text
Specifies the text for the link below the results box that selects all the items in the box. This property applies to all prompts with multiple selections and search, tree prompts, and value prompts. The default link text is Select All.

Applies to

“Select & Search Prompt” on page 834, “Tree Prompt” on page 840, “Value Prompt” on page 841

Results Text
Specifies the title that appears above the results box in select & search prompts. The default title text is Results.

Applies to

“Select & Search Prompt” on page 834

Reverse Category Order
Specifies whether to change the order of the categories, such as the bars in a bar chart. The default is No, which means that for a horizontal bar chart, bars start from the bottom to the top.

Applies to

“Category axis” on page 780

Right Position
Specifies the position of the right edge of the colored region.

Applies to

“Colored Region” on page 785
Right Position (px)
Specifies the pixel position of the right edge of the legend measured from the left edge of the chart.

Applies to
“Legend” on page 806

Rollup Aggregate Function
Specifies the type of aggregation to apply to summarized values. These values appear at the higher levels of lists and crosstabs. For OLAP data sources, a rollup aggregate function of Count Distinct is supported for only levels and member sets.

Applies to

Rollup Method
Specifies the aggregation method to use to summarize data in the visualization when users select multiple values in an Active Report control, such as a check box control.

Applies to
“Value” on page 841

Rollup Processing
Specifies where to compute aggregates. The Extended setting means that aggregates are computed using an extended aggregate operation. The Database setting means that aggregates are computed by the database software. The Local setting means that aggregates are computed by the data retrieval software in the report server, using a running aggregate.

Applies to
“Query” on page 828

Root Members Only
Specifies whether the set contains the root members or all of the members of the hierarchy.
Applies to


Rotate Values

Controls whether or not the values displayed on the chart are rotated when the Chart Orientation property is set to Horizontal. May help make values easier to read on horizontal charts. This property applies only to legacy charts.

Applies to

“Combination Chart” on page 785, “Pareto Chart” on page 823, “Progressive Chart” on page 827

Row Coordinate

Uniquely identifies the row of a node member or spacer on an edge of the crosstab. You cannot modify this value.

Applies to

“Crosstab Intersection” on page 789

Rows Per Page

Specifies the maximum number of rows to show at one time. For Value and Select & Search prompts, allows you to extend or reduce the maximum number of rows beyond the default of 5000.

Applies to


Scale

Specifies whether the numeric scale is logarithmic or linear.

Applies to

Scale Interval
Specifies the interval between ticks on the numeric scale. If no value is specified, one will be calculated based on the data.

Applies to

Scope
Specifies the scope of the filter in terms of the number of levels.

Applies to
“Summary Filter” on page 836

Search Instructions Text
Specifies the instructions that appear above the keyword search box in select & search prompts. The default text is as follows: Type one or more keywords separated by spaces.

Applies to
“Select & Search Prompt” on page 834

Search Text
Specifies the label that appears on the Search button in select & search prompts. The default label text is Search.

Applies to
“Select & Search Prompt” on page 834

Seconds Text
Specifies the title that appears above the seconds box in interval prompts. The default title text is s.

Applies to
“Interval Prompt” on page 804

Select UI
Specifies the user interface to use for the button in the iterator.

Applies to
“Iterator First” on page 805, “Iterator Last” on page 805, “Iterator Next” on page 805, “Iterator Previous” on page 805
Select UI
Specifies the user interface to use for the labels of iterator values.

Applies to
“Iterator Label Area” on page 805

Select UI
Specifies which interface the prompt control renders.

Applies to
“Value Prompt” on page 841

Select UI
Specifies which interface the prompt control renders.

Applies to
“Date Prompt” on page 797

Select UI
Specifies which interface the prompt control renders.

Applies to
“Time Prompt” on page 839

Select UI
Specifies which interface the prompt control renders.

Applies to
“Date & Time Prompt” on page 796

Separator
Specifies the separator to use when showing multiple values.

Applies to
“Context Item Text” on page 787, “Legend” on page 806

Series Color
Specifies whether data marker colors are synchronized between charts when you use the same data series for multiple charts within a combination chart. For example, if you create a line chart and a bar chart using the years data series, and you set this property to Yes, the data marker for 2010 is the same color in both charts. Use this property to eliminate redundant legend entries and help you interpret the data.
Applies to

“3-D Combination Chart” on page 772, “Combination Chart” on page 785, “Combination Chart” on page 786

Series Type
Specifies how the series will be rendered; for example, absolute, stacked, or stacked 100%.

Applies to

“Area” on page 773, “Bar” on page 774, “Line” on page 808

Set Definition
Specifies a set of members.

Applies to


Set Operation
Specifies the set operation to apply to one or more queries, that results in a projection list on which other queries can be based.

Applies to

“Query Operation” on page 828

Set Sorting
Specifies how the set is sorted. By default, the set is not sorted.

Applies to

Set Variable Values
Sets the value of variables when the control or an item in the control is selected.

Applies to


Share Result Set
Specifies whether to share an identical query between data containers that use it. To share a query, data containers must be lists, repeaters, or repeater tables and must use the same grouping structure and list of properties. The data containers cannot be part of a master detail relationship. When set to Yes, the query sends only one request to the database and shares the result. When set to No, the query is not shared.

Applies to

“List” on page 808, “Repeater” on page 831, “Repeater Table” on page 831

Sharing
The properties that enable the set to be shared.

Applies to

Show Caption
Specifies whether, or where, to show the caption.
Applies to

“Field Set” on page 800

Show Data Points
Specifies whether to show value markers and how they are formatted.

Applies to

“Line” on page 808

Show Data Range in Legend
Specifies whether to show the full range of data in the legend. If set to No, only the values from the palette will be shown.

Applies to

“Point Layer” on page 825, “Region Layer” on page 830

Show Features with No Data
Indicates whether to show the features of a map that do not have corresponding data.

Applies to

“Point Layer” on page 825, “Region Layer” on page 830

Show Feelers
Specifies whether feeler lines are rendered for each marker.

Applies to

“3-D Scatter Chart” on page 772

Show Hover
Specifies whether to highlight areas in the data container that are clickable.

Applies to

Show Icon
When defined, specifies whether to show the icon for each list item in the drop-down list.

Applies to
“Data Drop-Down List” on page 794, “Drop-Down List” on page 799

Show Icon
Specifies whether to show the icon for the current label.

Applies to
“Iterator Label Area” on page 805

Show Labels
Specifies whether labels are rendered for the display layer in a map.

Applies to
“Display Layer” on page 798

Show Legend Values
Specifies whether and how to show legend values.

Applies to
“Legend” on page 806

Show line
Specifies whether a line may be rendered. This allows you to show markers without lines.

Applies to
“Line” on page 808

Show Pointer Cursor
Specifies whether to show the pointer cursor as a hand when hovering over an area that is clickable. You can set the value of this property for multiple objects simultaneously by first control+clicking the items in the report.

Applies to
Show Tooltip
Specifies whether to show tooltips of the slider values when the thumb is clicked or moved.

Applies to
“Continuous Values Slider” on page 787, “Data Discrete Values Slider” on page 794, “Discrete Values Slider” on page 798

Show Values
Specifies which values to show in a chart legend.

Applies to
“Legend” on page 807

Show Values
Specifies whether values and labels are rendered for the region or point layer in a map.

Applies to
“Point Layer” on page 825, “Region Layer” on page 830

Show Values
Specifies the appearance of data labels in area, bar, line, Pareto, and progressive charts.

Applies to

Show Values
Specifies whether to show the labels for the data markers on the cumulation line in a Pareto chart.

Applies to
“Cumulation Line” on page 792

Show Values
Specifies the appearance of data labels in a pie chart.

Applies to
“Pie Chart” on page 824

Show Values
Specifies the appearance of data labels in a scatter chart.
Applies to

“Scatter Chart” on page 833

Show Values
Specifies the appearance of data labels in a bubble chart.

Applies to

“Bubble Chart” on page 777

Size
Specifies the height and width of the control.

Applies to


Size & Overflow
Specifies the height and width of the object, as well as how overflow content should be treated, using scroll bars and clipping.

Applies to


Size & Overflow
Specifies the absolute height and width of the object, as well as how overflow content should be treated, using scroll bars and clipping.
Applies to


Size Legend Title
Specifies a title within the legend above the palette for the point size. If this object is not defined, no additional title is drawn. If no legend is drawn, this object is ignored. Styling for this object is inherited from the legend title.

Applies to

“Point Layer” on page 825

Slice Direction
Specifies the direction in which slices appear in a pie chart.

Applies to

“Pie Chart” on page 824

Slide Animation Direction
Specifies the direction from which a new card appears. Select Auto Horizontal or Auto Vertical to automatically set the direction based on the order of cards in the deck. For example, when Auto Horizontal is specified, if you view the first card in a deck and you select to view the fourth card, the fourth card appears from the right. If you then select to view the first card, the first card appears from the left.

Applies to

“Data Deck” on page 794, “Deck” on page 797

Slider Values Definition
Defines the values of the slider.

Applies to

“Discrete Values Slider” on page 798

Solve Order
Specifies the solve order in the crosstab. The item with the lowest solve order value is calculated first, followed by the next lowest value, and so on. For identical values, column items are calculated first, then row items, and then the measure.
Applies to

“Crosstab Summary” on page 791

Solve Order

Specifies the solve order in crosstabs and charts. The item with the lowest solve order value is calculated first, followed by the next lowest value, and so on. For identical values, in crosstabs, column items are calculated first, then row items, and then the measure. In charts, x-axis items are calculated first and then legend items.

Applies to

“Calculated Measure” on page 779, “Calculated Member” on page 779, “Data Item” on page 795, “Dimensional Edge Summary” on page 798, “Edge Summary” on page 799

Sorting

Specifies the desired sort sequence.

Applies to


Source Type

Specifies the source type of the text.

Applies to

“Chart Text Item” on page 782, “Chart Text Item” on page 783, “Cumulation Line Label” on page 792, “Hyperlink” on page 802, “Hyperlink Button” on page 803, “Text Item” on page 839

Source Type

Specifies the source type of the HTML text.

Applies to

“HTML Item” on page 802, “Rich Text Item” on page 832

Source Type

Specifies the source type of the URL.

Applies to

“Hyperlink” on page 802, “Hyperlink Button” on page 803, “Image” on page 803
Source Type
Specifies the source type of the bookmark.

**Applies to**

“Bookmark” on page 776

Spacing & Breaking
Specifies text properties such as line height, letter spacing, and word breaking.

**Applies to**


Spider Effects
Specifies whether the chart is rendered with web-like flat concentric circles. The Radial Axis property must be set to Show for this property to take effect.

**Applies to**

“Polar Chart” on page 825, “Radar Chart” on page 828

SQL
The text of the typed-in SQL. It is assumed to be appropriate for the type and data source. If it is not the query may fail, or produce unexpected results.

**Applies to**

“SQL” on page 835

SQL Join Syntax
Controls the syntax to generate for joins. Click Implicit to generate joins in the WHERE clause. Click Explicit to generate INNER JOIN syntax. If unspecified, the value of the corresponding governor in the model is used.

**Applies to**

“Query” on page 828
SQL Syntax
Specifies the syntax of the SQL in the query. A value of Cognos SQL indicates that IBM® Cognos® extended SQL-92 syntax is used. A value of Native SQL indicates that native database SQL is used. You cannot use IBM Cognos SQL if the Processing property for the query is set to Database Only; it must have a value of Limited Local. A value of Pass-Through indicates that a standalone query text syntax is used. The default value is Native SQL.

Applies to
“SQL” on page 835

Standard Deviations
Specifies a distance from the mean in standard deviations. This value can be positive or negative. A value of zero indicates the mean value.

Applies to
“Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Baseline” on page 775, “Marker” on page 816

Standard Deviations
Specifies a distance from the mean in standard deviations. This value can be positive or negative. A value of zero indicates the mean value.

Applies to
“Marker” on page 816

Standard Deviations
Specifies a distance from the mean in standard deviations. This value can be positive or negative. A value of zero indicates the mean value.

Applies to
“Marker” on page 816

Start Position
Specifies the position of one edge of the colored region along the numeric axis. The region extends from the position defined in this property to the position defined in the End Position property. The value that you specify in this property must be less than the value that is specified in the End Position property.

Applies to
“Colored Region” on page 785

Static Choices
Represents a collection of static choices used by the prompt object.

Applies to
“Select & Search Prompt” on page 834, “Value Prompt” on page 841
Step

Specifies at which intervals the thumb can come to a rest.

Applies to

“Continuous Values Slider” on page 787

Style Variable

Specifies a variable based on which the object can be conditionally styled.

Applies to


Subtitle

Specifies whether a chart subtitle is rendered. This property applies only to legacy charts.

Applies to

“Win-Loss Chart” on page 842
Subtitle
Specifies whether a chart subtitle is rendered.

Applies to

Summarize Small Items
Specifies whether to summarize small items, such as slices, lines, areas, bars, or columns, in the chart and how they are summarized. You cannot summarize small items in charts that have matrix edges or in charts that have multiple numeric axes.

Applies to
“Combination Chart” on page 786, “Pareto Chart” on page 822

Summarize Small Slices
Specifies whether to summarize small items, such as slices, lines, areas, bars, or columns, in the chart and how they are summarized. You cannot summarize small items in charts that have matrix edges or in charts that have multiple numeric axes.

Applies to
“Pie Chart” on page 824

Summary Text
Specifies summary text for table-like objects. Use to make your reports accessible for people who use screen readers. The summary text is never displayed in visual Web browsers. Summary text is used only for screen readers and speech browsers. Set this property only on tables that are used to display data, and not on tables that are used for layout purposes. To have a table object function as a data table, ensure that one or more of its cells have the Table Header property set to Yes.

Applies to
“Crosstab” on page 788, “List” on page 808, “Repeater Table” on page 831, “Table” on page 837

Suppress
Specifies the type of suppression to apply to the query results. This property overrides the corresponding SAP BW governor in the model. If unspecified, the value of the governor in the model is used.

When using compatible query mode, this property applies to Essbase, SAP BW, and MSAS data sources. When using dynamic query mode, this property applies to all OLAP data sources, including OLAP over relational (DMR).

Applies to
“Query” on page 828

Suppression
Specifies zero suppression options for the object.
Applies to


Table Header

Specifies whether the cell is a table header. Use to make reports accessible for people who use screen readers. When set to Yes, screen readers and speech browsers programmatically create relationships between the table header and table cells.

Applies to

“List Cell” on page 809, “Table Cell” on page 837

Table of Contents

Specifies the name of the table of contents to which the entry belongs.

Applies to

“Table of Contents Entry” on page 838

Table of Contents Name

Specifies the name that uniquely identifies the table of contents.

Applies to

“Table of Contents” on page 837

Table Properties

Specifies the properties for the table object.

Applies to

“Crosstab” on page 788, “List” on page 808, “Repeater Table” on page 831, “Static Repeater Table” on page 835, “Table” on page 837

Tab Orientation

Specifies the orientation of the tabs in the tab control.
Applies to

“Data Tab Control” on page 796, “Tab Control” on page 836

Tabs Definition
Defines the tabs in the tab control.

Applies to

“Tab Control” on page 836

Tab Width
Specifies the width of tabs in the tab control.

Applies to

“Data Tab Control” on page 796, “Tab Control” on page 836

Target Color
Specifies a color for the vertical lines that mark the target ranges for target measure values in a metrics chart.

Applies to

“Metrics Range Chart” on page 818

Target Marker
Specifies whether the status indicators will appear in the legend.

Applies to

“Metrics Range Chart” on page 818

Target Marker Border Color
Specifies a color for the borders around target value markers in a metrics chart.

Applies to

“Metrics Range Chart” on page 818

Target Marker Position
Specifies whether the status indicators will be rendered over the first bar in the cluster or the middle of the cluster. Does not apply to stacked charts.

Applies to

“Metrics Range Chart” on page 818

Target Range (%)
Specifies target ranges centered around target measure values.
Applies to

“Metrics Range Chart” on page 818

Text

Specifies the static text to render.

Applies to

“Chart Text Item” on page 782, “Chart Text Item” on page 783, “Cumulation Line Label” on page 792, “Hyperlink” on page 802, “Hyperlink Button” on page 803, “Text Item” on page 839, “Total Column” on page 840

Text Color

Specifies the color of the text of each label.

Applies to

“Check Box Group” on page 784, “Data Check Box Group” on page 793, “Data Radio Button Group” on page 795, “Radio Button Group” on page 829

Text Color

Specifies the text color of the button label.

Applies to

“Iterator First” on page 805, “Iterator Label Area” on page 805, “Iterator Last” on page 805, “Iterator Next” on page 805, “Iterator Previous” on page 805

Text Orientation

Specifies the orientation of the text on the gauge axis.

Applies to

“Gauge Axis” on page 800

Text Source Variable

Specifies a variable based on which the text source can be chosen.

Applies to

“Chart Text Item” on page 782, “Chart Text Item” on page 783, “Cumulation Line Label” on page 792, “Hyperlink” on page 802, “Hyperlink Button” on page 803, “Text Item” on page 839

Text Truncation

Specifies whether and how legend items are truncated.

Applies to

“Axis Labels” on page 774, “Legend” on page 807
**Tick Interval**

Specifies the interval, in slider values, between the appearance of ticks. For example, if you enter 3, a tick will appear for every third value in the slider. To apply this property, a value other than None or No Labels must be specified for the Tick Type property.

**Applies to**

"Data Discrete Values Slider" on page 794, "Discrete Values Slider" on page 798

**Tick Interval**

Specifies the interval, in slider values, between the appearance of ticks. For example, if you enter 20 in a slider that contains values between 0 and 100, a tick appears at 20, 40, 60, 80, and 100. To apply this property, a value other than None or No Labels must be specified for the Tick Type property.

**Applies to**

"Continuous Values Slider" on page 787

**Tick Label Skip Interval**

Specifies the interval, in ticks, between labels. This helps to avoid label collisions in the slider. If there is a collision between two labels, the second label is not shown.

**Applies to**

"Continuous Values Slider" on page 787, "Data Discrete Values Slider" on page 794, "Discrete Values Slider" on page 798

**Tick Type**

Specifies the type of label to show for the ticks in the slider.

**Applies to**

"Continuous Values Slider" on page 787, "Data Discrete Values Slider" on page 794, "Discrete Values Slider" on page 798

**Title**

Specifies whether a chart title is rendered. This property applies only to legacy charts.

**Applies to**

"Win-Loss Chart" on page 842

**Title**

Specifies whether a chart title is rendered.

**Applies to**

Title

Specifies whether a chart title is rendered.
Tolerance Color
Specifies a color for the vertical lines that mark the tolerance ranges for target measure values in a metrics chart.

 Applies to
“Metrics Range Chart” on page 818

Tolerance Label
Specifies whether the label for the Target Tolerance will appear in the legend.

 Applies to
“Metrics Range Chart” on page 818

Tooltips
Specifies whether tooltips are shown in a chart when you hover over data elements. Tooltips are not supported in PDF documents.

 Applies to
“Combination Chart” on page 786, “Pareto Chart” on page 822, “Progressive Chart” on page 826

Tooltips
Specifies whether tooltips are shown in a pie chart when you hover over data elements. Tooltips are not supported in PDF documents.

 Applies to
“Pie Chart” on page 824

Tooltips
Specifies whether tooltips are shown in a chart when you hover over data elements. Tooltips are not supported in PDF documents.

 Applies to
**Tooltips**
Specifies whether tooltips are shown in a visualization when you hover over data elements. Tooltips are not supported in PDF documents.

**Applies to**

- "Data Container" on page 793

**Tooltips**
Specifies whether tooltips are shown in the chart when you hover over data elements. Tooltips are not supported in PDF documents. This property applies only to legacy charts.

**Applies to**


**Top Position**
Specifies the position of the top edge of the colored region.

**Applies to**

- "Colored Region" on page 785

**Top Position (px)**
Specifies the pixel position of the top edge of the legend measured from the bottom of the chart.

**Applies to**

- "Legend" on page 806

**Total Column**
Specifies whether a total column is rendered. This property applies only to legacy charts.

**Applies to**

- "Progressive Chart" on page 827

**Total Column**
Specifies whether a bar that shows the total cumulative value is rendered.

**Applies to**

- "Progressive Chart" on page 826
**Total Column Color**
Specifies the color, gradient, or pattern of the total column on the progressive chart.

**Applies to**
“Total Column” on page 840

**To Text**
Specifies the label that appears beside the end of a range. This property applies to the following prompt types: date, date & time, time, and interval. The default label text is To.

**Applies to**
“Date & Time Prompt” on page 796, “Date Prompt” on page 797, “Interval Prompt” on page 804, “TextBox Prompt” on page 838, “Time Prompt” on page 839, “Value Prompt” on page 841

**Track Length**
Specifies the track length of the slider.

**Applies to**
“Continuous Values Slider” on page 787, “Data Discrete Values Slider” on page 794, “Discrete Values Slider” on page 798

**Trendline Label**
Specifies whether to show the default label for the trend line. When set to No, you can type your own label text.

**Applies to**
“Trendline” on page 841

**Trendlines**
Adds a trend line or curve that indicates the general direction of the data over time.

**Applies to**
“Bubble Chart” on page 777, “Combination Chart” on page 786, “Scatter Chart” on page 833

**Truncation**
Specifies whether labels can be truncated.

**Applies to**
Truncation Text

Specifies the text to append when a label is truncated.

Applies to

"Category Axis" on page 780, "Gauge Labels" on page 801, "Legend" on page 806,
"Pie Labels" on page 824, "X Axis" on page 843, "Y Axis" on page 845

Type

Specifies the type of variable.

Applies to

"Variable" on page 842

Type

Specifies the type of trend line.

Applies to

"Trendline" on page 841

Type

Specifies the behavior of the prompt button.

Applies to

"Prompt Button" on page 827

Type

Specifies the type of object.

Applies to

"Calculated Measure" on page 779, "Calculated Member" on page 779, "Custom
Groups" on page 793, "Dimensional Edge Summary" on page 798, "Edge
Summary" on page 799, "Explicit Member Set" on page 799, "Hierarchy Set" on
page 802, "Intersection (Tuple)" on page 804, "Level Set" on page 808, "List
Summary" on page 813, "Measure" on page 817, "Member" on page 817, "Member
Children Set" on page 817, "Member Property" on page 818, "Named Set" on page
819, "Percentaged Measure" on page 823, "Set Expression" on page 835

Update Variables Live

Specifies whether to update variables as the thumb is moved. When set to Yes, the
report is updated live when the thumb is moved. When set to No, the report is
updated only when the thumb is stopped and released.

Applies to

"Continuous Values Slider" on page 787, "Data Discrete Values Slider" on page
794, "Discrete Values Slider" on page 798
Upper Range Skew (%)
Specifications a percentage that affects the positioning of tolerance bar and range with respect to the target value.

Applies to

“Metrics Range Chart” on page 818

URL
Specifies the URL, using static text.

Applies to

“Hyperlink” on page 802, “Hyperlink Button” on page 803

URL
Specifies the URL, using static text.

Applies to

“Image” on page 803

URL Source Variable
Specifies a variable based on which the URL source can be chosen.

Applies to

“Hyperlink” on page 802, “Hyperlink Button” on page 803, “Image” on page 803

Usage
Specifies whether the usage of this object is Optional, Required, or Disabled. When Optional, this condition applies if all parameters referenced by the expression are provided with values. If the expression does not refer to any parameters, then this condition is always applied. When Disabled, this condition is never applied, which is useful for authoring and testing purposes.

Applies to

“Detail Filter” on page 798, “Summary Filter” on page 836

Use 1.x Behavior
Specifies that IBM® Cognos® ReportNet® query semantics are used if they differ from IBM Cognos Business Intelligence query rules.

Applies to

“Query” on page 828

Use Aggregate Cache
For SAP BW data sources, specifies whether to use the aggregation cache. We recommend that you use the default value.
**Use Detail Value on Page**
Specifies whether to render a detail value or an aggregate value for a text item that uses a data item as its source. Use this property only when you want to render the value that appears in the first or last detail row of a list, repeater or repeater table on the same page as the text item.

**Use Embedded Data**
Specifies whether to use data that is defined in the visualization definition instead of data that you insert from the package.

**Use for Parameter Info**
Specifies whether the query should be given priority when determining parameter information. Queries with this property set to Yes are checked for parameter information first, followed by queries with this property set to Default. Queries with this property set to No will not be checked for parameter information unless it is referenced in a query that will be checked. Setting this property to Yes on parameterized queries can improve performance in displaying prompt pages.

**Use Local Cache**
Specifies whether a query is a candidate for reusing the query results set. If set to Yes, the query engine can reuse the existing SQL results from the cache. If set to No, the query is executed rather than using cached results. This property applies only to relational and dimensionally-modeled relational (DMR) data sources.

**User SAP Member Cache**
Signals to the SAP BW provider whether the query associated with this property is cached to the IBM® Cognos® Business Intelligence member cache. When set to Yes, the member cache is populated with the dimensions in the query, encrypted, and saved for later use.
Use Same Range For All Instances

Specifies that all instances of the chart use the same maximum value. When set to No, the axis maximum value is recalculated for each chart instance. It is only relevant if the chart is involved in a master detail relationship.

Applies to


Use SAP MUN as Business Key

Specifies whether to return the full SAP MUN as the value for the business key. Use this query hint to allow a drill-down on a business key for a non-leaf member of an external hierarchy. When set to Yes, the full MUN appears in the report output.

Applies to

"Query" on page 828

Use Set Aggregation

Specifies which aggregation clause is used to calculate the summary. When set to Yes, the aggregation clause within set is used. When set to No, the aggregation clause within detail is used.

For more information, see the topic Summarizing Values in Crosstabs.

Applies to

"Dimensional Edge Summary" on page 798, "Edge Summary" on page 799

Use SQL Parameters

Specifies whether the generated SQL uses parameter markers or literal values. When set to Marker, specifies that the generated SQL uses markers to denote that the value will be provided later. When set to Literal, uses literal values in the generated SQL. If not specified, the server determines the behavior.

Dynamic SQL applications have the ability to prepare statements which include markers in the text which denote that the value will be provided later. This is most efficient when the same query is used many times with different values. The technique reduces the number of times a database has to hard parse an SQL statement and it increases the re-use of cached statements. However, when queries navigate larger amounts of data with more complex statements, they have a lower chance of matching other queries. In this case, the use of literal values instead of markers may result in improved performance.

Applies to

"Query" on page 828
Use SQL With Clause
Specifies whether to send a request to the database using an SQL WITH clause. When set to Yes, and if the database supports WITH clauses, a WITH clause request is generated. When set to No, or if the database does not support WITH clauses, a request using derived tables is generated.

Applies to
“Query” on page 828

Use Thousands Separator
Specifies whether to delimit digit groups with the thousands separator.

Applies to
“Text Box Prompt” on page 838

Use Value
Specifies the values used by the prompt object.

Applies to
“Tree Prompt” on page 840

Use Value
Specifies the values used by the prompt object. These values can be different than the ones that are rendered to the user.

Applies to
“Select & Search Prompt” on page 834, “Value Prompt” on page 841

Value Location:
Specifies where values and labels are to be rendered in the chart.

Applies to

Value Markers
Specifies whether to show special value markers and how they are formatted.

Applies to
“Line” on page 808
Value Range
Controls how the minimum and maximum are calculated for the visualization. When all data is selected, the minimum and maximum are calculated to span all associated Active Report control filters. When filtered data is selected, the minimum and maximum are calculated based on any filter choices that users make in associated Active Report controls. When specified values is selected, you explicitly specify the minimum and maximum.

Applies to
“Value” on page 841

Value Representation
Specifies whether values are rendered as percentages.

Applies to
“Pie Chart” on page 823

Values
Specifies what values to show in the chart and whether to show the corresponding measure, series, or category label.

Applies to
“3-D Scatter Chart” on page 772, “Bubble Chart” on page 776, “Polar Chart” on page 825, “Scatter Chart” on page 833

Values
Specifies whether values are rendered in the chart.

Applies to

Values
Specifies whether values are rendered.

Applies to
“Pie Chart” on page 823

Values
Specifies whether values are rendered in the chart.

Applies to
“Marimekko Chart” on page 815
Value Type
Specifications whether absolute values are rendered rather than cumulative values.

Applies to

“Area” on page 773, “Bar” on page 774, “Line” on page 808

Variable
Specifies the variable associated to the slider.

Applies to

“Continuous Values Slider” on page 787

Variable for Maximum
Specifies the variable to use for the maximum value when the slider range type is Range.

Applies to

“Continuous Values Slider” on page 787

Variable for Minimum
Specifies the variable to use for the minimum value when the slider range type is Range.

Applies to

“Continuous Values Slider” on page 787

Vertical Alignment
Specifies how objects contained in this object are vertically aligned.

Applies to


Visible
Specifies whether a column or row is visible when the active report is executed.
You can set the value of this property for multiple objects simultaneously by first control+clicking the items in the report.
Visible

Specifies whether to display the object. When set to No, the object is hidden but a fixed space is reserved in the report.

Applies to


Visual Angle

Specifies the angle, in degrees, in which the chart objects will be displayed when the chart has 3-D effects. This property applies only to legacy charts.

Applies to

“Combination Chart” on page 785, “Marimekko Chart” on page 815, “Metrics Range Chart” on page 818, “Pareto Chart” on page 823, “Progressive Chart” on page 827

White Space

Specifies how to handle the white space inside the object and whether text in the object is wrapped or appears all on one line. The term white space refers to the spaces between words, where text can be wrapped.

When set to Normal, the default, text will wrap when necessary, such as when space is limited.

When set to No Wrap, the text will never wrap to the next line in all report outputs, even if space is limited.
Width

Specifies the width of the visualization, in pixels.

Applies to

“Visualization” on page 842

Width (px)

Specifies the width of the note, in pixels.

Applies to

“Note” on page 819

Win Color

Specifies a color, color gradient, or pattern to apply to the win values.

Applies to

“Win-Loss Chart” on page 842

Win-Loss Threshold

Specifies the win-loss value in a win-loss chart. It represents values that are ties, which are mapped on the zero line.

Applies to

“Win-Loss Chart” on page 842

X Axis

Specifies whether the axis is rendered.

Applies to

“3-D Combination Chart” on page 772
**X Axis**
Specifies whether the axis is rendered.

**Applies to**

“Bubble Chart” on page 776, “Scatter Chart” on page 833

**X-Axis Data Item Value**
Specifies the scatter marker position on the X-axis.

**Applies to**

“Scatter Marker” on page 834

**Y1 Axis**
Specifies whether the axis is rendered.

**Applies to**

“Combination Chart” on page 785, “Metrics Range Chart” on page 818

**Y2 Axis**
Specifies whether the axis is rendered.

**Applies to**

“Combination Chart” on page 785

**Y2 Axis Position**
Specifies how the second Y axis is rendered. When Y2 Axis Position is set to Dual, the Y2 axis appears across from the Y1 numeric axis. When Y2 Axis Position is set to Bipolar, the Y2 axis appears below the Y1 axis. For example, in a combination chart showing Revenue and Quantity sold by Retailer type, with the Y2 Axis Position set to Dual, the Revenue columns and the Quantity sold line overlap because the Revenue axis (Y1) and the Quantity sold axis (Y2) are across from each other. However, with the Y2 Axis Position set to Bipolar, the Revenue columns appear above the Quantity sold line, and the data does not overlap. This property applies only to legacy charts.

**Applies to**

“Combination Chart” on page 785

**Y Axis**
Specifies whether the axis is rendered.

**Applies to**

“3-D Combination Chart” on page 772
Applies to

“Bubble Chart” on page 776, “Marimekko Chart” on page 815, “Scatter Chart” on page 833

Y-Axis Data Item Value
Specifies the scatter marker position on the Y-axis.

Applies to

“Scatter Marker” on page 834

Data Formatting Properties

The following is a list of properties available in the data formatting dialog.

"Not Applicable" Characters
Specifies the characters to be displayed when the value to be formatted was not applicable. The default value is two dashes (--). Note that the format will be applied only if the data source supports this error condition.

Any Error Characters
Specifies the characters to be displayed when the value to be formatted was not available because of an error. This property is overridden by the more specific formatting error conditions, such as Security Error Characters. The default value is two dashes (--). Note that the format will be applied only if the data source supports this error condition.

Calendar Type
Specifies the type of calendar to be displayed. The date values will be mapped to the selected calendar before being formatted. The default value is inherited from the user's content language. Note that the Japanese Imperial setting is only applicable for Japanese languages.

Clock
Specifies whether to display the time in 12-hour or 24-hour format. The default value is inherited from the user's content language.

Currency
Specifies the currency to be used. The default currency symbol will be displayed unless the values of the Currency Display and Currency Symbol properties are changed. The default value is inherited from the model.

Currency Display
Specifies whether to display the international or local currency symbol. By default, the local currency symbol is displayed.

Currency Symbol
Specifies a character or characters to use as the symbol to identify the local currency. This symbol will precede the number and any sign, even if it is a leading
A space between the symbol and the numeric value can be specified by entering it in this property, after the symbol. The default value is inherited from the user's content language.

**Currency Symbol Position**

Specifies where the currency symbol will appear. If End is selected, any spaces that follow the character or characters in the Currency Symbol or International Currency Symbol properties will be rendered between the number and the symbol. The default value is inherited from the user's content language.

**Date Ordering**

Specifies the order in which to display the day, month, and year. The default value is inherited from the user's content language.

**Date Separator**

Specifies the character to be displayed between the year, month, and day. The default value is inherited from the user's content language.

**Date Style**

Specifies the date style. The results rendered are determined by the language. Generally, Short uses only numbers, Medium uses some abbreviated words, Long uses complete words, and Full includes all available details.

**Decimal Separator**

Specifies the character that will separate non-decimal numbers from decimals. This property is ignored if no decimals are displayed. The default value is inherited from the user's content language.

**Digit Shaping**

Specifies the digit shaping option to apply. When National is selected, digit shapes are determined from the user's content language. When Contextual is selected, digit shapes are determined from adjoining characters in the value. For example, if the most recent strongly directional character before the numerical character is left-to-right, the number is displayed as a European number. If the most recent strongly directional character before the numerical character is right-to-left, the number is displayed in Arabic-Indic format. If there are no strongly directional characters before the numerical character, the number is displayed according to the base text direction of the field. When None is selected, no shaping is performed, and the value as it appears in the data source is shown. The default depends on the type of value. When the value is a string, the default is None.

**Digit Shaping**

Specifies the digit shaping option to apply. When National is selected, digit shapes are determined from the user's content language. When None is selected, no shaping is performed, and the value as it appears in the data source is shown. The default depends on the type of value. When the value is a number, the default is the ICU default for the language.

**Display AM / PM Symbols**

Specifies whether to display the AM or PM symbols. The default value is inherited from the user's content language.
**Display As Exponent**
Specifies whether to render values in scientific notation, using exponents. If this property is set to No, scientific notation will not be used. If this property is not specified, scientific notation will be used only when values exceed the maximum number of digits. The default value is inherited from the user's content language.

**Display Days**
Specifies whether to display the day. The format of the day can be controlled by selecting one of the specific formats. Selecting Julian means that the 3-digit day of the year will be displayed. The default value is inherited from the user's content language.

**Display Eras**
Specifies whether to display the era. The default value is inherited from the user's content language.

**Display Hours**
Specifies whether to display the hours. The default value is inherited from the user's content language.

**Display Milliseconds**
Specifies whether to display the milliseconds. The format of the milliseconds can be controlled by selecting one of the specific formats. This property is ignored if seconds are not displayed. The default value is inherited from the user's content language.

**Display Minutes**
Specifies whether to display the minutes. The format of the minutes can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

**Display Months**
Specifies whether to display the month. The format of the month can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

**Display Seconds**
Specifies whether to display the seconds. The format of the seconds can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

**Display Time Zone**
Specifies whether to display the time zone. The default value is inherited from the user's content language.
Display Weekdays
Specifies whether to display the weekday. The format of the weekday can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

Display Years
Specifies whether to display the year. The first two digits of the year, which indicate the century, can be controlled by selecting one of the associated property values. The default value is inherited from the user's content language.

Display Years
Specifies whether to display the year.

Divide By Zero Characters
Specifies the characters to be displayed when a numeric value is the result of a division by zero. The default value is /0. Note that the format will be applied only if the data source supports this error condition.

Exponent Symbol
Specifies the character to be displayed to identify exponents if the scientific notation is used. The symbol will be rendered after the number, separated by a space. The default value is inherited from the user's content language.

Group Size (digits)
Specifies the primary grouping size. If a value is specified it represents the number of digits to the left of the decimal point to be grouped together and separated by the thousands separator. The default value is inherited from the user's content language.

International Currency Symbol
Specifies a character or characters to use as a symbol to identify the international currency. This symbol will replace the currency symbol. A space between the symbol and the numeric value can be specified by entering it in this property, after the symbol. The default value is inherited from the user's content language.

Mantissa (digits)
Specifies the number of digits to be displayed following the exponent symbol if the scientific notation is used.

Maximum No. of Digits
Specifies the maximum number of digits that can be displayed. If the maximum number of digits is not sufficient to display the value, a scientific notation will be used. The default value is inherited from the user's content language.

Minimum No. of Digits
Specifies the minimum number of digits that can be displayed. If the minimum number of digits is too high to display a value, the padding character will be used. The default value is inherited from the user's content language.
**Missing Value Characters**
Specifies the character or characters to be displayed when the value is missing. If no value is entered for this property, an empty string will be displayed.

**Negative Pattern**
Specifies a presentation format, based on patterns, for negative numbers. Some restrictions exist. The numerical part of the negative pattern is ignored. Only the suffix and the prefix are used. For example, in the pattern ABC#,##0.#EFG, ABC is the prefix, EFG is the suffix and #,##0.# is the numerical part of the pattern.

**Negative Sign Position**
Specifies where the negative sign will appear. The default value is inherited from the user's content language.

**Negative Sign Symbol**
Specifies how to display negative numbers. The default value is inherited from the user's content language.

**No. of Decimal Places**
Specifies the number of digits to be displayed to the right of the decimal point. If this property is not set, the number of decimal places will vary depending on the number rendered.

**Numeric Overflow Characters**
Specifies the characters to be displayed when a numeric value is the result of a numeric overflow. The default value is two dashes (--) . Note that the format will be applied only if the data source supports this error condition.

**Padding Character**
Specifies the character that will be used to pad values that have fewer digits than the minimum number of digits. The default value is inherited from the user's content language.

**Pattern**
Specifies a presentation format that is based on patterns. The pattern format overrides formats specified in other properties. For example, to format the date as 2009/12/31 23:59:59 PM, use the pattern yyyy/MM/dd hh:mm:ss aa. For example, to format thousands using the letter K, set the Format Type to Number, set the Scale to -3 (to remove 000), and then use the pattern to ####K.

**Percentage Symbol**
Specifies whether to display the values per hundred (percent) or per thousand. The symbol will be appended to the number and any trailing sign. A space between the numeric value and the symbol can be specified by entering it in this property, after the symbol. The default value is inherited from the user's content language.

**Percent Scale (integer)**
Scale to be applied to value after formatting. If omitted, no percent scale will be applied and the value will be formatted according to the normal decimal positioning associated with the percent (or per mille) symbol.
Scale
Specifies how many digits to move the decimal delimiter for formatting purposes. For example, move the decimal three spaces to present values in thousands. The default value is inherited from the database field.

Secondary Group Size (digits)
Specifies the secondary grouping size. If a value is specified it represents the number of digits to the left of the primary group that will be grouped together and separated by the thousands separator. If this property is left blank, the secondary grouping of digits is the same number as the primary group size, as specified by the Group Size (digits) property. The default value is inherited from the user's content language.

Security Error Characters
Specifies the characters to be displayed when the value to be formatted was not available for security reasons. The default value is #!Security. Note that the format will be applied only if the data source supports this error condition.

Thousands Separator
Specifies how to delimit digit groups, such as thousands. This property is only used if the Use Thousands Separator property is set to Yes. The default value is inherited from the user's content language.

Time Separator
Specifies the character to be displayed between the hour, minute, and second. The default value is inherited from the user's content language.

Time Style
Specifies the time style to be displayed. The exact results that will be rendered are determined by the language. Generally, Short means that the minimum details will be displayed, Long adds seconds, and Full means that all details are displayed, including the time zone. The default value is inherited from the user's content language.

Time Unit
Specifies the unit of measure of the value. This property will be ignored if any day or time components are shown. The default value is inherited from the user's content language.

Use Thousands Separator
Specifies whether the grouping delimiter will be applied as defined by the Group Size property. The default value is inherited from the user's content language.

Zero Value Characters
Specifies the character or characters to be displayed when the value is zero (0). If no value is entered for this property, the Maximum No. of Digits property determines how many zero digits are displayed.
Appendix H. Prompt API for IBM Cognos BI

The JavaScript Prompt API provides report authors with a method of customizing prompt interaction in the reports they author.

With the prompt API, JavaScript applications can interact with IBM Cognos Business Intelligence report prompts for the purposes of validation or custom interaction. The prompt API can query and set user selections, validate typed-in values using patterns, set default values using expressions or query data, and more. Examples are provided to show various usage scenarios of the prompt API.

Related concepts:
“Prompt API samples” on page 528

The following reports are some of the reports found in the Samples_Prompt_API folder.

cognos.Prompt object

Defines the class that contains a prompt object from a report.

This object should only be accessed through an instance of cognos.Report.

cognos.Prompt.getControlByName method

Returns the control object associated with a name.

Syntax
(cognos.Prompt.Control) getControlByName(String sName)

Parameters
{String} sName
   Specifies the prompt name property set for the control in IBM Cognos Report Studio.

Returns
{cognos.Prompt.Control}
   Object associated to the specified sName control. Returns null if the control does not exist.

Example

The following is an example of the clearMyPrompt() function.
<script type="text/javascript">
function clearMyPrompt() {
   var oCR = cognos.Report.getReport("_THIS_"),
   var myPrompt = oCR.prompt.getControlByName("myPrompt");
   myPrompt.clearValues();
}
</script>

cognos.Prompt.getControls method

Returns an array of all controls associated with the report.
**Syntax**

(cognos.Prompt.Control[]) getControls()

**Parameters**

None.

**Returns**

{cognos.Prompt.Control[]}

An array of controls associated with the report.

**Example**

This example implements a function, `clearAllValues()`, that resets (clears the selection of) all prompt controls for a page.

```javascript
function clearAllValues() {
    var oCR = cognos.Report.getReport("_THIS_*");
    var aControls = oCR.prompt.getControls();
    for (var i = 0; i < aControls.length; i++) {
        aControls[i].clearValues();
    }
}
```

---

cognos.Prompt.Control object

A control is a visual element in the prompt page, such as a text prompt or a value prompt.

**cognos.Prompt.Control.addValues method**

Add an array of values to a control.

If the control does not support multiple values, only the first one provided is used.

**Syntax**

{void} addValues(aValues)

**Parameters**

{cognos.Value[]} aValues

The new values to use for this control.

**Returns**

{void}

**Example**

This example adds an array of two values (use and display) to a prompt.

```javascript
var oCR = cognos.Report.getReport("_THIS_*");
function setPromptValue( promptName, useValue, displayValue ) {
    var oP = oCR.prompt.getControlByName( promptName );
    var oValue = { 'use': useValue, 'display': displayValue };
cognos.Prompt.Control.clearValues method
Clear (or deselect) all values for this control.

Syntax
For a text prompt, the text is set to empty. For a value prompt, all the selections are removed.
{void} clearValues()

Parameters
None.

Returns
{void}

Example
This example resets all the controls of a page.
function clearAllValues() {
  var oCR = cognos.Report.getReport("_THIS_");
  var aControls = oCR.prompt.getControls();
  for (var i = 0; i < aControls.length; i++) {
    aControls[i].clearValues();
  }
}

cognos.Prompt.Control.getName method
Returns the name of the control.

The name is set in IBM Cognos Report Studio in the name property set for the control.

Syntax
{String} getName()

Parameters
None.

Returns
{String}
The value specified by the name property set for the control.

cognos.Prompt.Control.getValues method
Returns the current value for the control.

This method always returns an array, even if there is only one value selected.
• The following is an example for a text box prompt (single):
• The following is an example for a select prompt (multiple):
  [ {use: "CAN", display: "Canada"}, {use: "JPN", display: "Japan"} ]

• The following are examples of range prompts:
  A range is an array of two elements, with the mandatory names start and end. 
The value of each element is a regular array as in earlier examples. When the 
range values are equal, this method returns a normal value instead of a range:
  [ {use: useValue, display:displayValue} ]

  When the range values are different, this method returns a range:
  [ start:{use: useValue, display:displayValue}, end:{use: useValue, display:displayValue} ]

Syntax
{cognos.Value[]} getValues(boolean v_allOptions)

Parameters
{Boolean} v_allOptions

This optional parameter is applicable only to value prompts. The parameter 
specifies whether to retrieve all values or only selected values.

If the parameter is true, then all options are returned. If the parameter is false 
or missing, then only the selected options are returned.

The default value of this parameter is false.

Returns
{cognos.Value[]}

An array of values for the control.

Example
This example demonstrates how to adjust your code based on whether the result is 
a single value or a range.

function isRangeOutOfLimits(datePrompt, rangeLimit) {
  var result = false;
  var v = datePrompt.getValues();
  var rangeValue = v[0];
  var rangeDaysDiff = 0;
  if (rangeValue.start) {
    rangeDaysDiff = 0;
    var startDate = rangeValue.start.use;
    var endDate = rangeValue.end.use;
    rangeDaysDiff = dateUtils.subtractDate(endDate,startDate);
  }
  if (rangeDaysDiff > 0 && rangeDaysDiff <= rangeLimit) {
    result = true;
  }
  return result;
}

This example demonstrates the use of the parameter v_allOptions.

var allValues = valueControl.getValues(true);
var selectedValues = valueControl.getValues();

_cognos.Prompt.Control.setValidator_ method
Changes the default validation function for a control to one defined by the user.
When the specified function returns false, the UI element associated with the control indicates that a validation error occurred. When used in a multi-select control, the **Insert** button is disabled.

**Syntax**

```javascript
{void} setValidator(oFct)
```

**Parameters**

```javascript
{function} oFct
    A user-defined function that takes the user input as a parameter and returns a Boolean value.
```

**Returns**

```javascript
{void}
```

**Example**

This example demonstrates how to ensure that a valid postal code is provided in the form A1A 1A1.

```javascript
textBox.setValidator(
    function (values) {
      var result = false;
      if (values && values.length > 0) {
        var sValue = values[0]['use'];
        var rePostalCodeFormat = new RegExp( "[a-z][0-9][a-z] ?[0-9][a-z][0-9-]", "gi" );
        if ( rePostalCodeFormat.test(sValue) ) {
          result=true;
        }
      }
      return result;
    }
);
```

**cognos.Prompt.Control.setValues method**

Resets the control and adds an array of values to a control.

If the control doesn't support multiple values, only the first value provided is used. This is a convenience method that issues consecutive calls to `clearValues()` and `addValues()`.

**Syntax**

```javascript
{cognos.Value[]} aValues
```

**Parameters**

```javascript
{cognos.Value[]} aValues
    New values to use for this control.
```

**Returns**

```javascript
{void}
```

**Example**

```javascript
function setPromptValue( promptName, value ) {
  var oCR = cognos.Report.getReport("_THIS_");
  var oP = oCR.prompt.getControlByName(promptName);
};
```
oP.setValues( oValues );
}
oValues = [['use': 'ca', 'display': 'Canada'],
['use': 'us', 'display': 'USA']];
setPromptValue( 'countries', oValues );

cognos.Report object

Represents a report object in the content store.

Properties

prompt

References a cognos.Prompt object for this instance.

Example

This example creates a new cognos.Report object from the current instance.

```javascript
var ocr= cognos.Report.getReport("_THIS_" );
```

The new ocr object is specific to the cognos.Report namespace and avoids collisions with other objects in your scripts. The keyword _THIS_ should always be used. The Report Server will replace it with a unique namespace during report execution.

cognos.Report.getReport method


Namespaces are used to support HTML fragments. Always use a proper namespace when using this method to retrieve a cognos.Report object.

Syntax

```
```

Parameters

{String} sNamespace Optional

Returns

{cognos.Report}

The cognos.Report instance for this namespace.

Example

```javascript
var oCR = cognos.Report.getReport( "_THIS_" );
```

cognos.Report.sendRequest method

Sends a requests with the current parameters and values.

Syntax

```
{void} sendRequest(eAction)
```
Parameters
{cognos.Report.Action} eAction
   Specifies one of the cognos.Report.Action enumeration values. No other values are allowed.

Returns
{void}

Example
function customButton( eAction )
{
   var oCR = cognos.Report.getReport("_THIS_";
   oCR.sendRequest( eAction );
}

cognos.Report.Action object
   Defines constants for report navigation.

   Go back one prompt page.

   Syntax
   <static> <constant> cognos.Report.Action.BACK

   Cancel report execution and navigate to the previous page.

   Syntax
   <static> <constant> cognos.Report.Action.CANCEL

cognos.Report.Action.FINISH
   Submit parameter values and skip all remaining optional prompts and prompt pages.

   Syntax
   <static> <constant> cognos.Report.Action.FINISH

cognos.Report.Action.NEXT
   Submit parameter values and go to the next prompt page.

   Syntax
   <static> <constant> cognos.Report.Action.NEXT

   If the report contains prompt pages, the first prompt page is displayed. Otherwise, if the report doesn't contain prompt pages, this constant will re-prompt for values.

   Syntax
   <static> <constant> cognos.Report.Action.REPROMPT
**cognos.Value class**

JSON structure to represent parameter values.

This object includes the following valid attributes:
- `display`
- `end`
- `start`
- `use`

**Simple values**

```
{ 'use': '[a].[b].[c]', 'display': 'Canada' }
```

**Range values**

```
{  
  'start': { 'use': '2007-01-01', 'display': 'January 1, 2007'  
  }  
  'end': { 'use': '2007-12-31', 'display': 'December 31, 2007'  
}
```

**Multiple values**

```
[  
  { 'use': '12', 'display': 'Canada' },  
  { 'use': '41', 'display': 'Germany' },  
  { 'use': '76', 'display': 'Japan' }  
]```
Notices

This information was developed for products and services offered worldwide.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service. This document may describe products, services, or features that are not included in the Program or license entitlement that you have purchased.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group
Attention: Licensing
3755 Riverside Dr
Ottawa, ON K1V 1B7
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.
IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and [trademark information]" at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Glossary

This glossary includes terms and definitions for IBM Cognos Business Intelligence and IBM Cognos Framework Manager.

The following cross-references are used in this glossary:
- See refers you from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- See also refers you to a related or contrasting term.

To view glossaries for other IBM products, go to [www.ibm.com/software/globalization/terminology](http://www.ibm.com/software/globalization/terminology) (opens in new window).

**A**

accessible permission
A privilege that permits the access or use of an object.

accountability scorecard
A scorecard that Metric Studio automatically builds for each user which contains the metrics and projects they own.

agent
A process that performs an action on behalf of a user or other program without user intervention or on a regular schedule, and reports the results back to the user or program.

alias
An alternative name used instead of a primary name.

anonymous access
A type of access that allows users and servers to access a server without first authenticating with it.

application tier component
For installation, the set of processors that access the query databases to gather information and then render the results as PDF and HTML reports and metrics. Application tier components also pass requests to Content Manager and render the results that Content Manager retrieves from the content store.

attribute
1. In dimensional models, a property that provides qualitative information about members of a level in a dimension. For example, the Store level within the Retailer dimension might have properties such as address or retail space. In general, dimensional attributes do not have measure values or rollups associated with them, but are used to locate or filter members.
2. In relational models, a query item that is not a measure or identifier. When a query item is an attribute, it is not intended to be aggregated, or used for grouping or generating prompt pick lists.
3. In BI Modeling, a characteristic of an entity which is descriptive rather than a unique identifier or an aggregative measure.

authentication
The process of validating the identity of a user or server.

authentication provider
The communication mechanism to an external authentication source. Functionalities, such as user authentication, group membership, and namespace searches, are made available through authentication providers.

**B**

burst
To create several report results by running a single report once. For example, the user can create a report that shows sales for each employee, and run it once, sending different results to regional managers by bursting on region.

burst key
The dimension or level of a query in the
report specification that is used to create, or burst, a set of report results.

CA
See certificate authority.

calculated member
A member of a dimension whose measure values are not stored but are calculated at run time using an expression.

canvas
An area within a dashboard or workspace that users interact with to create, view, and manipulate content and data.

capability
A group of functions and features that can be hidden or revealed to simplify the user interface. Capabilities can be enabled or disabled by changing preference settings, or they can be controlled through an administration interface.

cardinality
1. For relational data sources, a numerical indication of the relationship between two query subjects, query items, or other model objects.
2. For OLAP data sources, the number of members in a hierarchy. The cardinality property for a hierarchy is used to assign solve orders to expressions.

cascading prompt
A prompt that uses values from a previous prompt to filter the values in the current prompt or pick list.

certificate
In computer security, a digital document that binds a public key to the identity of the certificate owner, thereby enabling the certificate owner to be authenticated. A certificate is issued by a certificate authority and is digitally signed by that authority. See also certificate authority.

certificate authority (CA)
A component that issues certificates to each computer on which components are installed.

CGI
See Common Gateway Interface.

cipher suite
The combination of authentication, key exchange algorithm, and the Secure Sockets Layer (SSL) cipher specification used for the secure exchange of data.

class style
A combination of formatting characteristics, such as font, font size, and border, that the user names and stores as a set.

CM
See Content Manager.

Common Gateway Interface (CGI)
An Internet standard for defining scripts that pass information from a web server to an application program, through an HTTP request, and vice versa.

condition
An expression that can be evaluated as true, false, or unknown. It can be expressed in natural language text, in mathematically formal notation, or in a machine-readable language.

conformed dimension
A dimension with a single definition that can be reused or shared across multiple coordinated data marts.

constraint
1. A security specification that denies one or more users the ability to access a model component or to perform a modeling or authoring task.
2. A restriction on the possible values that users can enter in a field.

contact
A named email address to which reports and agent e-mails can be sent. Contacts are never authenticated.

content locale
A code that is used to set the language or dialect used for browsers and report text, and the regional preferences, such as formats for time, date, money, money expressions, and time of day.

Content Manager (CM)
The service that retrieves information from the content store, and saves information to the content store.

content store
The database that contains the data needed to operate, such as report specifications, published models, and security rights.
credential  
A set of information that grants a user or process certain access rights.

cube  
A multidimensional representation of data needed for online analytical processing, multidimensional reporting, or multidimensional planning applications.

custom set  
In Analysis Studio, a named object which can include filter rules, calculations, and sort rules. Custom sets can define a set of members that is different from any set originally defined in the cube model. See also predefined set.

dashboard  
A web page that can contain one or more widgets that graphically represent business data.

data source  
The source of data itself, such as a database or XML file, and the connection information necessary for accessing the data.

data source connection  
The named information that defines the type of data source, its physical location, and any sign-on requirements. A data source can have more than one connection.

data tree  
Within a studio, a structure that contains objects such as query subjects, query items, dimensions, levels, and members. A data tree is used as a palette of the available data that can be inserted into calculations, filters, display areas, and other authoring gestures.

deployment  
The process of moving an application (such as a report or model) to a different instance. For example, reports are often created in a test environment and then deployed to production. When an application is deployed, it is exported, transferred, and imported.

deployment archive  
A file used for deployment. A deployment archive contains the data from the content store that is being moved.

deployment specification  
A definition of what packages to move (deploy) between source and target environments, the deployment preferences, and the archive name. Deployment specifications are used for import and export.

derived index  
A calculated metric that provides a status and a score based on other metrics.

details-based set  
A set based on an item and its immediate details. See also set.

dimension  
A broad grouping of descriptive data about a major aspect of a business, such as products, dates, or locations. Each dimension includes different levels of members in one or more hierarchies and an optional set of calculated members or special categories.

dimensional data source  
A data source containing data modeled using OLAP concepts, including dimensions, hierarchies, and measures.

drill down  
In a multidimensional representation of data, to access information by starting with a general category and moving downwards through the hierarchy of information, for example from Years to Quarters to Months.

E  

event  
A change to a state, such as the completion or failure of an operation, business process, or human task, that can trigger a subsequent action, such as persisting the event data to a data repository or invoking another business process.

event key  
A combination of data items that uniquely defines an event instance. Identifying an event instance enables the agent to determine if it is new, ongoing or stopped.

event list  
The set of detected event instances.
evaluated by the task execution rules to determine which agent tasks should be performed.

**F**

**fact**  See measure.

**G**

gateway
An extension of a web server program that transfers information from the web server to another server. Gateways are often CGI programs, but may follow other standards such as ISAPI and Apache modules.

glyph  The actual shape (bit pattern, outline) of a character image. For example, italic A and roman A are two different glyphs representing the same underlying character. Strictly speaking, any two images which differ in shape constitute different glyphs. In this usage, glyph is a synonym for character image, or simply image (The Unicode Standard – Version 1.0).

governor
A set of rules to limit user activities, such as the execution of reports, that either take too long or consume too many resources.

group  A collection of users who can share access authorities for protected resources.

**grouping**  In reporting, the process of organizing common values of query items together and only displaying the value once.

**H**

**hierarchy**
The organization of a set of entities into a tree structure, with each entity (except the root) having one or more parent entities and an arbitrary number of child entities.

**I**

**information card**
A display of high-level information about dashboard, workspace, or report content, such as owner, contact information, date modified, and an optional thumbnail view of the dashboard, workspace, or report.

**information pane**
In Analysis Studio, a pane that helps the user to confirm their selection in the data tree by displaying related information, such as the level and attributes.

**initiative**
A task developed to achieve objectives or close the gap between performance and targets. Initiatives are associated with individual objectives and often known as projects, actions, or activities.

**item**  See member.

**J**

**job**
A group of runnable objects, such as reports, agents, and other jobs that the user runs and schedules as a batch.

**job step**
The smallest part of a job that can be run separately. A job step can be a report or it can be another job.

**L**

**layout**
The arrangement of printed matter on a screen or page, including margins, line spacing, type specification, header and footer information, indents, and more.

**level**
A set of entities or members that form one section of a hierarchy in a dimension and represent the same type of object. For example, a geographical dimension might contain levels for region, state, and city.

**locale**
A setting that identifies language or geography and determines formatting conventions such as collation, case conversion, character classification, the language of messages, date and time representation, and numeric representation.

**M**

**MDX**  See Multidimensional Expression Language.

**measure**
A performance indicator that is quantifiable and used to determine how well a business is operating. For example,
measures can be Revenue, Revenue/Employee, and Profit Margin percent.

**member**
A unique item within a hierarchy. For example, Camping Equipment and 4 Man tent are members of the Products hierarchy.

**metric**
A measure to assess performance in a key area of a business.

**metric extract**
A set of mappings between an existing Cognos data source and a Metric Studio object or value. For example, a cube measure named Revenue is mapped to a Metric Studio metric named Revenue Actual Value.

**metric package**
In Cognos Connection, a representation of a Metric Studio application. A metric package contains connection information, reports, and metric management tasks for that application. See also **package**

**metric store**
A database that contains content for metric packages. A metric store also contains Metric Studio settings, such as user preferences.

**metric type**
A category of metrics that defines the business rules such as performance pattern, units, and meaning of a group of metrics. For example, Revenue can be a metric type, and European Revenue and North American Revenue would be metrics of this type.

**model**
A physical or business representation of the structure of the data from one or more data sources. A model describes data objects, structure, and grouping, as well as relationships and security. In Cognos BI, a model is created and maintained in Framework Manager. The model or a subset of the model must be published to the Cognos server as a package for users to create and run reports.

**model segment**
A part of a Framework Manager project, such as a parameter map, a data source, a namespace, or a folder, that is a shortcut to a second project. Segments are used to simplify model maintenance or to facilitate multi-user modeling.

**multidimensional data source**
See **dimensional data source**

**Multidimensional Expression Language (MDX)**
The multidimensional equivalent of Structured Query Language (SQL).

**N**

**named set**
See **predefined set**

**namespace**
A part of the model in which the names may be defined and used. Within a namespace, each name has a unique meaning.

**news item**
A single entry in a Really Simple Syndication (RSS) compatible format. It can include a headline, text, and a link to more information. A news item task in an agent can be used to create news items for display in a Cognos Connection portlet.

**normalization**
The process of restructuring a data model by reducing its relations to their simplest forms. It is a key step in the task of building a logical relational database design. Normalization helps avoid redundancies and inconsistencies in data. An entity is normalized if it meets a set of constraints for a particular normal form (first normal form, second normal form, and so on).

**O**

**object**
In Report Studio, an empty information container that can be dragged to a report from the Toolbox tab and then filled with data. Reports are made up of objects, which include crosstabs, text items, calculations, graphics, and tables.

**object extract**
An extract that defines the metadata for a Metric Studio object, such as a user defined column, a scorecard, or a data source.
package
A subset of a model, which can be the whole model, to be made available to the Cognos server. See also metric package.

page set
In Report Studio, a set of one or more designed pages which repeat in the report output for each instance of a chosen query item. See also set.

passport
Session-based information, stored and encrypted in Content Manager memory, regarding authenticated users. A passport is created the first time a user accesses Cognos 8, and it is retained until a session ends, either when the user logs off or after a specified period of inactivity.

portlet
A reusable component that is part of a web application that provides specific information or services to be presented in the context of a portal.

predefined set
A set of members defined inside an OLAP data source as a list or by an expression. Predefined sets can be used in analysis and report authoring. See also custom set.

product locale
The code or setting that specifies which language, regional settings, or both to use for parts of the product interface, such as menu commands.

project
1. In Metric Studio, a task or set of tasks undertaken by a team and monitored on a scorecard. A project tracks dates, resources, and status.
2. In Framework Manager, a set of models, packages, and related information for administration, and for sharing model information.
3. In Metric Designer, a group of extracts. Each extract contains the metadata that is used to populate the Metric Studio data store or to create applications.

prompt
A report element that asks for parameter values before the report is run.

properties pane
Within a studio, a pane that provides an overview of the properties for selected data. The properties pane can also be used to make several changes and apply them at the same time, instead of repeating several different commands.

publish
In Cognos BI, to expose all or part of a Framework Manager model or Transformer PowerCube, through a package, to the Cognos server, so that the data can be used to create reports and other content.

Q
query
The simple report specifications created and edited by Query Studio.

query item
A representation of a column of data in a data source. Query items may appear in a model or in a report and contain a reference to a database column, a reference to another query item, or a calculation.

query subject
A named collection of query items that are closely functionally related. Query subjects are defined using Framework Manager to represent relational data and form the set of available data for authoring reports in Query Studio and Report Studio. A query subject is similar to a relational view in that it can be treated as a table but does not necessarily reflect the data storage.

R
Really Simple Syndication (RSS)
An XML file format for syndicated web content that is based on the Really Simple Syndication specification (RSS 2.0). The RSS XML file formats are used by Internet users to subscribe to websites that have provided RSS feeds. See also Rich Site Summary.

repeater
In Report Studio, a cell container that repeats values within itself with no predefined internal structure.
repeater table
In Report Studio, a table-like container that repeats cells across and down the page or row in the associated query.

report
A set of data deliberately laid out to communicate business information.

report output
The output produced as a result of executing a report specification against a data set.

report specification
An executable definition of a report, including query and layout rules, which can be combined with data to produce a report output.

report view
A reference to another report that has its own properties, such as prompt values, schedules, and results. Report views can be used to share a report specification instead of making copies of it.

response file
An ASCII file that can be customized with the setup and configuration data that automates an installation. During an interactive installation, the setup and configuration data must be entered, but with a response file, the installation can proceed without any intervention.

Rich Site Summary (RSS)
An XML-based format for syndicated web content that is based on the RSS 0.91 specification. The RSS XML file formats are used by Internet users to subscribe to websites that have provided RSS feeds. See also Really Simple Syndication.

RSS
1. See Really Simple Syndication
2. See Rich Site Summary

S

score
A number or ranking that expresses applicability in relation to a standard.

scorecard
A collection of metrics representing the performance of one unit or aspect of an organization.

scorecard structure
The hierarchy of scorecards that reflects how an enterprise organizes its metrics.

security provider
See authentication provider.

selection-based set
A collection of individual items that the user has explicitly selected. The items or members may be selected from one or more levels of the same hierarchy. See also set.

session
The time during which an authenticated user is logged on.

set
A collection of related items or members. Members in a set may be specifically chosen, or selected by one or more filter rules. See also custom set, details-based set, page set, predefined set, selection-based set, stacked set.

stacked set
Two or more sets arranged one above another in rows or side-by-side in columns. See also set.

strategy
The overall plan of action (such as for a brand unit, business unit, channel, or company) to achieve a stated goal. Strategies normally cover a period of more than one year.

strategy map
In Metric Studio, a visual representation of the strategy and the objectives of that strategy for an organization. For example, a strategy map may show employees how their jobs are aligned to the overall objectives of the organization.

summary
In reporting and analysis, an aggregate value that is calculated for all the values of a particular level or dimension. Examples of summaries include total, minimum, maximum, average, and count.

T

task
An action performed by an agent if the event status meets the task execution rules. For example, an agent can send an email, publish a news item, or run a report.

task execution rule
A user-specified option within an agent that determines which statuses and values
cause a task to be run. It determines which tasks to execute for each event instance.

**template**
In report authoring, a reusable report layout or style that can be used to set the presentation of a query or report.

**thumbnail**
An icon-sized rendering of a larger graphic image that permits a user to preview the image without opening a view or graphical editor.

**tuple**
An ordered collection of two or more members from different dimensions. For example, the tuple (2007, Camping Equipment, Japan) returns the value for the intersection of the three members: 2007, Camping Equipment, and Japan. Tuples can be used to filter and sort data, and to create calculations.

**U**

**union set**
See stacked set

**user**
Any individual, organization, process, device, program, protocol, or system that uses the services of a computing system.

**user-defined column**
In metric management, a column used to represent a value other than the actual or target. It may be an industry benchmark or any other useful additional numerical information for a period, including a calculation based on the other values of the metric. User-defined columns may be different for each metric type.

**W**

**watch list**
A list of metrics that each user has chosen to monitor closely. If notification is enabled in Metric Studio, the user will receive email notification of changes to these metrics. Users can also choose to display their watch list as a portlet within Cognos Connection.

**watch rule**
A user-defined condition that determines whether a report is delivered to the user. When the rule is run, the output is evaluated and, if it satisfies the condition or rule, the report is delivered by email or news item. Watch rules limit report delivery to those reports containing data of significance to the user.

**Web Services for Remote Portlets**
A standard for creating presentation-oriented web services so that they can be easily integrated within other applications, such as web portals.

**widget**
A portable, reusable application or piece of dynamic content that can be placed into a web page, receive input, and communicate with an application or with another widget.

**work area**
The area within a studio that contains the report, analysis, query, or agent currently being used.

**workspace**
See dashboard
Index

Special characters
-- characters
errors when filtering 535
marking special cells 387
troubleshooting in reports 481
! characters 533
* characters 387
% calculations
in dimensional reporting 303
in relational reporting 250

Numerics
100% stacked charts 101
3-D charts 102

A
absolute charts 100
access code
setting for active report 173
accessibility features 467
keyboard shortcuts 467
accessible reports 468
adding alternate text 470
adding summary text 471
associating labels to prompt controls 472
conditional blocks 473
emphasizing text 471
headings 471
report output options 469
specifying cell headers in tables 471
active reports 171
adding controls 186
adding interactivity 184
controls 171
converting existing reports 172
creating and managing active report variables 184
defining connections between controls 192
drill-through links do not work 506
example 199, 203
in workspaces in IBM Cognos Workspace 4
integrating with IBM Cognos Workspace 184
running 199
saving in MHT format 199
setting access code 173
setting properties 173
specifying maximum number of rows that can appear 173
specifying reaction behavior of controls 195
specifying selection behavior of controls 194
summarizing data 198
unable to add to Cognos Workspace 498
unable to view in Firefox 498
unable to view in Firefox 8 or later 498
using master detail relationships when authoring decks 197
variables 172
adding data 41
aggregating
limitations with measures 534
mapping functions from data sources 228
member sets 272
samples and time dimensions 509
specifying aggregation properties in models 222
troubleshooting running totals 496
using a simple summary in dimensional reporting 268
using simple summaries in relational reporting 221
values in crosstabs and charts 269
aliases in maps 166
aligning objects 339
alternate text
accessibility 470
Analysis Studio
creating templates in Report Studio 451
annotations
cart chart baselines 128
notes in charts 127
area charts 88
arithmetic overflow errors 497
Auto Group & Summarize 221
automating IBM Cognos BI using the IBM Cognos Software Development Kit 37
averages
moving and rolling 272
axes 14
in charts 80
titles in maps 164
axis scales 122

B
backgrounds
adding background effects to charts 334
adding colors to objects 353
changing in current default charts 115
changing in legacy charts 115
colors not appearing in templates 478
inserting background images 334
troubleshooting in charts 479
bar charts 87
base text direction 56
baselines
adding to charts 128
bevels
adding to legacy charts 116
bidirectional support 56
digit shaping 389
enabling 45
reports 5
binary round-off errors 486
BlackBerry devices
creating mobile reports for 61
block objects 31
blocks
inserting 336
resizing 349
bookmarks
adding 442
creating for drill-through access 418

© Copyright IBM Corp. 2005, 2013
See burst reports

**bubble charts** 91
- specifying bubble size in current default charts 148

**bubble size** 15

**Build Prompt Page tool** 314

**customizing current default charts** 152

**burst reports** 379
- creating against cubes 384
- creating against dimensional data sources 384
- creating calculated fields to define recipients 379
- defining groups 381
- defining recipients 379
- denial of service errors in dimensional reporting 499
- denial of service errors in relational reporting 245
- enabling in IBM Cognos Connection 383
- Microsoft Excel format 553
- setting options 382
- troubleshooting 495

**business keys**
- using for drill-through 421

**button bars** 187

**calculations**
- calculations do not appear in target report 505
- troubleshooting, drilling through 505
- calculators do not appear in target report 505
- troubleshooting, drilling through 505
- CAMID,
  - See IBM Cognos Access Manager ID

**cascading prompts**
- adding 325
- cascading styles sheets 351
- cast_Date functions 351
- troubleshooting 493

**changed features**
- version 10.1.1 10
- version 10.2.0 7

**chart configurations**
- 100% stacked 101
- 3-D 102
- stacked 100
- standard 100

**chart properties**
- specifying 81

**chart types**
- area charts 88
- bar charts 87
- bubble charts 91
- bullet charts 93
- column charts 84
- combination charts 90
- gauge charts 94
- line charts 85
- Marimekko charts 97
- metrics range charts 99
- microcharts 96
- Pareto charts 95
- pie charts 86
- point charts 89
- polar charts 98
- progressive column charts 96
- quadrant charts 92
- radar charts 98
- scatter charts 90
- win-loss charts 96
- charts 12, 79

**charts**
- accessible reports 473
- adding alternate text 470
- aggregating values 269
- axes 14
- background effects 334
- background effects in legacy charts 116
- backgrounds in current default charts 115
- backgrounds in legacy charts 115
- bands of color in the background 123
- baselines 128
- borders 334
- bubble size 15
- bullet 13
- changing backgrounds in current default charts 115
- changing backgrounds in legacy charts 115
- changing colors 108
- changing lines 119
- changing scales 119
- color by value 15
- colored regions in current default charts 118
- configurations 83
- converting current default charts to a matrix 143
- converting types 103
- creating 79
- cumulation line in current default Pareto charts 151
- cumulation line in legacy Pareto charts 151
- current default 79
- current default in Report Studio 12
- customizing current default combination charts 141
- customizing labels 119
- customizing legend items in current default charts 125
- customizing legend items in legacy charts 126
- customizing palettes 108
- defining colors by values in current default charts 147
- defining query contexts 138

**changed features**
- version 10.1.1 10
- version 10.2.0 7

**chart configurations**
- 100% stacked 101
- 3-D 102
- stacked 100
- standard 100

**chart properties**
- specifying 81

**chart types**
- area charts 88
- bar charts 87
- bubble charts 91
- bullet charts 93
- column charts 84
- combination charts 90
- gauge charts 94
- line charts 85
- Marimekko charts 97
- metrics range charts 99
- microcharts 96
- Pareto charts 95
- pie charts 86
- point charts 89
- polar charts 98
- progressive column charts 96
- quadrant charts 92
- radar charts 98
- scatter charts 90
- win-loss charts 96
- charts 12, 79
charts (continued)
  displaying every second label 479
  drilling through 155
  drilling up and down 154
  drop shadows 334
  elements 80
  enhanced styles 14
  fills 334
  gridlines 123
  inserting microcharts into crosstabs 152
  legacy 79
  legacy and current default 462
  legacy in Report Studio 12
  limitations 501
  markers 131
  matrices 13
  metrics range 156
  notes 15, 127
  pie 14
  positioning legends in current default charts 124
  properties not supported in Microsoft Excel 553
  regression lines in current default charts 135
  regression lines in legacy charts 136
  resizing 349
  resizing current default 108
  resizing legacy 108
  resolving multiple calculations 539
  series colors in current default combination charts 141
  setting properties 104
  showing data labels 136
  showing data labels in current default charts 138
  showing data labels in current default pie and donut charts 137
  showing data points 140
  showing value markers 140
  specifying bubble size in current default bubble charts 148
  specifying default measures 74
  summarizing small slices or items in current default charts 139
  tooltips 104
  troubleshooting axis labels 478
  troubleshooting differences in appearance when run in different formats or on different operating systems 499
  troubleshooting gray gradient backgrounds 479
  types 83
  types not supported in Microsoft Excel 553
  checkbox groups 188
  classes
    changing the default for a report 351
  clipboards
    copying reports 453
    opening reports 454
  Cognos Business Insight
    changed name 7
  Cognos Business Insight Advanced
    changed name 7
  Cognos SQL
    in dimensional reporting 295
    in relational reporting 246
  Cognos Statistics
    object missing from a report 502
    removal in version 10.2.1 3
  Cognos Workspace
    creating reports for workspaces 57
    filters in workspaces 59
  color by value 15
  colored regions
    adding to current default charts 118
  colors
    adding to current default charts 147
    adding to objects 353
    changing in charts 108
    not appearing in templates 478
  column charts 84
  columns 41
    adding multiple items in a single column 342
    calculated in dimensional reporting 302
    calculated in relational reporting 248
    grouping in relational reporting 214
    hiding or showing in list controls 197
    setting the group span 216
    sorting in dimensional reporting 283
    sorting in relational reporting 235
    swapping with rows 74
    width limitations in Microsoft Excel 551
  combination charts 90
  concatenating strings 539
  Condition Explorer 26
  conditional block lists
    inserting 337
  conditional blocks
    accessible reports 473
  conditional palettes
    creating 111
  conditional rendering 370
    adding multiple layouts 375
    adding variables 370, 371
    example 373
    hiding and showing objects 372
  conditional styles
    creating 365
    using variables 369
  container direction 56
  context filters
    creating 279
    prompting with 281
  continuous values sliders 189
  control data items
    referencing 192
  controls
    active reports 171
    adding data 190
    adding to an active report 186
    button bars 187
    buttons 189
    checkbox groups 188
    continuous values sliders 189
    data button bars 187
    data deeks 186
    data checkbox groups 188
    data discrete values sliders 189
    data drop down lists 188
    data iterators 189
    data list boxes 188
    data radio button groups 188
    data repeater tables 186
    data tab controls 187
    data toggle button bars 187
    decks 186
    defining connections 192
    discrete values sliders 189
    drop down lists 188
    iterators 189

Index 997
controls (continued)
list boxes 188
radio button groups 188
referencing data items defined in static controls 192
repeater tables 186
row numbers 186
specifying reaction behavior 195
specifying selection behavior 194
tab 187
toggle button bars 187
variable text items 186
converting
lists to crosstabs 75
lists to repeaters 65
reports to templates 451
to MDX in dimensional reporting 297
to SQL in dimensional reporting 297
to SQL in relational reporting 247
copying
reports to the clipboard 453
copying reports 454
Count
changes between IBM Cognos ReportNet and IBM Cognos BI 465
Count Distinct 465
crosstab node members 67
crosstab nodes 67
crosstabs 67
aggregating values 269
allow sorting 197
applying table styles 340
creating single-edge 70
crosstab node members 67
crosstab nodes 67
default measures 74
defining independent drill-through definitions 427
discontinuous 76
formatting 69
indenting data 74
inherited table styles 5
intersections 69
nesting data 71
order of styles 69
overflow errors 493
resolving multiple calculations 539
showing values as percentages 72
CRX-API-0018 errors 537
CSV format
importing your own files 355
producing reports in 48
cubes
troubleshooting drilling through to relational data 503
cumulation lines
in current default charts 151
in legacy charts 151
currencies
data format limitations in Microsoft Excel output 552
formatting 387
formatting disappearing in SSAS 2005 495
current default charts 79, 462
custom groups
creating 215, 265
dashboards (continued)
using bullet charts 93
using gauge charts 94
using global filters 313
data
adding 41
data button bars 187
data checkbox groups 188
data containers
inserting from the Toolbox tab 41
data decks 186
data discrete values sliders 189
data does not appear in target report
troubleshooting drilling through 496
data drop down lists 188
data formats 387
date and time symbols 394
decimal format symbols 401
locale-sensitive 390
Microsoft Excel limitations 552
specifying default 387
specifying for objects 388
using patterns 393
data is not filtered in target report
troubleshooting drilling through 505
data items tab 25
data iterators 189
data list boxes 188
data packages
refreshing 40
specifying 39
data points
showing in charts 140
data radio button groups 188
data repeater tables 186
data series in charts 80
data sources 253
data stores
conformed dimensions 420
data tab controls 187
data toggle button bars 187
data trees
customizing 255
date & time prompts 316
date prompts 316
dates
filtering 234
using for drill-through access 423
decimals
rounding 388
decks 186
using master detail relationships 197
denial of service errors
in dimensional reporting 499
in relational reporting 245
depth
charts 104
derived columns 234
detail filters 229
diagrams
Metric Studio 158
dial charts 94
dictionary properties in maps 166
digit shaping 56
in charts and maps 389
dimensional 265
dimensional coercion rules 544
ences
dimensional data sources
  creating burst reports 384
  creating master detail relationships 295
  report limitations 531
  running reports against 531
  using with queries 285
dimensional reporting 253
  adding data 253
  best practices 22
  drilling up and down 307
  filtering data 278
  sorting data 281
  summarizing data 266
  using calculations 302
  working with queries 283
dimensionally-modeled relational data sources
  adding dimensional data 253
  limitations when aggregating measures 534
dimensions 41, 253
  conformed 285
  conformed for drill-through access 420
  non-conformed 285
  overlapping named set levels 479
  searching 256
discontinuous crosstabs
  creating 76
  discrete values sliders 189
  distributing reports, See burst reports
  dividing by zero
troubleshooting 477
  Don't Print class 352
  double counting 465
drilling down 307
  creating drill-up/drift-down reports 308
  in charts 154
  using member sets 307
drilling through 415
  business keys 421
  calculations do not appear in target report 505
  charts 155
  concepts 416
  creating drill-through reports 423
  cubes to relational data 503
  data does not appear in target report 496
  data is not filtered in target report 505
  defining different drill-through definitions for
crosstabs 427
IBM Cognos Visualizer reports 434
  in maps 169
  links not active in Safari Web browsers 496
  members and values 418
  Microsoft Excel limitations 552
  model-based access 416
  multiple values 425
  nested crosstab only filters on some items 505
  packages 418
  paths 416
  PowerCubes and packages 423
  PowerPlay reports 434
  problems with active reports 506
  relational data to cubes 504
  report formats 417
  report-based access 416
  selection contexts 417
  specifying drill-through text 428
drilling through (continued)
  troubleshooting PowerCubes 503
  using bookmarks 418
  using dates 423
  using member unique names 419
  using scope in model-based reporting 422
  using URLs 418
  with conformed dimensions 420
  wrong data appears in target report 496
drilling up 307
  creating drill-up/drift-down reports 308
  in charts 154
  using member sets 307
drop down lists 188
  defining no list value item 196
drop shadows
  adding to legacy chart elements 116
drop shadows in charts 334
Dynamic Query Migration validation option 43

E
  emailing reports,
  See burst reports
  empty cells
    specifying what appears for empty data containers 393
    suppressing 391
error messages
  CRX-API-0018 537
  denial of service in dimensional reporting 499
  denial of service in relational reporting 245
  HRESULT 488
  missing items 494
  MSR-PD-0001 359
  MSR-PD-0012 483
  MSR-PD-0013 483
  OP-ERR-0199 214, 478
  OP-ERR-0201 479
  OP-ERR-0209 535
  OP-ERR-0210 535
  OP-ERR-0212 535
  OP-ERR-0213 535
  ORA-00907 494
  overflow errors in crosstabs 493
  PCA-ERR-0057 497
  PCA-ERR-0087 499
  QE-DEF-0288 493
  QE-DEF-0478 544
  ReferenceError: HTMLIsIndexElement is not defined 498
  RPC-DEF-0177 497
  RSV-SRV-0025 488
  RSV-SRV-0040 477
  TypeError: _IS1 is undefined 498
  UDA-SQL-0114 488, 497
  UDA-SQL-0206 488
  UDA-SQL-0564 497
errors
  application errors upgrading a report 477
  filtering results in error characters 535
  null values in count summaries 480
  out of memory 310, 500
  parse errors with upgraded reports 493
  recursive evaluation 497
  troubleshooting error cells in reports 534
  escape characters
    in calculations in dimensional reporting 538
    in calculations in relational reporting 251
examples for relational metadata
prompts 408
Excel 2000 format
deprecation notices 21
Explorer Bar 26
expression editor
Aster Data 593
Aster Data Data Type Formatting 595
Aster Data Math 595
Aster Data String 593
Aster Data Trigonometry 596
Block Functions 592
browsing data 233
Business Date/Time Functions 587
Common Functions 713
Constants 584
Constructs 586
creating filters in dimensional reporting 278
creating filters in relational reporting 228
creating parameters to create prompts 319
Date Functions 667
DB2 598
DB2 Math 598
DB2 Trigonometry 598
dimensional coercion rules 544
Dimensional Functions 725
Greenplum 616
Greenplum Data type formatting 617
Greenplum Math 618
Greenplum String 616
Greenplum Trigonometry 618
Informix 621
Informix Math 621
Informix Trigonometry 621
Macro Functions 699
Member Summaries 581
MS Access 626
MS Access Cast 626
MS Access Math 627
MS Access Trigonometry 628
MySQL 634
MySQL Math 635
MySQL String 634
MySQL Trigonometry 636
Netezza 638
Netezza Fuzzy 639
Netezza Math 638
Netezza Phonetic 640
Netezza Trigonometry 638
Operators 555
Oracle 644
Oracle Math 644
Oracle Trigonometry 644
Paracel 653
Paracel Data type formatting 654
Paracel Math 654
Paracel String 653
Postgres 655
Postgres Data type formatting 657
Postgres Math 657
Postgres String 655
Postgres Trigonometry 658
Red Brick 660
Report Functions 751
Salesforce.com 667
SAP BW 666
SAP BW Math 667
expressions
length 537
using summary functions in relational reporting 224
extended data items 264
enabling for new reports 350
report option 34
extensible visualizations 174
external data
MSR-PD-0012 error when importing 483
MSR-PD-0013 error when importing 483
supported data sources 10

F
facts 41
field sets
inserting 336
fills
adding to charts 334
adding to legacy charts 116
filtering
-- error characters 535
creating detail filters 229
creating global prompts in dashboards 313
creating summary filters 229
data from SAP BW data sources 536
data in dimensional reporting 278
data in relational reporting 228
date columns 234
drilling through from IBM Cognos Series 7 to IBM Cognos BI 435
limitations when specifying the scope 535
limitations with dimensional data sources 534
members in a set 263
multiple-fact queries 285
removing or editing 233
removing or editing detail filters 233
removing or editing summary filters 233
suppressing null values 391
to top or bottom values 262
troubleshooting _make_timestamp columns 488
using context filters 279
using slicers 279
workspaces in Cognos Workspace 59
layouts (continued)
using the page structure view 329
legacy charts 79, 462
legends 13
adding titles in maps 168
hiding or showing in maps 164
in charts 80
in maps 167
positioning in current default charts 124
level hierarchies 253
levels 41, 253
member unique names 419
limitations
aggregating measures in dimensionally-modeled relational or relational data sources 534
line breaks 348
line charts 85
line spacing 348
lineage information 54
linear scales
showing 119
lines
adding regression lines to charts 135
adding regression lines to legacy charts 136
changing in charts 119
linking data items for drilling 308
list boxes 188
list reports
hiding columns 64
lists 63
adding headers and footers 331
allow sorting 197
applying table styles 340
converting to crosstabs 75
converting to repeaters 65
formatting 64
grouping data 214
hiding or showing columns 197
inherited table styles 5
limitations 532
single-edge crosstabs 70
subtotals in grouped lists 478
literal strings
in calculations in dimensional reporting 538
in calculations in relational reporting 251
locales
locale-sensitive properties 390
location-aware reports
creating 61
locking objects 31
logarithmic scales 122
showing 119
maps (continued)
mapping data values 166
Microsoft Excel limitations 552
notes 168
parts of a map report 162
point layers 167
region layers 165
resizing 349
showing or hiding properties 165
updated 4, 9
using from previous versions 9
margins
setting for objects 342
Marimekko charts 97
markers
adding to charts 131
master detail relationships
creating in dimensional reporting 293
creating in relational reporting 244
creating when authoring active reports 197
denial of service errors in dimensional reporting 499
denial of service errors in relational reporting 245
linking members from two dimensional data sources 295
using to join page sets 439
matrices 13
matrix charts 143
matrix reports 67
MDX
adding your own in dimensional reporting 296
converting to in dimensional reporting 297
working with in dimensional reporting 295
measures 41
specifying defaults 74
member properties 253
inserting 258
member sets
aggregating 272
creating 258, 307
member unique names 547
relational metadata 419
troubleshooting running reports with 548
members 41
drill-through access 418
excluding 261
expanding and collapsing 263
folder 253
inserting properties 258
intersections in dimensional reporting 306
moving 261
Metric Studio diagrams 158
metrics range charts 99
examples 156
MHT
saving active reports 199
macrocharts 96
inserting 152
Microsoft Excel
grouping and ungrouping repeating cells 50
limitations of nested report objects 552
producing reports in 49
report limitations 549
missing items
troubleshooting 494
missing values 542
suppressing 391
mobile devices 61
creating reports for 61
inline prompts 7
models 41
aggregation properties 222
sample models and packages 510
using design filters 45
moving averages 272
MSR-PD-0001 errors 359
MSR-PD-0012 errors 483
MSR-PD-0013 errors 483
multilingual reports 376
MUNs, See member unique names

N
named sets
limitations when summarizing 268
nested or parallel sets overlapping 479
names
updating references 459
native SQL
in dimensional reporting 295
in relational reporting 246
nested crosstab only filters on some items
troubleshooting drilling through 505
nested list reports
not running after upgrading 478
nested report objects
Microsoft Excel limitations 552
nested sets
unexpected summary values 483
nesting
data in crosstabs 71
new features
version 10.1.0 11
version 10.1.1 8
version 10.2.0 4
version 10.2.1 1
no data
specify not to render pages 47
specifying what appears 393
No Data Contents 393
notes 15
adding to charts 127
adding to maps 168
null values 542
calculations with SAP BW data sources 484
suppressing 391
suppressing with filters 391
troubleshooting in count summaries 480
number data formats
Microsoft Excel limitations 552
numeric shaping
See digit shaping

O
objects 31
adding 30
aligning 339
as containers 31
finding in reports 32
hiding and showing 372
hierarchies 32
objects (continued)
indenting 341
inheriting formatting from parents 32
inline or block 31
inserting formatting objects 338
locking and unlocking 31
reusing for layouts 344
specifying data format 388
updating reused 346
OLAP data sources
inserting data 253
limitations with relational functions 533
OP-ERR-0199 errors 214, 478
OP-ERR-0201 errors 479
OP-ERR-0209 errors 535
OP-ERR-0210 errors 535
OP-ERR-0212 errors 535
OP-ERR-0213 errors 535
opening reports
from other studios 454
ORA-00907 errors 494
Oracle data sources
troubleshooting 494
Oracle Essbase
changes 479
out of memory errors 310
overflow errors in crosstabs 493

P
packages 41, 253
changing 459
drilling through 418
referencing items in child queries 243
refreshing 40
selecting for your report 39
packed bubble visualizations 179
padding
applying to objects 341
page breaks 449
controlling 350
creating 437
creating with page layers 449
Page Explorer 26
page layers 449
page numbers
adding 446
modifying with options 449
page sets
creating 437
joining nested page sets 439
pages 29
adding 437
associating queries with 437
not rendering when there is no data 47
pagination
horizontal in PDF output 449
specifying for HTML output 350
palettes 14
conditional 111
customizing in charts 108
parameter values
clearing 45
parameters
for prompts 319
using in master detail relationships in dimensional reporting 293
regression lines (continued)  
in current default charts 135  
in legacy charts 136  
relational functions  
limitations when used with OLAP data sources 533  
relational reporting 213  
adding data 213  
best practices 22  
filtering data 228  
grouping data 214  
sorting data 255  
summarizing data 219  
troubleshooting drilling through to cubes 504  
using calculations 248  
working with queries 236  

Render Page when Empty 47  
repeater tables 186  
adding interactive behavior 4  
repeaters  
converting from lists 65  
formatting 340  
repeating cells  
grouping and ungrouping in Excel 2007 output 50  
report formats  
CSV 48  
Excel 49  
HTML 45  
PDF 45  
specifying 45  
XML 52  
report outputs  
accessibility options 469  
Microsoft Excel 2007 10  
report properties 350  
report samples 507  
report specifications 37  
Report Studio  
bubble charts 15  
bullet charts 13  
chart axes 14  
chart enhancements 12  
chart formatting 14  
chart legends 13  
color by value 15  
matrix charts 13  
new chart technology 12  
notes 15  
pie charts 14  
setting options 33  
ReportNet reports  
formatting changes when upgrading 495  
reports  
active 171  
adding interactivity 184  
changing the maximum number of tuples allowed 499  
converting to active reports 172  
multilingual 376  
out of memory errors in interactive HTML output 500  
supporting bidirectional languages 56  
validating 43  
validation options 43  
resizing  
current default charts 108  
legacy charts 108  
objects 349  
reusing  
styles 330  
rich text items  
inserting 337  
supported elements 338  
rolling averages 272  
rollup calculations  
in dimensional reporting 303  
in relational reporting 250  
round-off errors 486  
routing numbers 388  
rows  
swapping with columns 74  
rows per page  
controlling for multiple containers in HTML and PDF 48  
setting 45  
RQP-DEF-0177 errors 464, 497  
RSV-SRV-0025 errors 488  
RSV-SRV-0040 errors 477  
runtime options  
accessibility options 469  
running and moving summaries  
limitations with dimensional functions 538  
running reports 45  
against dimensional data sources 531  
against SAP BW 531  
cannot find database errors 493  
ORA-00907 errors 494  
Report Studio is slow 488  
running totals  
dimensional functions 302  
troubleshooting 496  

S  
Safari Web browsers  
drill-through links not active 496  
Sample Outdoors Company 508  
databases, models, and packages 510  
samples 507  
Sample Outdoors Company samples 507  
samples 507  
cubes 511  
database, models, and packages 510  
employees 509  
GO data warehouse 510  
GO Data Warehouse (analysis) package 514  
GO Data Warehouse (query) package 522  
GO Sales (analysis) package 523  
GO Sales (query) package 524  
GO Sales transactional database 511  
interactive 526  
packages 512  
Prompt API folder 528  
Sales and Marketing (Cube) package 512  
sales and marketing data 509  
Sample Outdoors Sales (cube) package 512  
The Sample Outdoors Company 508  
SAP BW data sources  
creating burst reports 384  
creating expression 539  
large queries 498  
limitations when calculating data 248  
limitations when sorting data 281  
null results 484  
query calculations 480  
running reports against 531
SAP BW data sources (continued)
  sectioning reports 480
  units of measure notation 531
SAP variable properties
  unsupported 531
scales
  changing for chart axes 122
  changing intervals 119
  linear 119
  logarithmic 119
scatter charts 90
scope
  filtering 229
  using in drill-through access 422
screen resolution requirements 25
scrollbars
  adding to block objects 349
search tab 25
searching
  dimensions 256
  section headings 219
sections
  creating for dimensional data 449
  creating on reports with SAP BW data sources 480
  removing 219
Secure Socket Layer
  Microsoft Excel limitations 552
select & search prompts 315
set definitions 261
set expressions
  building in dimensional reporting 304
  limitations in list reports 532
sets
  editing 261
  filtering 263
  reusing 259, 260
  sharing 2, 259, 260
sets of members
  creating 258
  nested or parallel sets overlapping 479
shared sets
  copying 260
  creating 259
  managing 260
sharing
  sets 259, 260
  showing objects 372
single-edge crosstabs 70
singletons 43
sizes
  specifying for objects 349
slicer filters
  prompting with 281
slicers
  creating 279
  solve order 539
  combined with calculated rollup aggregate function 541
dimensional data sources 542
sorting
  crosstabs 197
  data in dimensional reporting 281
  data in relational reporting 235
  limitations with SAP BW data sources 281
  lists 197
  multiple columns in dimensional reporting 283
  multiple columns in relational reporting 235
source tab 25
source trees
  customizing 255
spaces
  adding around objects 341
Specifying chart properties 81
spider charts 98
SQL
  adding your own in dimensional reporting 296
  adding your own in relational reporting 247
  Cognos in dimensional reporting 295
  Cognos in relational reporting 246
  converting to in dimensional reporting 297
  converting to in relational reporting 247
  first-rows optimization in dimensional reporting 297
  first-rows optimization in relational reporting 248
  native in dimensional reporting 295
  native in relational reporting 246
  working with in dimensional reporting 295
  working with in relational reporting 246
SSAS 2005 cubes
  troubleshooting 481
SSAS 2005 data sources
  creating expression 539
  disappearing data formats 495
SSAS cubes
  problems when very large 310
  stacked charts 100
  standard charts 100
  star charts 98
  strings
    concatenating 539
  studios
    Oracle Essbase changes 479
styles
  conditional 365
  reusing 330
subtotals
  troubleshooting in grouped lists 478
summaries
  adding simple in dimensional reporting 268
  adding simple in relational reporting 221
  adding to active reports 198
  aggregate in dimensional reporting 273
  aggregate in relational reporting 224
  automatic in dimensional reporting 273
  automatic in relational reporting 224
  average in dimensional reporting 274
  average in relational reporting 225
  calculated 541
  calculated in dimensional reporting 274
  calculated in relational reporting 225
  count distinct in dimensional reporting 275
  count distinct in relational reporting 226
  count in dimensional reporting 274
  count in relational reporting 225
  custom in dimensional reporting 275
  custom in relational reporting 226
  FOR clauses 485
  maximum in dimensional reporting 275
  maximum in relational reporting 226
  median in dimensional reporting 276
  median in relational reporting 226
  minimum in dimensional reporting 276
  minimum in relational reporting 226
  none in dimensional reporting 276
  none in relational reporting 226
  not applicable in dimensional reporting 276
summaries (continued)
not applicable in relational reporting 226
null values 542
rolling and moving averages 272
standard deviation in dimensional reporting 276
standard deviation in relational reporting 227
summarize in dimensional reporting 276
summarize in relational reporting 227
total in dimensional reporting 277
total in relational reporting 227
troubleshooting 490
variance in dimensional reporting 277
variance in relational reporting 227
summarizing data
in dimensional reporting 266
in relational reporting 219
summary filters 229
summary functions
in dimensional reporting 223
in relational reporting 224
limitations when using FOR clauses 533
using in expressions in relational reporting 224
summary text
adding to tables 471
summary values
unexpected when using nested sets 483
supporting bidirectional languages 5
suppressing
data 391
Symbian devices
creating mobile reports for 61

T

tab controls 187
table styles
inherited 5
tables
adding summary text 471
applying styles 340
inserting 340
Microsoft Excel width limitations 551
specifying cell headers 471
using for layout 340
tables of contents
creating 443
tabular data
viewing 45
target reports
creating for drilling through from IBM Cognos Series 7 435
templates
background colors not appearing 478
converting from reports 451
creating 451
text
adding to reports 333
creating labels for prompts 472
specifying flow in objects 347
text box prompts 315
textures
adding to legacy charts 116
time periods
showing data only for specific 273
time prompts 316
TM1 data sources
order of metadata tree 483
TM1 data sources (continued)
report differences 482
toggle button bars 187
toolbox tab 25
tooltips
adding to charts 104
hiding or showing in maps 164
top filtering 262
totals
troubleshooting running totals 496
tree prompts
adding 316
controlling the data that appears 326
treemap visualizations 176
trend lines 119
in current default charts 135
in legacy charts 136
troubleshooting
differences in the appearance of charts run in different formats or on different operating systems 499
out of memory errors with reports run in interactive HTML 500
result set of multi-fact query contains blanks 501
unable to open saved Active Report output in Firefox 8 or later 498
tuples
changing the maximum allowed in reports 499
in dimensional reporting 306
TXT data
importing your own files 355

U
UDA-SQL-0043 errors 497
UDA-SQL-0114 errors 488, 497
UDA-SQL-0206 errors 488
UDA-SQL-0458 errors 464
UDA-SQL-0564 errors 497
unbalanced hierarchies 264
unexplained number calculations 486
union query
creating 239
units of measure 387
notation for SAP BW data sources 531
unlocking objects 31
upgrading
SAP BW prompt variables 462
upgrading reports
applications errors 477
errors 464
formatting not retained 495
members not suppressed for SAP BW 463
overview 461
reports will not open 493
thousands separators missing 463
user interface 25

V
validating reports 43
value markers
showing in charts 140
value prompts 315
values
showing crosstab values as percentages 72
specifying defaults in prompts 324
variable text items 186
variables
  active reports  4, 172, 184
  adding  370, 371
  conditional styles  369
viewing reports
  troubleshooting  495
views
  page design view  28
  page structure view  28
visual aids  27
visualizations
  adding  174
  changing the nesting order  182
  example  209
  extensible  174
  resizing  181
  specifying the categorical data to display  182
  specifying the range of values to display  182
  updating  183
  using embedded data  182

W
waterfall charts  96
Web browser settings  36
Web-safe color palettes  353
white spaces
  adding padding to objects  341
  margins  342

widths
  Microsoft Excel limitations  551
  specifying for objects  349
win-loss charts  96
Windows Mobile devices
  creating mobile reports for  61
word breaks  348
workspaces
  Cognos Workspace  57
  prompts in Cognos Workspace  58
wrong data appears in target report
  troubleshooting drilling through  496

X
XLS format
  importing your own files  355
  limitations  549
XML
  importing your own files  355
  producing a report  52
  report specifications  37
XQE-CON-0007 errors  498

Z
zeros
  suppressing  391