Cognos(R) 8 Business Intelligence

REPORT STUDIO

USER GUIDE

COGNOS®

THE NEXT LEVEL OF PERFORMANCE™

Product Information

This document applies to Cognos(R) 8 Version 8.2 and may also apply to subsequent releases. To check for newer versions of this document, visit the Cognos Global Customer Services Web site (http://support.cognos.com).

Copyright

Copyright (C) 2007 Cognos Incorporated.

Portions of Cognos(R) software products are protected by one or more of the following U.S. Patents: 6,609,123 B1; 6,611,838 B1; 6,662,188 B1; 6,728,697 B2; 6,741,982 B2; 6,763,520 B1; 6,768,995 B2; 6,782,378 B2; 6,847,973 B2; 6,907,428 B2; 6,853,375 B2; 6,986,135 B2; 6,995,768 B2; 7,062,479 B2; 7,072,822 B2.

Cognos and the Cognos logo are trademarks of Cognos Incorporated in the United States and/or other countries. All other names are trademarks or registered trademarks of their respective companies.

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. Cognos does not accept responsibility for any kind of loss resulting from the use of information contained in this document.

This document shows the publication date. The information contained in this document is subject to change without notice. Any improvements or changes to either the product or the document will be documented in subsequent editions.

U.S. Government Restricted Rights. The software and accompanying materials are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to the restrictions in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, or subparagraphs (C) (1) and (2) of the Commercial Computer Software - Restricted Rights at 48CFR52.227-19, as applicable. The Contractor is Cognos Corporation, 15 Wayside Road, Burlington, MA 01803.

This software/documentation contains proprietary information of Cognos Incorporated. All rights are reserved. Reverse engineering of this software is prohibited. No part of this software/documentation may be copied, photocopied, reproduced, stored in a retrieval system, transmitted in any form or by any means, or translated into another language without the prior written consent of Cognos Incorporated.

Table of Contents

Introduction *21*

Chapter 1: Report Studio *23*
- Planning Reports *23*
- Working in Report Studio *23*
  - The User Interface *23*
  - Basic Report Structure *25*
  - Working with Objects *27*
  - Set Options *28*
  - Verify Web Browser Settings *29*
- Creating a Report *30*
  - Specify the Package *30*
  - Choose a Report Template *31*
  - Add Data to a Report *31*
  - Working with Dimensional Data *32*
  - Save a Report *34*
  - Run a Report *35*
- Print a Report *39*
- Creating Your Own Report Templates *39*
  - Convert a Report to a Template *39*
  - Create a Query Studio Template *40*
- Techniques for Creating Accessible Reports *40*
- Techniques for Creating Reports for Cognos 8 Go! Office *41*
- Managing Your Reports *42*
- The Cognos 8 SDK *42*

Chapter 2: Types of Reports *43*
- List Reports *43*
  - Group Data *43*
  - Set the Group Span for a Column *45*
  - Format Lists *47*
- Crosstab Reports *48*
  - Crosstab Nodes and Crosstab Node Members *48*
  - Create a Single-Edge Crosstab Report *49*
  - Create a Nested Crosstab Report *49*
  - Specify the Default Measure *50*
  - Format Crosstabs *51*
  - Change a List into a Crosstab *52*
  - Example - Add Aggregate Data to a Crosstab Report *52*
  - Example - Create a Discontinuous Crosstab *53*
- Charts *54*
  - Example - Create a Column Chart to Plot Order Trends *55*
  - Customize a Chart *56*
  - Customize the Colors of a Chart *58*
  - Change a Chart Background *60*
  - Add a Baseline to a Chart *61*
  - Customize the Lines in a Chart *61*
  - Add a Marker to a Chart *62*
  - Add a Note to a Chart *62*
  - Specify the Chart Label Text and Value Text Shown *65*

Specify the Properties of a Gauge Chart 65
Create a Drill-up and Drill-down Chart 66
Example - Show Values in the Chart Legend 66
Define Query Context When Customizing Legend Entries, Legend Titles, or Axis Labels 67
Example - Customize the Axis Titles 68
Example - Create a Conditional Palette 69
Maps 70
Example - Create a Map Report 71
Example - Define Data Values for the Region Layer 72
Ignore Data with no Features 73
Match Data Values to Names in the Map File 73
Define Data Values for the Point Layer 74
Add Another Color to the Region or Point Layer 74
Specify the Size of Points 75
Add Titles, Subtitles, Footers, and Axis Title 75
Add Legend Titles 75
Add a Note to a Map 75
Customize a Map 76
Drill Through to Another Report From a Map 77
Edit a Map 78
Additional Maps 78
Location Intelligence 78
Repeaters 79
Convert a List into a Repeater 79
Example - Create Mailing Labels 79

Chapter 3: Formatting a Report *81*
Recommendation - Laying Out a Report 81
Report Layout Guidelines 82
The Page Structure View 82
Add a Header or Footer 82
Add Borders 83
Add Text 84
Specify the Font 84
Add Color 84
Insert an Image 85
Insert a Background Image 85
Add a Bookmark 85
Example - Add a Table of Contents to a Report 87
Insert Other Objects 88
Elements Supported in Rich Text Items 90
Example - Add a Multimedia File to a Report 90
Align an Object 91
Use Tables to Control Where Objects Appear 91
Apply a Table Style 91
Apply Padding 92
Set Margins 92
Reuse a Layout Object 92
Change a Reused Object 93
Update Reused Objects 93
Add a Page 94
Create Page Sets 94
Join Nested Page Sets 95
Example - Preparing a Product List Report 95
Reorder Columns 96
Rename a Column 97

Swap Columns and Rows 97
Setting Object Properties 98
Control Page Breaks and Page Numbering 98
Specify Text Properties 99
Specify Text Flow 99
Specify the Height and Width 100
Control How Objects Flow Around Other Objects 100
Specify Borders and Cell Size in Tables 101
Setting Report Properties 101
Create and Modify Classes 101
Modify Classes to Format Query Studio Reports 102
Modifying the Default Layout Style Sheet 102

Chapter 4: Working with Data *103*
Filter Data 103
Define a Slicer 104
Filtering Data Using an SAP BW Data Source 105
Ignoring the Time Component in Date Columns 105
Sort Data 106
Perform Advanced Sorting 107
Create Sections 107
Remove Sections 108
Summarizing Data 108
Use the Aggregation Properties Set in the Model to Specify Detail or Summary Aggregation 109
Set the Auto Group & Summarize Property 109
Use the Aggregate Button to Add Simple Summaries 109
Specify Aggregation Properties for a Data Item 110
Use Aggregate Functions in Data Item Expressions 110
Specify the Aggregation Mode 111
Aggregating Values in Crosstabs and Charts 111
Create a Calculation 113
Multiple Calculations in Crosstabs and Charts 114
Format Data 116
Specify the Format for an Object 117
Specifying the Number of Decimals 117
Locale-sensitive Properties 117
Specify the List of Properties for a Layout Object 118
Show Data for a Specific Time Period 118
Adding Prompts 119
Use the Build Prompt Page Tool 119
Build Your Own Prompt and Prompt Page 119
Create a Parameter to Produce a Prompt 123
Create a Prompt Directly in a Report Page 124
Modifying Prompts 124
Control the Data That Appears in a Tree Prompt 128
Working with Queries 128
Relating Queries to Layouts 129
Working with Dimensional Data Sources 129
Multiple-fact Queries 129
Add Queries to a Report 131
Create a Union Query 132
Create a Join Relationship 135
Add Dimension Information to a Query 135
Reference Package Items in Child Queries 137
Working with SQL or MDX 138
Add a Query Macro 140

Chapter 5: Working with Existing Reports *145*
Open and Save a Report Locally *145*
Open a Report from the Clipboard *146*
Remove Upgrade Messages *146*
Open Files from Other Studios *146*
Components of an Analysis Studio Query Specification *147*
Managing Changes in the Package *149*
Update Name References *150*
Change the Package Connection *150*
Add Multiple Items to a Single Column *150*
Example - Create a Report with Multiple Items in One Column *151*
Bursting Reports *152*
Defining Burst Recipients *152*
Specify Burst Groups *154*
Set Burst Options *155*
Enable Bursting *155*
Creating Burst Reports Using a Dimensional Data Source *156*
Example - Burst a Report *156*
Customizing Reports Using Conditions *157*
Add a Variable *158*
Adding Conditional Formatting to a Report *159*
Add Conditional Rendering *160*
Add Multiple Layouts *162*
Using Cognos 8 to Create Multilingual Reports *163*
Drill-through Access *164*
Understanding Drill-through Concepts *165*
Set Up Drill-through Access in a Report *168*
Create a Drill-up/Drill-down Report *172*
Member Sets *173*
Example - Create a Drill-up/Drill-down Report *174*
Create a Master-Detail Relationship *175*
Link Members from Two Dimensional Data Sources *176*

Chapter 6: Try It Yourself Exercises *177*
Try It Yourself - Create a List Report *177*
Try It Yourself - Create a Crosstab *178*
Try It Yourself - Create Charts *179*
Try It Yourself - Create Map Reports *188*
Show the Distribution of Revenue by Country *189*
Show Revenue and Margins for Cities in the United States *190*
Try It Yourself - Add Prompts *191*
Try It Yourself - Create a Multiple-Page Report *192*
Try It Yourself - Create a Template *194*
Try It Yourself - Create an Invoice *195*
Try It Yourself - Create a Dashboard Report *197*
Try It Yourself - Create a Dynamic Report That Shows Period-to-date Data *199*
Try It Yourself - Create a Report with Drill-through Access to Itself *200*

Appendix A: Troubleshooting *205*
A Web Server Error When Trying to Browse Images *205*
Do You Want to Allow this Webpage to Have Access to Your Clipboard Message Appears in Report Studio *205*
Values Not Recognized in Multilingual Query Items *206*
Changes in the Model Are Not Reflected in the Report If Model Versioning Is Used *206*
Problems When Printing a PDF Manual *206*
A Running Total in Grouped Reports Gives Unexpected Results *207*
Browser Does Not Display Report *207*
Unexpected Alignment of Items in a Report *207*

The java.lang.OutOfMemory Error Message Appears in Cognos Connection or Report Studio 207
Report Studio Does Not Start 207
A Report is Not Rendered Properly 208
Unable to Delete a Prompt 208

Appendix B: Samples ***209***
Sample Reports in the GO Sales Package 209
Conditional Display 209
Custom Legend 209
Orders Report 209
Product Report 209
Retailer Report (Multiple Prompt Values) 209
Returns by Order Method - Prompted Chart 209
Revenue by Sales Territory 209
Top 5 Sales Staff 209
Top Revenue (Conditional) 209
Sample Reports in the GO Sales and Retailers Package 210
Actual Sales Against Target Sales 210
Actual Sales Against Target Sales - Burst 210
Banded Report 210
Basket Analysis with Total Contribution (Top 10 Rank) 210
Business Details Drill Through 210
Consumer Trends 210
Cost of Goods 210
Custom Grouping 210
Customer Invoice 210
Global Sales 210
Global Sales (1) 210
GO Business View 211
GO Media 211
Mailing Labels 211
Margin Cost and Volume Report 211
Multi-Grain Fact 211
Multiple Charts 211
Order Analysis 211
Percent Contribution by Country 211
Product Comparison Charts 211
Product Line by Year 211
Product Line by Year-Prompt 211
Product Revenue 211
Product Revenue - Lifetime/Q2 212
Product Summary 212
Products Ranked by Revenue 212
Quantity by Retailer 212
Regional Orders 212
Report with Totals 212
Retailer Contact 212
Retailer Contact (Multiple Prompt Values) 212
Returns by Order Method 212
Revenue by Date and Amount 212
Sales Representative Contact List 212
Sales Reps Performance 212
Union Crosstab 213
Waterfall Chart 213
Sample Reports in the GO Data Warehouse Package 213
Employee Profile 213

Sales Target by Region 213
Tool Tips 213
Sample Reports in the Great Outdoors Company Package 213
Margins and Revenue Map for United States 213
Product Cost by Years 213
Profit and Revenue Combination Chart 213
Revenue by Product Line 213
Revenue by Sales Branch 213
Revenue by Year over Year Growth 213
Revenue Chart by Sales Territory and Year 214
Revenue for Specified Order Method and Sales Territory 214
Sales Territory Map 214
Slicers Filters 214

Appendix C: Chart Types 215
Choosing a Chart Type and Configuration 216
Pie Charts 216
Column Charts 217
Progressive Column Charts 217
Pareto Charts 218
Bar Charts 218
Line Charts 219
Area Charts 219
Combination Charts 220
Radar Charts 220
Scatter Charts 221
Bubble Charts 221
Point Charts 222
Quadrant Charts 222
Polar Charts 223
Metrics Range Charts 223
Gauge Charts 224
Chart Configurations 224
Standard Charts 225
Stacked Charts 225
100% Stacked Charts 226
3-D Charts 226

Appendix D: Summary Functions 227
Mapping Aggregation Functions From Data Sources to Framework Manager and Report Studio 230
Changes in the Behavior of Count Between Cognos ReportNet(R) 1.1 and Cognos 8 231

Appendix E: Using the Expression Editor 233
Creating Expressions Using SAP BW Data Sources 234
Browse the Data of a Data Item 234
Using Quotation Marks in Literal Strings 235
Recommendation - Use Member Unique Name (MUN) Aliases 235
Functions Not Available When Creating a Report or Layout Expression 235
Dimensional Coercion Rules 236
Operators 239
Summaries 244
Member Summaries 253
Constants 256
Constructs 257
Business Date/Time Functions 258
Block Functions 261
Macro Functions 262

Common Functions 266
DB2 280
Informix 292
MS Access 296
Oracle 302
Red Brick 309
SQL Server 313
Teradata 319
SAP BW 324
Sybase 325
Report functions 331

Appendix F: Producing Reports in Microsoft Excel Format ***341***

Microsoft Excel Limitations 341
Unable to Load Images from the Cognos 8 Content Store in a Report 341
A Blank Worksheet is Opened 341
A Warning Message Appears When Excel Opens a Cognos 8 Report 341
Using Reports Saved in XLS Format 341
Loading Excel Reports in Netscape 7.01 Is Not Supported 341
Nested Labels in Charts Are Not Supported 341
Data Series Are Truncated 342
Charts and Custom Colors 342
Repeating Pie Charts 342
Discrete Axis Label Skip Control in Charts 342
Formatting Limitations 342
Overline Text Format 344
Text Strings with More Than 255 Characters 345
Reports with More Than 256 Columns 345
Table and Column Width 345
Excel Formats and Secure Socket Layer (SSL) 345
Number Format Becomes Currency Format in Japanese Excel 345
Report Shows Data in Wrong Columns 345
Cognos 8 Limitations 346
Accessing Reports on a Remote Server 346
Drill-through Reports 346
Map Reports 346
Formats Not Supported for Reports in Excel Format 346
Hyperlink Buttons 346
Emailing Reports in Excel Format 346
Charting Support in Excel and Cognos 8 346

Appendix G: Report Studio Object and Property Reference ***351***

Report Studio Objects 351
3-D Area 351
3-D Bar 351
3-D Combination Chart 351
3-D Line 352
3-D Scatter Chart 352
Angular Axis 352
Angular Measure 352
Area 352
As of Time Expression 352
Axis Title 353
Bar 353
Baseline 353
Baseline 353
Baseline 353
Baseline 353

Block 354
Bookmark 354
Bubble Chart 354
Bubble Measure 354
Calculated Member 354
Caption 354
Chart Body 355
Chart Footer 355
Chart Node Member 355
Chart Subtitle 355
Chart Text Item 355
Chart Title 355
Class 356
Combination Chart 356
Component Override 356
Conditional Block 356
Conditional Block 356
Crosstab 356
Crosstab Columns 357
Crosstab Columns 357
Crosstab Corner 357
Crosstab Fact Cells 357
Crosstab Intersection 358
Crosstab Member Fact Cells 358
Crosstab Node Member 358
Crosstab Rows 358
Crosstab Rows 358
Crosstab Space 359
Cumulation Line 359
Cumulation Line Axis 359
Cumulation Line Label 359
Data Item 359
Date & Time Prompt 359
Date Prompt 360
Default Measure 360
Detail Filter 360
Dimension 360
Display Layer 360
Fact 360
Field Set 361
Gauge Chart 361
Gauge Labels 361
Gauge Numerical Axis 361
Generated Prompt 361
HTML Item 361
Hyperlink 362
Hyperlink Button 362
Image 362
Interval Prompt 362
Join 362
Key 362
Layout Component Reference 363
Legend 363
Legend Title 363
Level 363
Level Hierarchy 363
Line 363

List 364
List Cell 364
List Column 364
List Column Body 364
List Columns 364
List Columns 365
List Columns Body Style 365
List Columns Title Style 365
List Column Title 365
List Footer 365
List Header 366
List Page Footer 366
List Page Header 366
List Row 366
List Row Cells Style 366
Map 366
Map Location 367
Map Location 367
Map Refinement Location 367
Map Refinement Location 367
Marker 367
Marker 367
MDX 367
Member Hierarchy 368
Member Property 368
Member Set 368
Metrics Range Chart 368
Metric Studio Diagram 368
Note 368
Note Content 369
Numerical Axis 369
Numerical Axis 369
Ordinal Axis 369
Page 369
Page Body 369
Page Footer 369
Page Header 370
Page Set 370
Pareto Chart 370
Pie Chart 370
Pie Labels 370
Point Layer 371
Point Measure 371
Point Size Measure 371
Polar Chart 371
Progressive Chart 371
Prompt Button 372
Query 372
Query Operation 372
Query Reference 372
Radar Chart 372
Radial Axis 373
Radial Measure 373
Region Layer 373
Region Measure 373
Regression Line 373
Repeater 373

Repeater Table 373
Repeater Table Cell 374
Rich Text Item 374
Scatter Chart 374
Select & Search Prompt 374
Slicer Member Set 374
SQL 375
Summary Filter 375
Table 375
Table Cell 375
Table Row 375
Target Measure 375
Text Box Prompt 375
Text Item 376
Time Prompt 376
Tolerance Measure 376
Total Column 376
Total Column Label 376
Tree Prompt 376
Value Prompt 377
Variable 377
X Axis 377
X Axis 377
X Axis Measure 377
Y Axis 377
Y Axis 378
Y Axis 1 378
Y Axis 2 378
Y Axis Measure 378
Z Axis 378
Z Axis 378
Z Axis Measure 379
Report Studio Properties 379
3-D Viewing Angle 379
Absolute Position 379
Across 379
Aggregate Function 379
Allow 45° Rotation 380
Allow 90° Rotation 380
Allow Is Missing 380
Allow Skip 380
Allow Stagger 380
Angular Axis 380
Application 380
Auto Group & Summarize 381
Auto-Sort 381
Auto-Submit 381
Auto Truncation 381
Avoid Division by Zero 381
Avoid Label Collision 381
Axis Assignment 381
Axis Assignment 382
Axis Labels 382
Axis Line 382
Axis Title 382
Background Color 382
Background Image 382

Bar, Line, or Area Index *383*
Baselines *383*
Block Variable *383*
Border *383*
Border Color *383*
Borders *384*
Bottom Position (px) *384*
Bottom Position (px) *384*
Box Type *384*
Box Type *384*
Calculation Intersection *385*
Calendar Type *385*
Caption *385*
Caption *385*
Cardinality *385*
Cascade Source *385*
Case Insensitive *385*
Catalog *385*
Chart Orientation *386*
Chart Type *386*
Chart Type *386*
Class *386*
Clock Mode *386*
Color Legend Title *386*
Color Legend Title *387*
Column Intersection *387*
Column Titles *387*
Component Reference *387*
Component Reference *387*
Conditional Palette *387*
Connecting Lines *387*
Contents Height *388*
Cross Product Allowed *388*
Cumulation Axis *388*
Cumulation Label *388*
Cumulative Line *388*
Current Block *388*
Custom Label *388*
Data Format *389*
Data Item *389*
Data Item *389*
Data Item Label *389*
Data Item Label *389*
Data Item Label *390*
Data Item Label *390*
Data Item Value *390*
Data Item Value *390*
Data Item Value *390*
Data Item Value *390*
Data Item Value *391*
Data Item Value *391*
Data Item Value *391*
Data Language *391*
Data Source *391*
Default Measure *391*
Default Measure Solve Order *391*
Default Selections *392*

Default Title 392
Define Contents 392
Define Member Sets 392
Depth 392
Description 392
Detail 392
Diagram Identifier 392
Dial Outline Color 393
Dictionary 393
Display After Overall Header 393
Display Frequency 393
Display Milliseconds 393
Display Seconds 393
Display Value 393
Down 394
Drill Throughs 394
Duplicates 394
Embed 394
Execution Method 394
Execution Optimization 394
Expression 395
Expression 395
Expression 395
Face Color 395
Fact Cells Precedence 395
Fill Effects 395
First Column Color 396
First Date 396
First Label Index 396
Floating 396
Font 396
Font Auto-Sizing 397
Footer 397
Foreground Color 397
Gauge Labels 397
Gauge Palette 398
Generated SQL/MDX 398
Gridlines 398
Grouping & Sorting 398
Grouping Type 398
Group Span 398
Has Fact Cells 398
Height (px) 398
Hide Adornments 399
Hide Text 399
Hole Size (%) 399
Horizontal Alignment 399
HTML 399
HTML Source Variable 399
Ignore Data with No Features 400
Include Zero For Auto Scale 400
Indentation Length 400
Indentation Start Level 400
Intersection Name 400
Join Relationships 400
Label 401
Label 401

Label *401*
Label *401*
Label *401*
Label *401*
Label Control *401*
Labels *402*
Labels *402*
Last Date *402*
Left Position (px) *402*
Left Position (px) *402*
Legend *402*
Legend Label *402*
Legend Title *403*
Limit Type *403*
Line *403*
Line Style *403*
Line Styles *403*
Line Weight (pt) *403*
Map & Layers *403*
Map Drills *403*
Margin *404*
Marker Color *404*
Marker Label *404*
Markers *404*
Marker Text Location *404*
Master Detail Relationships *404*
Maximum Characters *404*
Maximum Execution Time *405*
Maximum Rows Retrieved *405*
Maximum Size (pt) *405*
Maximum Tables *405*
Maximum Text Blob Characters *405*
Maximum Truncation Characters *405*
Maximum Value *406*
MDX *406*
Member Offset (%) *406*
Minimum Size (pt) *406*
Minimum Value *406*
Minor Gridlines *406*
Multi-Line *407*
Multi-Select *407*
Name *407*
Name *407*
Name *408*
Name *408*
Name *408*
Name *408*
Negative Column Color *408*
No Data Features Size (pt) *408*
Note Border *408*
Notes *408*
Number of Regression Lines *409*
Numbers Only *409*
Numerical Axis *409*
Numerical Axis *409*
Numeric Value *409*
Numeric Value *409*

Ordinal Axis *409*
Outer Join Allowed *410*
Override Dimension Info *410*
Overrides *410*
Padding *410*
Pagination *410*
Palette *410*
Palette *411*
Parameter *411*
Percentile *411*
Percent of Axis *411*
Performance Pattern *411*
Pie Labels *411*
Point Color *411*
Point Shape *411*
Point Shape *412*
Point Size (pt) *412*
Polynomial Exponent *412*
Position *412*
Positive Column Color *412*
Pre-populate If Parent Optional *412*
Pre-populate Levels *412*
Pre-Sort *413*
Processing *413*
Progressive Axis *413*
Projection List *413*
Properties *413*
Push To Bottom *413*
Query *413*
Radar Type *414*
Radial Axis *414*
Range *414*
Range Label *414*
Regression Line *414*
Regression Type *414*
Relative Alignment *415*
Render Variable *415*
Repeater Direction *415*
Report Expression *415*
Report Expression *415*
Report Expression *415*
Report Expression *416*
Report Expression *416*
Report Expression *416*
Report Expression *416*
Report Expression *416*
Report Expression *416*
Required *417*
Right Position (px) *417*
Rollup Aggregate Function *417*
Rollup Processing *417*
Rotate Labels *417*
Row Intersection *418*
Rows Per Page *418*
Scale *418*
Scale Interval *418*
Scope *418*

Selector *418*
Selector *418*
Select UI *419*
Separator *419*
Series Color *419*
Set Operation *419*
Show Caption *419*
Show Data Range in Legend *419*
Show Features with No Data *419*
Show Feelers *419*
Show Legend Values *420*
Size & Overflow *420*
Size Legend Title *420*
Solve Order *420*
Sorting *420*
Source Type *420*
Source Type *421*
Source Type *421*
Source Type *421*
Source Type *421*
Source Type *421*
Source Type *421*
Spacing & Breaking *422*
Spider Effects *422*
SQL *422*
SQL Syntax *422*
Standard Deviations *422*
Static Choices *422*
Style Variable *422*
Subtitle *423*
Suppress *423*
Table Properties *423*
Target Color *423*
Target Marker *423*
Target Marker Border Color *424*
Target Marker Position *424*
Target Range (%) *424*
Text *424*
Text Flow & Justification *424*
Text Source Variable *424*
Title *425*
Tolerance Color *425*
Tolerance Label *425*
Tooltips *425*
Top Position (px) *425*
Total Column *425*
Total Column Color *425*
Truncation *426*
Truncation Text *426*
Type *426*
Type *426*
Upper Range Skew (%) *426*
URL *426*
URL Source Variable *426*
Usage *427*
Use 1.x Behavior *427*
Use Detail Value On Page *427*

Use For Parameter Info 427
Use Local Cache 427
Use Same Range For All Instances 427
Use SQL With Clause 428
Use Thousands Separator 428
Use Value 428
Use Value 428
Value Location 428
Value Representation 428
Values 428
Values 429
Values 429
Value Type 429
Vertical Alignment 429
Visible 429
Visual Angle 429
White Space 430
Width (px) 430
X Axis 430
Y1 Axis 430
Y2 Axis 430
Y2 Axis Position 430
Y Axis 430
Data Formatting Properties 431
"Not Applicable" Characters 431
Any Error Characters 431
Calendar Type 431
Clock 431
Currency 431
Currency Display 431
Currency Symbol 431
Currency Symbol Position 431
Date Ordering 431
Date Separator 432
Date Style 432
Decimal Separator 432
Display AM / PM Symbols 432
Display As Exponent 432
Display Days 432
Display Eras 432
Display Hours 432
Display Milliseconds 432
Display Minutes 432
Display Months 433
Display Months 433
Display Seconds 433
Display Time Zone 433
Display Weekdays 433
Display Years 433
Display Years 433
Divide By Zero Characters 433
Exponent Symbol 433
Group Size (digits) 433
International Currency Symbol 434
Mantissa (digits) 434
Maximum No. of Digits 434
Minimum No. of Digits 434

Missing Value Characters 434
Negative Pattern 434
Negative Sign Position 434
Negative Sign Symbol 434
No. of Decimal Places 434
Numeric Overflow Characters 434
Padding Character 435
Pattern 435
Percentage Symbol 435
Percent Scale (integer) 435
Scale 435
Secondary Group Size (digits) 435
Security Error Characters 435
Thousands Separator 435
Time Separator 435
Time Style 435
Time Unit 436
Use Thousands Separator 436
Zero Value Characters 436

Appendix H: Using Patterns to Format Data ***437***
Pattern Guidelines 437
Date and Time Symbols 438
Decimal Format Symbols 444

Appendix I: Drilling Through from Cognos Series 7 to Cognos 8 ***447***
Setting Up Drill-through Access from PowerPlay Web 447
Setting Up Drill-through Access from Cognos Visualizer 447
Create and Test the Target Report 447

Glossary ***449***

Index ***453***

Introduction

This document includes 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.

To use this guide, you should have

- knowledge of your business requirements
- experience using a Web browser, developing Web applications, and writing reports
- knowledge of databases and data modeling concepts

The following documents contain related information and may be referred to in this document. To view these documents, either consult your administrator to determine the online location of Cognos documentation or search the Knowledge Base of the Cognos Global Customer Services Web site: (http://support.cognos.com/kb-app/knowledgebase). If you require logon credentials, either consult with your administrator or send an email to support.america@cognos.com.

Document	Description
Cognos 8 Getting Started	Teaching new users how to use Cognos 8
Cognos Connection User Guide	Using Cognos Connection to publish, find, manage, organize, and view Cognos content, such as scorecards, reports, analyses, and agents
Cognos 8 Administration and Security Guide	Managing servers, security, reports, and portal services; setting up Cognos samples; and customizing Cognos 8
Analysis Studio User Guide	Exploring, analyzing, and comparing dimensional data
Framework Manager User Guide	Creating and publishing models using Framework Manager
Metric Studio User Guide	Authoring scorecard applications and monitoring the metrics within them
Query Studio User Guide	Creating self-service business intelligence reports
Map Manager Installation and User Guide	Installing and using Map Manager to import and manage maps that are used in map reports
Cognos 8 Troubleshooting Guide	Helping administrators to quickly identify and solve problems

For more information about using this product, visit the Cognos Global Customer Services Web site (http://support.cognos.com). For information about education and training, click the **Training** link from this site.

This document is available as online help and as an online book.

From within the online help, you can click the following link to open a printable version of this document (PDF).

Our documentation includes user guides, tutorial guides, reference books, and other materials to meet the needs of our varied audience.

Online Help

All information is available in online help. Online help is available from the help button in a Web browser, or the **Help** menu and help button in Windows products. You can also download the online help from the Cognos Global Customer Services Web site (http://support.cognos.com).

Books for Printing

The information in each online help system is available in online book format (PDF). However, the information from a given help system may be divided into more than one online book. Use online books when you want to print a document or when you want to search the whole document.

You can print selected pages, a section, or the whole book. Cognos grants you a non-exclusive, non-transferable license to use, copy, and reproduce the copyright materials, in printed or electronic format, solely for the purpose of providing internal training on, operating, and maintaining the Cognos software.

Online books are available from the introduction to the online help for each component.

All online books are available on the Cognos documentation CD.

You can also read the product readme files and the installation guides directly from Cognos product CDs.

Chapter 1: Report Studio

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

Planning Reports

Before you use Report Studio to design and create a report, some information is required. For each report, answer the following questions:

- Who is the target audience?
- What business question do you want to answer?
- Which type of report will best answer the business question?
- What are the data sources, and where are they?
- Which data items and calculations are needed?
- What are the security needs?
- How will the report be distributed?

Answering some of these questions involves working with other Cognos 8 components. For example, Framework Manager is the metadata tool used to model your enterprise data. Models created in Framework Manager provide the data items that you use in reports.

Working in Report Studio

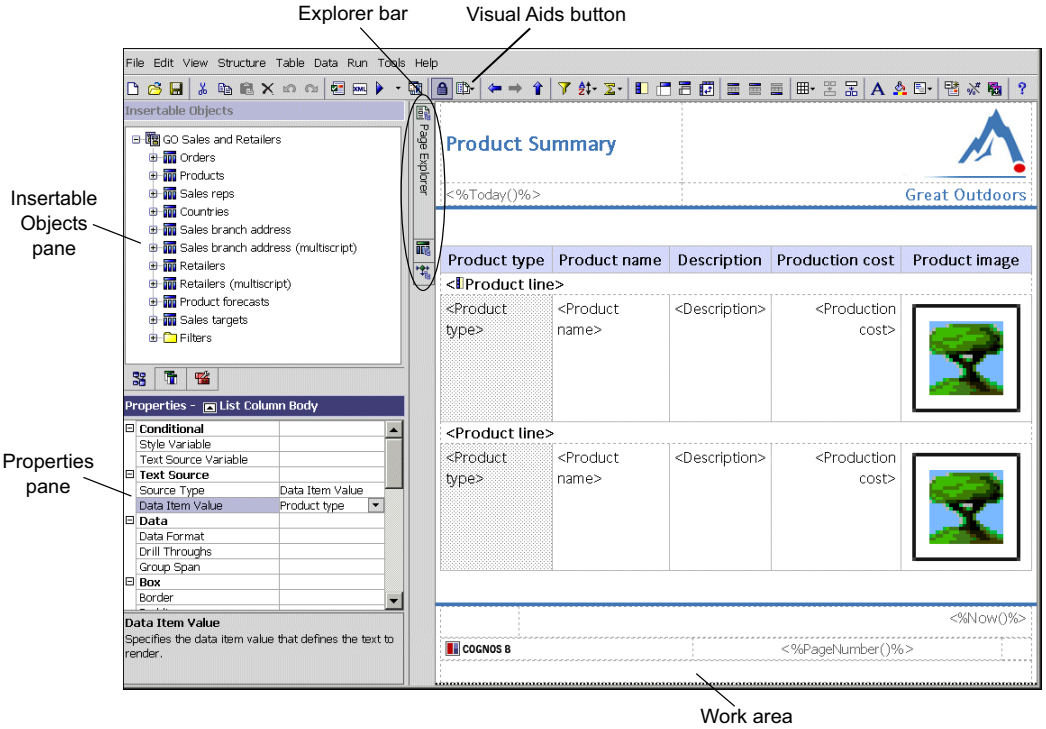
To create reports in Report Studio, you must become familiar with the Report Studio environment.

To work effectively in Report Studio, do the following:

- ❑ Familiarize yourself with the user interface.
- ❑ Learn about basic report structure, which includes the layout and queries.
- ❑ Learn how to work with report objects.
- ❑ Set the options you want.
- ❑ Verify Web browser settings.

The User Interface

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



Insertable Objects Pane

The **Insertable Objects** pane contains objects that you can add to a report. You add objects to a report by dragging them to the work area.

The **Insertable Objects** 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.

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 clicking F1. For example, you can view the list of objects that uses 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: View a description of the currently selected property at the bottom of the pane. From the **View** menu, click **Property Descriptions**.

Explorer Bar

Use the following buttons on the Explorer bar to work with different parts of a report:

- Pause the pointer over the page explorer button to go to a specific report page or prompt page.

  Tip: You can also add a new report page or add a prompt page (p. 119), as well as create and modify classes (p. 101).

- Pause the pointer over the query explorer button to work with queries (p. 128).

  You use Query Explorer to create or modify queries and to perform complex tasks, such as defining union joins and writing SQL statements.

- Pause the pointer over the condition explorer button to work with variables.
  You use variables to define conditions (p. 157) in a report.

Work Area

The work area is where you design reports.

Visual Aids Button

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

Option	Description
Hide boundary lines	Hides all boundary lines.
Show boundary lines	Shows all boundary lines.
Show boundary lines (override)	Shows boundary lines when the **Border** property for an object is set to **None**.
Show repeating	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.
Show drag & drop padding	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 Report Studio uses to show drag-and-drop zones, only the minimum padding is shown.
Show hidden objects	Shows objects for which the **Box Type** property was set to **None**.
Show sorting	Shows the sorting icon for data items for which a sort order (p. 106) was specified.
Show grouping	Shows the grouping icon for grouped data items (p. 43).
Show source type	Shows the source type icon of objects, such as layout calculation.
Show drill throughs	Shows data items for which a drill-through definition (p. 168) was defined as hyperlinks.
Show bookmarks	Shows bookmarks (p. 85) inserted in the report.
Show master detail relationships	Shows master detail relationships (p. 175) defined in the report. **Tip:** Pausing the pointer over the master detail relationship icon shows the relationship.

Basic Report Structure

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

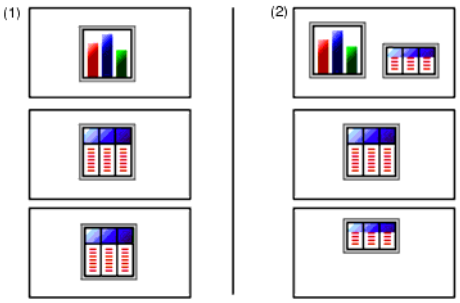
- present the data in a meaningful way by using lists, crosstabs, charts, and maps
- give the report the appearance you want by adding formatting, such as borders, color, images, and page numbers
- specify how the data flows from one page to the next

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.



(1) The chart appears on its own page. The list begins on the next page and fills subsequent pages until all rows appear.

(2) The chart and the first few rows of the list appear on the first page. The rest of the data in the list appears on the subsequent pages.

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 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

  Add a chart to show data in a graphical format.
- map

  Add a map to plot data geographically.
- repeater

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

  Add a text item to show textual information.
- block

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

  Add a table to arrange or align layout objects in a left to right, top to bottom fashion.

For information about other layout objects you can add to a report, see "Insert Other Objects" (p. 88).

Queries

Queries determine what data items appear in the report. Sometimes you want detail rows of data, which you obtain by using a simple SELECT statement. Other times you must calculate totals or averages using aggregate functions and grouped columns, or must apply filters to show only the data you want.

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.

For more information about queries, see "Working with Queries" (p. 128).

Working with Objects

You build reports by adding objects and manipulating them to obtain the results you want. To understand how to work with objects in Report Studio, you must be familiar with the following concepts:

- object types
- objects as containers
- locking and unlocking objects
- hierarchy of objects

For information about Report Studio objects, see "Report Studio Object and Property Reference" (p. 351).

Object Types

In 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.

Note: You can also use an object's floating (p. 100) 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 (p. 91), blocks (p. 88), and any report frame (p. 43), 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.

Tip: 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 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.

Set Options

Set the options you want when working in Report Studio.

Option	Description
Show members folder	When working with dimensional data, shows members folders for each level (p. 32).
Show startup dialog	Shows the **Welcome** dialog box at startup.
Resize window at startup	Maximizes the Report Studio window at startup.
In-place edit	Enables the editing of text in place when double-clicking.
Group properties	Groups properties in the **Properties** pane (p. 24), such as **Text Source**, **Data**, and **General**.
Allow local file access	Enables the opening and saving of reports on your computer (p. 145).
Reuse Cognos Viewer window	Reuses the same Cognos Viewer window when you rerun a report without first closing the window.
Resize Cognos Viewer window	Maximizes the Cognos Viewer window when you run a report.
Automatically validate expressions	Automatically validates expressions, such as filters, created in the expression editor (p. 233).
Delete unreferenced query objects	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.
Show hidden layout objects	Shows layout objects in which the **Visible** property was set to **No**.
Wrap text in editors	Automatically wraps text in all editors where you can define expressions.
Animate explorers	Animates the appearance of the **Page Explorer**, **Query Explorer**, and **Condition Explorer** (p. 24).
Use Windows skin	Replaces the current appearance of the interface with the display scheme specified by the Windows settings.
Automatically populate values list	When building expressions in the expression editor, automatically shows values when browsing the data of a data item (p. 234).
Alias member unique names	When working with a dimensional data source, creates an alias when you add a member to the report or to an expression (p. 235).

Steps

1. From the **Tools** menu, click **Options**.
2. On the **General**, **View**, and **Edit** tabs, set the options you want.
3. On the **General** tab, do the following:
   - In the **Request time out (seconds)** box, type the maximum time allowed for an operation to execute before it is cancelled.
   - If you are working with dimensional data, in the **Member display count limit** box, type the maximum number of members that can appear in the **Insertable Objects** pane when performing a search (p. 34).
   - If you are working with maps, in the **Feature display limit** box, type the maximum number of features that can appear in a map.
4. On the **View** tab, under **Layout dimensions**, type the width and height of the area where you will create reports.
5. On the **Edit** tab, do the following:
   - In **Tab stops**, specify the length of tabs.

     This option allows you to add and remove tabs by pressing Tab and Shift+Tab.
   - In **Aggregation mode**, if you are working with a dimensional data source, click the aggregation type you want Report Studio to use when aggregating values in crosstabs and charts.

Aggregation mode	Description
Within detail	Aggregates the visible details
Within aggregate	Aggregates the visible aggregates at the next lower level of detail
Within set	Aggregates the member sets

   For more information about aggregating values in crosstabs and charts, see "Aggregating Values in Crosstabs and Charts" (p. 111).
6. Click **OK**.

Verify Web Browser Settings

Cognos 8 uses the default browser configurations provided by Microsoft, Netscape, and Firefox. For all browsers, you must ensure that settings are enabled for cookies and Java scripts. Additional required settings are specific to the browser.

Browser	Setting
Internet Explorer	Allow Cookies
	Active Scripting
	Allow META REFRESH
	Run ActiveX controls and plug-ins
	Script ActiveX controls marked safe for scripting
	Binary and Script Behaviours
	Allow programmatic clipboard access

Note: 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. Cognos 8 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 browser configuration and cookies used by Cognos 8, see the Cognos 8 Installation and Configuration Guide.

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, as well as the layouts and styles used to present the data. For simplicity, the report specification is named the report.

Before creating a report, ensure that you have the planning information (p. 23) you need.

Creating a new report involves

- ❑ specifying the package
- ❑ choosing a report template
- ❑ adding data items
- ❑ saving the report
- ❑ running the report

You can then format the report (p. 81) and manipulate the data that will appear in the report (p. 103). For information about other tasks that are specific to a report type, see "Types of Reports" (p. 43).

Specify the 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, Framework Manager. A model is a set of related objects, such as query subjects, dimensions, filters, and calculations. When you open a package in Cognos 8, these model objects are visible in the left frame.

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

Steps

1. In the Cognos Connection Welcome page, click the **Report Studio** link.

   If there is more than one package available, the **Select a package** page appears. Otherwise, Report Studio starts.
2. If more than one package exists, click the one you want to use.
3. 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 report template.
   - To open an existing report or template, click **Open an existing report or template** and select a report.

Objects from the selected package, such as data items, appear on the source tab of the **Insertable Objects** pane.

Tip: You can later change packages (p. 150).

Refresh the Package

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

Steps

1. In the **Insertable Objects** pane, click the source tab.
2. Right-click its background and click **Refresh**.

Tip: You can also close and reopen the report to upgrade it to the latest version of the package.

Choose a Report Template

When creating a new report, choose a report template to get you quickly started. You can select a predefined template that is formatted as a particular report type (p. 43), or you can open an existing report as a template.

Steps

1. From the **File** menu, click **New**.
2. If you want to change the package, click the ellipsis points (...) and click a different package.
3. Choose a report template:
   - To create a report using one of Report Studio's predefined templates, double-click the template you want.

     Tip: Double-click **Query Studio Template** to create a report template that Query Studio users can apply to their reports.
   - To create a report using a previously created template, double-click **Existing**, locate the template, and click **Open**.

     Tip: Click the **Type** box and then click **Templates** to only see existing templates.
   - To open an existing report as a template, double-click **Existing**, locate the report, and click **Open**. You are actually opening a copy of the report, and the original report remains unchanged.

The report template appears in the report page. For more information about templates, see "Creating Your Own Report Templates" (p. 39).

Add Data to a Report

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

You may 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.

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.

Icon	Object
	Package, which contains the objects you can insert in a report.
	Query subject, which represents a table in the database.
	In relational data sources, query item, which represents a column of qualitative data in the database, such as product name or country. In dimensional data sources, level attribute, which represents a property of a level.
	Dimension, which represents a broad grouping of descriptive data about a major aspect of a business, such as products, dates, or markets.
	Hierarchy, which represents a collection of dimensional members organized into a tree structure, with each member having one or more parent members and an arbitrary number of child members.

Icon	Object
	Level, which is a set of members that have common attributes. For example, a geographical dimension might contain levels for country, region, and 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
	Measure or fact, a query item that represents a column of quantitative data in the database, such as revenue or quantity.
	Measures folder, which contains hierarchical measures.
	Model filter.

Step

- In the **Insertable Objects** pane, on the source tab, drag each data item to the location where you want it to appear.

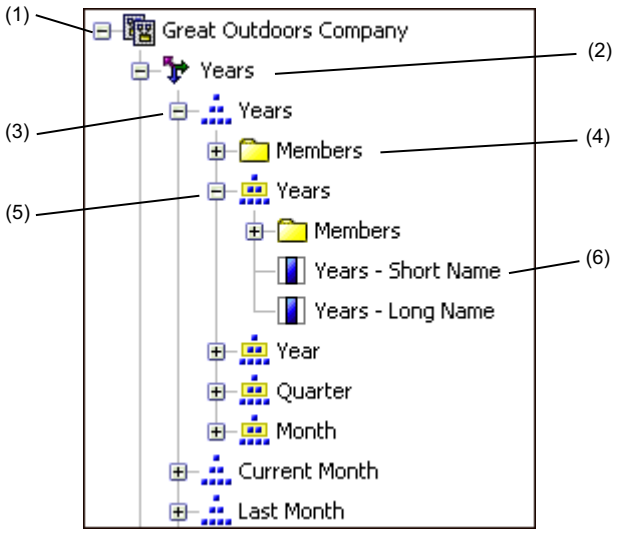
Tips: For more information about a data item, right-click the item and click **Properties**. For example, the **Usage** property identifies the intended use for the data represented by the data item. Other ways to select data items are to double-click each item, or right-click each item and click **Insert**.

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

Tip: 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.

Working with Dimensional Data

If you are working with a dimensional data source, data items are organized hierarchically. Dimensional data sources include OLAP data sources and Dimensionally Modeled Relational (DMR) data sources. The source tab in the **Insertable Objects** pane 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. Member 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.

Note: To see members folders, you must select the **Show members folder** check box in the **Tools** menu (p. 28).

(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.

Level	Level name	Description
Root	**Years**	The root level.
First	**Year**	Years in the **Years** root level. For example, 2004, 2003, and 2002.
Second	**Quarter**	Quarters for each year in the **Year** level. For example, 2004 Q1, 2004 Q2, and 2004 Q3.
Third	**Month**	Months for each quarter in the **Quarter** level. For example, Jan., Feb., and Mar.

Tip: The **Measures** dimension, which is not visible in the above diagram, 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" (p. 34).

Insert a Hierarchy

You can quickly insert entire hierarchies in a report using a single click of the mouse.

Steps

1. In the **Insertable Objects** pane, on the source tab, drag the hierarchy that you want to insert 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: When inserting all members of a hierarchy into a crosstab, you can see the hierarchical structure by specifying a value for the **Indentation Length** property.

Insert a Member Property

You can insert member properties in a report. For example, an employee level may have a property named gender.

Steps

1. In the **Insertable Objects** pane, on the source tab, drag the member property that you want to the report.
2. If the **Insert Member Property** dialog box appears, choose whether to repeat or group the property:
   - To repeat the property in the rows or columns of the crosstab, click **Insert property only**.
   - To insert the property and have it grouped, click **Insert property and group it**.

     The property will appear once in the rows or columns of the crosstab.

Search for Members

You can perform a member search to quickly find the data you want.

Steps

1. In the **Insertable Objects** pane, on the source tab, right-click a dimension, hierarchy, or level, and click **Search**.
2. In the **Words** box, type the words or characters you want to search for.
3. Click the search parameter you want to use.
4. If you want to perform a case-insensitive search, select the **Case insensitive** check box.
5. If you want to search all descendents instead of just the immediate children, select the **Search all descendents** 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.
6. Click **Search**.

Report Studio searches for all members for the object selected in step 1. The 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 into a report. This can save you time, as 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.

Save a Report

Save your report to preserve the modifications you made.

Reports are saved to the Cognos 8 server. You can also save your report on your computer (p. 145).

Steps

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 you want to save the report and type a file name.

   For information about setting up folders in Cognos Connection for your reports, see the 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 Cognos Connection. For more information, see the Cognos Connection *User Guide*.

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 Report Studio to ensure that you do not encounter multiple error messages when you run the report.

Steps

1. Open the report that you want.
2. If you want to clear parameters values stored on the Cognos 8 server, from the **File** menu, click **Clear Parameter Values**.

   Parameter values stored by the Cognos 8 server include signon, validation, and prompt information. For example, if you define two data source connections in Cognos Connection that point to the same data source, you will be prompted to choose one when you run a report. This information is stored so that you will not be prompted each time you run the report.

3. From the **Tools** menu, click **Validate Report**.

   A message box appears indicating whether any errors were found in the report.
4. If you require more detail from the validation process, revalidate the report by clicking **Validate with Options**, and do the following:
   - Click one of the following validation levels.

Validation level	Description
Error	Retrieves all errors returned from the query.
Warning	Retrieves all errors and warnings returned from the query.
Key Transformation	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 be helpful to understand the cause of errors and warnings returned from the query.
Information	Retrieves errors, warnings, key transformations, and other information related to query planning and execution.

   - 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 "Report Studio Object and Property Reference" (p. 351).

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

5. 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 (p. 113) and you want to ensure it is giving you the results you want.

   Tip: You can also view tabular data in Query Explorer (p. 128), which is useful when you are building queries.

6. 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 Cognos Connection.

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

7. Change any values you want for the current session and click **OK**.

Option	Notes
Format	None.
Paper size	Specify only if the output format is PDF.
Paper orientation	Specify only if the output format is PDF.
Data mode	Specify how much data is returned. **All Data** returns all data. **Limited Data** limits the amount of data returned based on design mode filters defined in the package. **No Data** returns artificial data instead of actual data from the data source. For more information about design mode filters, see the Framework Manager *User Guide*.
Language	The content language sets the preferred language for the data, Cognos Viewer, dates, and so on.
Rows per page	Specifies the number of rows to appear on each page. A **Rows Per Page** property exists in the **Properties** pane for lists and crosstabs. If you specify a value for this property, it overrides the same-named run option. This property applies to both HTML and PDF outputs. For more information about this property, see "Controlling the Rows Per Page for Multiple Containers in HTML and PDF" (p. 37).
Prompt	Select to be prompted (p. 119) for each prompt defined, unless the prompt is defined in a report page. If you clear the 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.

8. 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 Excel formats, and XML. You cannot produce a report in CSV or XML format if you have more than one query defined in the report, unless the additional queries are used for prompts.

The report runs in 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 Cognos Viewer depend on the capabilities set by the administrator for each user. For more information, see the *Administration and Security Guide*.

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 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 of 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 a Cognos cube as a data source.

This behavior also occurs for SAP BW data sources.

Unsupported SAP Variable Properties

Some SAP variable properties are not supported.

Exclusionary ranges appear as an inclusionary prompt.

Mandatory not initial appear as a mandatory prompt.

When using BEx to define variables in your SAP data source, avoid using exclusionary ranges and the mandatory not initial property.

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.

Cognos 8 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

Cognos 8 can produce reports in CSV format so you can open them in other applications, such as Microsoft Excel. Reports saved in CSV format

- are designed to 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 Microsoft Excel, Microsoft Windows Wordpad, and Star Office. 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 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 *Administration and Security Guide*.

Producing a Report in Excel Format

Cognos 8 can produce reports in Excel format. Three options are available:

- **Excel 2000 Single Sheet** will produce reports on one sheet that you can view in Microsoft Excel versions earlier than 2002.
- **Excel 2000** will produce reports that you can view in Microsoft Excel versions earlier than 2002.
- **Excel 2002** will produce reports that you can view in Microsoft Excel versions 2002 and later.

The Excel 2000 format can be viewed with versions of Excel 2000 and later. It supports up to 65,536 rows and multiple sheets.

Excel 2000 single sheet format offers improved security. Excel 2000 may have cookies in the URLs to spreadsheets, which could be used to illegally impersonate a user. Excel 2000 single sheet format does not use outgoing URLs. However, there is a limit of 65,536 rows, and page breaks for multiple-author pages are ignored.

Excel 2002 format and Excel 2000 single sheet format also offer the following benefits:

- Both work with SSL protocol.
- Both work with a single signon.

  Secure reports can be accessed without subsequent signons because the system automatically identifies users and provides security information.
- Both work with Netscape 7.01.
- Spreadsheets are contained in a single file for reliable spreadsheet navigation.

The Cognos 8 full date format does not always appear correctly in Microsoft Excel 2000 if it is on a system with a regional setting other than English. This does not occur with Microsoft Excel 2002 or 2003 versions. To fix this, reapply the proper format in Excel.

Limitations exist when producing reports in XLS format. For more information, see "Producing Reports in Microsoft Excel Format" (p. 341).

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 *c8_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 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.

Print a Report

Print your report to obtain a paper copy.

You can also print a report from Cognos Connection. For more information, see the Cognos Connection *User Guide*.

Steps

1. If you are producing the report in PDF, from the **File** menu, click **Page Setup** and set the page options you want.
2. Run the report.
3. From the **File** menu, click **Print**.
4. Select the print options you want and click **OK**.

Creating Your Own 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. A template can include the following objects:

- HTML items
- hyperlinks
- page numbers
- images
- tables
- text items
- blocks
- layout calculations
- date
- time

For more information about these objects, see "Formatting a Report" (p. 81).

To create a report template, you can

- convert a report to a template
- create a Query Studio template

For more information about how to create a report template, see the Report Studio Tour. In addition, you can use sample templates (p. 209) provided with Report Studio.

Tip: You can add your own report templates to the **New** dialog box. For more information, see the *Administration and Security Guide*.

Convert a Report to a Template

Convert a new or existing report to a template so it can be reused.

Steps

1. Create a new report or open an existing report.
2. Add the objects you want 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 Query Studio Template

Create a Query Studio template to provide Query Studio users with a layout that they can use to create reports or apply to existing reports. You can also use the Query Studio template to define a layout for prompt pages.

Steps

1. From the **File** menu, click **New**.
2. Click **Query Studio Template** and click **OK**.

3. Pause the pointer over the page explorer button and click the report page or prompt page you want to format.

   Tip: To create a new report page or prompt page, click the **Report Pages** or **Prompt Pages** folder.

4. In the **Insertable Objects** pane, click the toolbox tab.
5. Add the objects you want to the work area.

   If you add objects that are not supported by Query Studio, the objects will be ignored when you apply the template.
6. Save the template.

Techniques for Creating Accessible Reports

Creating accessible reports ensures access of information to all users, with all levels of ability.

For example, blind users 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 text bolding 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.
- Chart types are inaccessible to the screen reader. They are rendered as images, which the screen reader ignores. A good practice when including a chart is to ensure there is a corresponding table with the same information. In this way, visually impaired individuals can still obtain the same information from the table.
- Deliver reports through HTML, as it is the most supported output format for most screen readers.
- Always ensure there is a title on the report.
- Gain an understanding for screen reading technology.
- Avoid spelling and grammar errors, as they cause the screen reading software to misinterpret the information.
- You can use prompts in accessible reports, including check boxes, radio buttons, combo boxes, and multi-select boxes. Avoid using features like calendar boxes and up and down selections on time controls.
- When choosing to use embedded Web applications or drill-through paths, ensure the target application is also accessible.

- Avoid using large, complex list or crosstab reports. Displaying the information in multiple simple lists or crosstab reports is more manageable for users of Assistive Technology.

Techniques for Creating Reports for Cognos 8 Go! Office

Cognos 8 Go! Office provides an integrated environment for Cognos products and Microsoft Office. You can use Cognos 8 Go! 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 PowerPlay Web, Query Studio, or Report Studio.

Note that to access PowerPlay content, your administrator must configure PowerPlay to work with Cognos 8. PowerPlay content that is published only to Upfront is not available to Cognos 8 Go! Office.

Because Cognos 8 Go! 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 Cognos 8 Go! Office, follow these recommendations:

- Create content to meet specific Microsoft Office integration needs.

  For example, in Cognos 8, 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 Cognos Connection and organize them with your reports in **Public Folders** or **My Folders**. For more information, see the 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 Cognos 8 Go! 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 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 Cognos 8.
- Format elements in the Office application.

  Instead of formatting objects in Cognos 8, add the formatting that you want in the Office application. By applying less formatting in Cognos 8, more data can be imported 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 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 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.

- If you have an image in a list, specify its height and width.

  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 Cognos 8 have image maps associated with them to enable tooltips and hotspots. Cognos 8 Go! Office cannot import tooltips and hotspots into Office applications.
- Consider the additional limitations that exist when producing reports in Excel format (p. 341).

Managing Your Reports

After you create one or more reports, you can manage them for yourself and others. You can perform the following tasks in Cognos Connection:

- Schedule a report or a group of reports to run at a specific time.
- Distribute reports to other users.
- Print a report.
- Select the language used when a report is run.
- Set prompt values.
- Maintain the history of a report.
- Maintain different versions of a report.
- Create report views.

For information about how to perform these tasks, see the Cognos Connection *User Guide*.

The Cognos 8 SDK

When you create a report in Report Studio, you are creating a report specification. A report specification is an XML file that you can view (**Tools** menu, **Show Specification**). In addition, you can view the specification for a selected object with the **Show Specification** (**Selection**) menu option.

Tip: When you are viewing the report specification in Report Studio, you cannot modify or copy parts of it.

Instead of using Report Studio, you can programmatically create or modify reports by using an editing tool to work with report specifications. You then use the Cognos 8 software development kit (SDK) to implement the reports in your Cognos 8 environment. This is useful if, for example, you must make the same modification in many reports. Rather than opening each report in Report Studio and making the change, you can automate the process using the SDK, thereby saving you time. For more information about the Cognos 8 SDK, contact your local Cognos sales office.

Tip: You can also modify the XML code in a report specification by saving the report specification on your computer (p. 145).

Chapter 2: Types of Reports

You use different report types to look at data in different ways or to answer different business questions. For example, you create a list report to show your entire customer base, but you create a crosstab report to show which sales representative has the highest sales for each product line. You can create a chart to present data graphically, and use a repeater to create mailing labels.

In Report Studio, you can create the following types of reports:

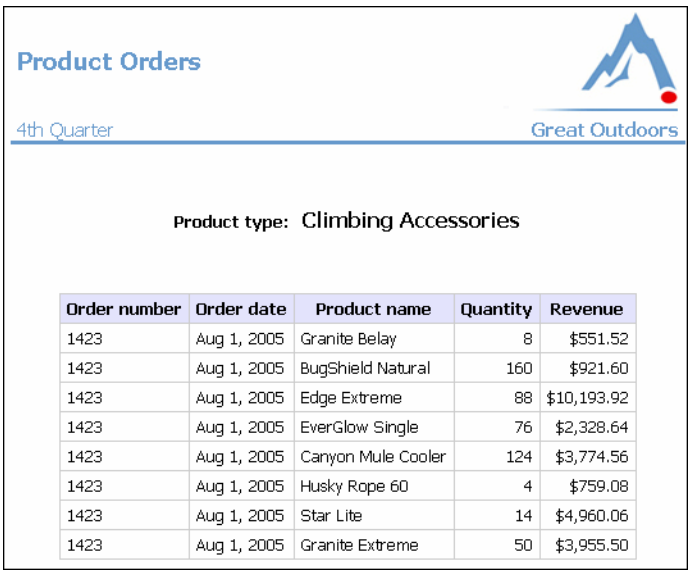
- list
- crosstab
- chart
- map
- repeater

You can also combine one or more of these in a single report.

List Reports

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

A list report is a report that 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.



Product Orders

4th Quarter

Great Outdoors

Product type: Climbing Accessories

Order number	Order date	Product name	Quantity	Revenue
1423	Aug 1, 2005	Granite Belay	8	$551.52
1423	Aug 1, 2005	BugShield Natural	160	$921.60
1423	Aug 1, 2005	Edge Extreme	88	$10,193.92
1423	Aug 1, 2005	EverGlow Single	76	$2,328.64
1423	Aug 1, 2005	Canyon Mule Cooler	124	$3,774.56
1423	Aug 1, 2005	Husky Rope 60	4	$759.08
1423	Aug 1, 2005	Star Lite	14	$4,960.06
1423	Aug 1, 2005	Granite Extreme	50	$3,955.50

For information about how to create a list report, see the Report Studio Tour.

Group Data

Group data items in a list report to remove duplicate values. For example, you have a report that shows products purchased. For each product, the product type is also shown. You group the Product type column to show only one instance of each product type in the list.

You can also group data items in repeaters (p. 79) and in page sets (p. 94).

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 Sorting" (p. 107).

If you are querying a dimensional data source, you can group or sort in the order of the existing hierarchy only. If you change the order, an error occurs. You can omit columns from the grouping. For example, if a level hierarchy contains the levels Country, State, and City, and another level hierarchy contains the level Product:

- Country, State, City is valid.
- Country, Product, and State is not valid because a level from another level hierarchy was inserted between two levels from another level hierarchy.
- Country, City, and Product is valid even though State is omitted.

To override the hierarchical order that the data source specifies, you can override the dimension information of the query. For more information, see "Add Dimension Information to a Query" (p. 135).

Steps

1. Click the column you want to group on.

   You can click either the column heading 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**.

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 (p. 82).

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 (p. 118) for a group

You can also perform advanced sorting (p. 107).

Steps

1. Click a column in the list.
2. Click the select ancestor button in the title bar of the **Properties** pane and click **List**.
3. In the **Properties** pane, double-click the **Grouping & Sorting** property.
4. If you want to change the grouping order, in the **Groups** pane, click the grouped item that you want in the **Groups** folder and drag to the location that you want in the grouping hierarchy.
5. If you want to create a new grouping, in the **Data Items** pane, click the data item that you want and drag it to the **Groups** folder in the **Groups** pane.
6. If you want to specify a list of properties for a group, do the following:
   - Click the group that you want.

     Tip: Click **Overall** to specify a list of properties for the overall group. The overall group is the grouping of all data into one group. You can specify a list of properties for the overall group to add overall totals to the list.

   - Click the properties button.
   - Select the appropriate check boxes.
   - Click **OK**.
7. Click **OK**.

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 and city are both grouped, you can choose to show the country name each time

- the country changes, by spanning Country by Country
- the city changes, by spanning Country 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.

Steps

1. Click the column for which you want to set the group span.
2. In the **Properties** pane, click the **Group Span** property and click the column you want to span.

Example - Remove Identical Values in a List

You are a report author at The Great 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.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.

4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:
   - **Order number**
   - **Order date**
   - **Product name**
   - **Quantity**
   - **Revenue**

   Tip: You can find these data items in the **Orders** and **Products** folders.
6. Group the **Order number** column.
7. Click the **Order date** column.
8. In the **Properties** pane, click the **Group Span** property and then click **Order number**.

When you run the report, the first row of the **Order date** column appears for each **Order number** row.

Order number	Order date	Product name	Quantity	Revenue
1153	Apr 14, 2004	BugShield Extreme	474	$2,697.06
		EverGlow Butane	74	$4,542.12
		EverGlow Single	130	$4,087.20
		Firefly 2	254	$6,375.40
		Firefly Multi-light	114	$2,829.48
		Insect Bite Relief	254	$1,229.36
		TrailChef Canteen	100	$1,181.00
1154	Oct 1, 2004	Canyon Mule Extreme Backpack	14	$6,447.28
		Firefly Extreme	104	$5,328.96
		TrailChef Water Bag	122	$803.98
1155	Apr 21, 2004	EverGlow Double	68	$3,546.20
		EverGlow Lamp	174	$4,838.94
		Seeker 50	30	$4,023.30
		Sun Blocker	86	$419.68
		Sun Shelter Stick	54	$225.18
		Sun Shield	376	$1,398.72
		TrailChef Canteen	58	$726.74
1156	Sep 9, 2004	Aloe Relief	90	$433.80
		Firefly Multi-light	94	$2,494.76
		Sun Blocker	60	$292.80

Example - Repeat a Column Value in a List

You are a report author at The Great 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.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.

5. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:
   - **Product line**
   - **Product type**
   - **Product name**
   - **Description**
   - **Production cost**

   Tip: You can find these data items in the **Products** folder.
6. Group the **Product line** and **Product type** columns.
7. Click the **Product line** column.
8. In the **Properties** pane, click the **Group Span** property and click **Product type**.

9. Click any part of the list and then click the select ancestor button in the title bar of the **Properties** pane.
10. Click **List**.
11. In the **Properties** pane, double-click the **Font** property.
12. In the **Size** box, click **8pt** and click **OK**.

When you run the report, the product line appears whenever the product type changes.

Product line	Product type	Product name	Description	Production cost
Camping Equipment	Lanterns	Firefly Multi-light	This light can be used as either a flashlight or as a lamp. Uses two C batteries, included. Burn time 7-8 hours. Water-resistant.	$17.78
		Flicker Lantern	Simple to use, just requires a candle to be inserted and lit. Windproof, lasts for hours.	$24.62
Camping Equipment	Packs	Canyon Mule Carryall	This pack has a large capacity, perfect for carrying all your camping gear. Solid metal zipper and leather bottom.	$41.18
		Canyon Mule Climber Backpack	This pack is perfect for day trips and short hikes. Also great for students. Separate front compartment, multi-layered interior organizer, 2 cm waist belt, 32,000 cu. cm.	$62.50
		Canyon Mule Cooler	A durable plastic cooler with hinged top, perfect for storing small items. Size:100 x 50 x 50 cm.	$24.00
		Canyon Mule Extreme Backpack	Perfect for long back country trips, this pack features an expandable front pocket, includes a large sleeping bag compartment, padded shoulder harness, back and waist belt. 90,000 cu. cm.	$238.88
		Canyon Mule Journey Backpack	Set of three packs: handlebar pack, front pack and rear pack. Designed to fit most mountain bikes. Includes mounting hardware. Total volume: 66,000 cu. cm.	$213.33
		Canyon Mule Weekender Backpack	A weekend getaway requires this pack. It features a large front compression pocket, harness and waist belt, foam molded back panel, vinyl covered gear loops. 50,000 cu. cm.	$166.66
Camping Equipment	Sleeping Bags	Hibernator	The Hibernator is a three-season sleeping bag. The rectangular shape allows for easy pairing. Nylon shell material. 8 cm loft. Comfortable down to 0°C. One size fits up to 195 cm.	$86.00
		Hibernator Camp Cot	Aluminum frame camp cot that is lightweight and durable. Size: 100 x 225 x 50 cm. Weight: 7 kg.	$65.33

Format Lists

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 (p. 91).

Steps

1. To format an individual column, do the following:
   - Click the column title or column body.

     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.

     To format the whole column, click the select ancestor button in the title bar of the **Properties** pane and click **List Column**.
   - In the **Properties** pane, click the property you want and specify a value.

     For example, to specify a background color, click **Background Color** and choose the color you want to use.
2. To format all columns, do the following:
   - Click a column in the list.
   - To format list column bodies, click the select ancestor button in the title bar of the **Properties** pane and click **List Columns Body Style**.
   - To format list column titles, click the select ancestor button in the title bar of the **Properties** pane and click **List Columns Title Style**.
   - To format entire columns, click the select ancestor button in the title bar of the **Properties** pane and click **List Columns**.
   - To format the entire list, click the select ancestor button in the title bar of the **Properties** pane and click **List**.
   - In the **Properties** pane, click the property you want and specify a value.

     For example, to specify a background color, click **Background Color** and choose the color you want to use.

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 have green as their background color.

For more information about formatting reports, see "Formatting a Report" (p. 81).

Crosstab Reports

Use crosstab reports to show information in a more compact form than in a grouped list. For example, create a crosstab report to show total sales by product line generated by each sales representative.

Like list reports, crosstab reports are reports that show data in rows and columns. However, the values at the intersection points of rows and columns show summarized information rather than detailed information.

Product Line Profitability

Jun 16, 2005

					Quarter 1	
		Revenue	Cost of Goods Sold	Gross profit	Gross Profit Margin	Revenue
Camping Equipment	Cooking Gear	$275,929.98	$188,731.64	$87,198.34	$68.40	$240,230.24
	Lanterns	$577,074.10	$387,926.78	$189,147.32	$67.22	$434,906.52
	Packs	$768,014.96	$526,788.08	$241,226.88	$68.59	$701,010.02
	Sleeping Bags	$665,818.24	$454,255.48	$211,562.76	$68.23	$462,310.80
	Tents	$2,329,489.68	$1,913,626.60	$415,863.08	$82.15	$2,136,008.34
Golf Equipment	Golf Accessories	$39,167.78	$18,713.30	$20,454.48	$47.78	$45,637.44
	Irons	$431,098.78	$228,206.50	$202,892.28	$52.94	$412,999.48

For information about how to create a crosstab report, see the Report Studio Tour.

Crosstab Nodes and Crosstab Node Members

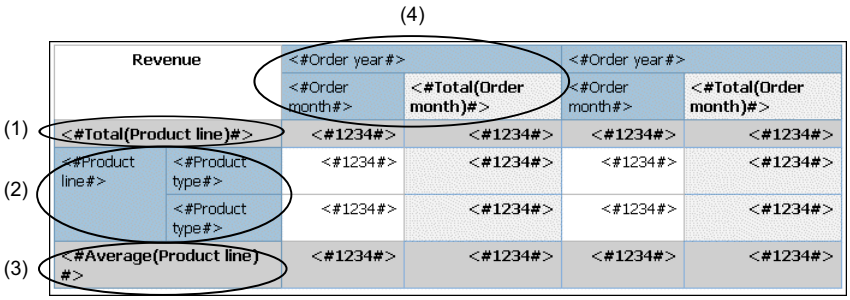
When you add data items to crosstabs, you create crosstab nodes and crosstab node members. These objects allow you to easily create crosstabs, using drag-and-drop operations.

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 that defines the members that appear in the crosstab.

The following crosstab contains four crosstab nodes.



Crosstab node 1 contains a single node member for the total. This node refers to the data item **Total(Product line)**.

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**.

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(Product line)** 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** from the **Insertable Objects** pane beneath **Product line**, **Country** becomes a new peer node to **Product line**. If the option is turned off, **Country** 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 (p. 53), turn the crosstab node creation option on. If you want the items on the edges of crosstabs to be related (contain the same nested items), turn the crosstab node creation option off.

Create a Single-Edge Crosstab Report

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

Steps

1. From the **File** menu, click **New**.
2. Click **Crosstab** and click **OK**.
3. In the **Insertable Objects** pane, on the source tab, click the data item you want to add to the crosstab and drag it to **Rows** or **Columns**.

   A black bar indicates where you can drop the data item.
4. Repeat step 3 to insert additional data items.

   If you dragged the data item in step 3 to **Rows**, drag the additional items above or below the first item.

   If you dragged the data item in step 3 to **Columns**, drag the additional items to the left or right of the first item.
5. To add measures to the crosstab, drag the measures you want to **Measures**.

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

Create a Nested Crosstab Report

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.

Rows

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.

Columns

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

- Inserting a data item to the left or to the right of a column creates a union relationship between the data item and the column.
- 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 as a nested column.

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

Steps

1. In the **Insertable Objects** pane, on the source tab, click the data item you want to add to the report.
2. Drag the data item to the location in which you want it to appear as a nested column or nested row.

   A black bar indicates where you can drop the data item.
3. Repeat steps 2 to 3 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 (p. 97).

Specify the Default Measure

Specify the default measure to be used when the measure is not specified on a row or column edge. 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 (p. 49). You then add Country under Order method. Since there is no measure specified for Country, you specify the default measure so that data is returned for each country.

Report Studio automatically sets the default measure when you insert a measure into the crosstab cells.

Steps

1. Click any part of the crosstab, and then click the select ancestor button in the title bar of the **Properties** pane.
2. Click **Crosstab**.
3. Click the **Default Measure** property and click the measure you want to use as the default measure.

Format Crosstabs

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.

You can also quickly format crosstabs by applying table styles (p. 91), and add white space to a crosstab by inserting crosstab space objects (p. 88).

Steps

1. To format the entire crosstab, do the following:
   - Click anywhere in the crosstab.
   - Click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab**.
   - In the **Properties** pane, click the property you want and then specify a value.

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

     For example, to specify a background color, click **Background Color** and choose the color you want to use.
3. To format all crosstab cells for a specific row or column, do the following:
   - Click the row or column.
   - 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 the menu option for selecting the member fact cells. The menu option indicates the crosstab cells you will be formatting. For example, if you have Product line as rows and you right-click it, the menu option will be **Fact Cells for "Product line"**.
   - In the **Properties** pane, click the property you want and then specify a value.

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

     For example, to specify a background color, click **Background Color** and choose the color you want to use.
5. To format an individual row, column, or intersection, do the following:
   - 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 you want to format.
   - In the **Properties** pane, click the property you want and then specify a value.

     For example, to specify a background color, click **Background Color** and choose the color you want to use.

Tip: In cases where fact cell formatting applied to rows conflicts with fact cell formatting applied to columns, you can use the **Fact Cells Precedence** property in the **Properties** pane to determine whether the row formatting or the column formatting has precedence. To specify this property, click anywhere in the crosstab. Then click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab**.

For more information about formatting reports, see "Formatting a Report" (p. 81).

Change a List into a Crosstab

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

Steps

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

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 (p. 97).

Example - Add Aggregate Data to a Crosstab Report

You are a report author at The Great 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.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Crosstab** and click **OK**.

5. In the **Insertable Objects** pane, on the source tab, add the following data items to the crosstab:
   - Double-click **Order method** to add it as columns.
   - Double-click **Product line** to add it as rows.
   - Click **Product type** and drag it just to the right of **Product line**.
     Product type is now nested in **Product line**.
   - Double-click **Quantity** to add it as a measure.
   - Double-click **Revenue** to add it as a second measure.

   Tip: You can find these data items in the **Orders** and **Products** folders.

6. Click any part of the crosstab and then click the select ancestor button in the title bar of the **Properties** pane.
7. Click **Crosstab**.
8. In the **Properties** pane, double-click the **Font** property.
9. In the **Size** box, click **8pt** and click **OK**.
10. Click one of the measures.
11. Click the aggregate button and click **Maximum**.

When you run the report, you can see that for personal accessories, **Knives** generated the highest sales volume for the **Special** order method, and **Watches** generated the most revenue. Watches ordered by the **Telephone** order method generated the largest revenue.

Quantity		Special		Web		Telephone		Fax		E-mail	
		Quantity	Revenue	Quantity	Revenue	Quantity	Revenue	Quantity	Revenue	Quantity	Revenue
Personal Accessories	Navigation	1,122	$128,903.66	12,394	$1,352,676.46	14,358	$1,558,740.66	2,896	$295,524.52	9,394	$1,073,282.9
	Watches	2,154	$186,322.90	20,596	$1,763,119.48	26,324	$2,336,315.08	7,014	$528,147.06	15,580	$1,382,134.8
	Knives	2,250	$110,664.58	27,646	$1,363,385.32	36,116	$1,939,505.08	6,622	$358,327.40	18,942	$1,001,273.0
	Eyewear	1,310	$121,044.48	11,926	$1,071,927.28	13,558	$1,228,356.90	2,536	$232,373.34	8,840	$786,715.6
	Binoculars	938	$115,291.98	8,774	$1,004,929.24	11,756	$1,401,917.74	2,042	$247,902.90	6,854	$798,805.5
	Maximum (Product type)	**2,250**	**$186,322.90**	**27,646**	**$1,763,119.48**	**36,116**	**$2,336,315.08**	**7,014**	**$528,147.06**	**18,942**	**$1,382,134.8**
Outdoor Protection	First Aid	1,312	$11,381.16	13,482	$106,083.68	17,146	$140,610.00	4,440	$36,000.00	11,660	$91,098.5
	Sunscreen	3,782	$17,465.08	42,894	$203,449.82	53,146	$245,641.54	11,778	$56,228.96	35,622	$165,188.5
	Insect Repellents	5,110	$31,637.28	55,908	$332,132.90	73,978	$433,721.42	14,210	$84,258.48	39,820	$231,845.2
	Maximum (Product type)	**5,110**	**$31,637.28**	**55,908**	**$332,132.90**	**73,978**	**$433,721.42**	**14,210**	**$84,258.48**	**39,820**	**$231,845.2**
Camping Equipment	Sleeping Bags	1,780	$208,640.76	22,224	$2,671,969.52	25,582	$3,179,557.22	4,030	$476,378.32	15,534	$1,921,429.7
	Tents	2,156	$768,556.90	28,658	$10,355,543.14	32,074	$12,217,810.88	7,720	$2,787,885.98	21,566	$7,643,334.9
	Packs	2,144	$314,551.30	22,356	$3,376,465.38	23,542	$3,641,360.36	5,138	$803,830.72	14,516	$2,200,011.2

Example - Create a Discontinuous Crosstab

You are a report author at The Great 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.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Crosstab** and click **OK**.

5. In the **Insertable Objects** pane, on the source tab, drag the following data items to the crosstab:
   - **Product line** from **Products** to **Rows**
   - **Order method** from **Orders** to **Columns**
   - **Revenue** from **Orders** to **Measures**

6. Pause the pointer over the query explorer button and click **Query1**.
7. In the **Insertable Objects** pane, on the toolbox tab, drag **Data Item** to the **Data Items** pane.

   The **Data Item Expression** dialog box appears.
8. In the **Expression Definition** box, type the following and click **OK**:

```
CASE
WHEN [gosales_goretailers].[Orders].[Order month] between 1 and
3 THEN 'Q1'
WHEN [gosales_goretailers].[Orders].[Order month] between 4 and
6 THEN 'Q2'
WHEN [gosales_goretailers].[Orders].[Order month] between 7 and
9 THEN 'Q3'
ELSE 'Q4'
END
```

9. In the **Properties** pane, double-click the **Name** property, rename the data item **Quarters**, and press the Enter key.

10. Pause the pointer over the page explorer button and click **Page1**.
11. In the **Insertable Objects** pane, on data items tab, drag **Quarters** to the left of **Order method**.

Tip: Ensure that the pointer is directly beside **Order method** before you drop **Quarters**. Otherwise, **Quarters** may appear as a nested row instead of a column.

12. Click **Quarters**.
13. In the **Properties** pane, double-click the **Sorting** property.
14. From the **Data Items** pane, drag **Quarters** to the **Sort List** pane.
15. Click the sort order button to sort quarters in ascending order.
16. Run the report.

All four quarters are followed by the order methods.

	Q1	Q2	Q3	Q4	E-mail	Fax	Mail
Camping Equipment	17,784,926.18	23,003,650.88	19,776,405.72	29,149,008.14	14,372,929.22	4,813,897.1	4,661,985.26
Golf Equipment	4,324,650.26	9,194,402.32	4,807,200.12	7,579,212.88	3,857,163.62	1,552,172.56	1,680,092.3
Mountaineering Equipment	3,318,640.62	6,587,232.2	4,574,665.8	6,410,811.98	3,297,131.46	1,220,329.38	973,246.5
Outdoor Protection	656,697.26	1,187,571.32	986,668.6	340,177.74	488,132.36	176,487.44	221,207.88
Personal Accessories	5,323,801.18	8,266,899.14	7,529,072.74	10,774,692.8	5,042,212	1,662,275.22	1,670,735.32

Charts

You can use Report Studio to create many chart types, including column, bar, area, and line charts.

You can also create custom charts that combine these chart types.

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.

We recommend that you work through the chart samples in the Try It Yourself appendix to familiarize yourself with some of the capabilities of charts in Report Studio. For more information, see "Try It Yourself - Create Charts" (p. 179).

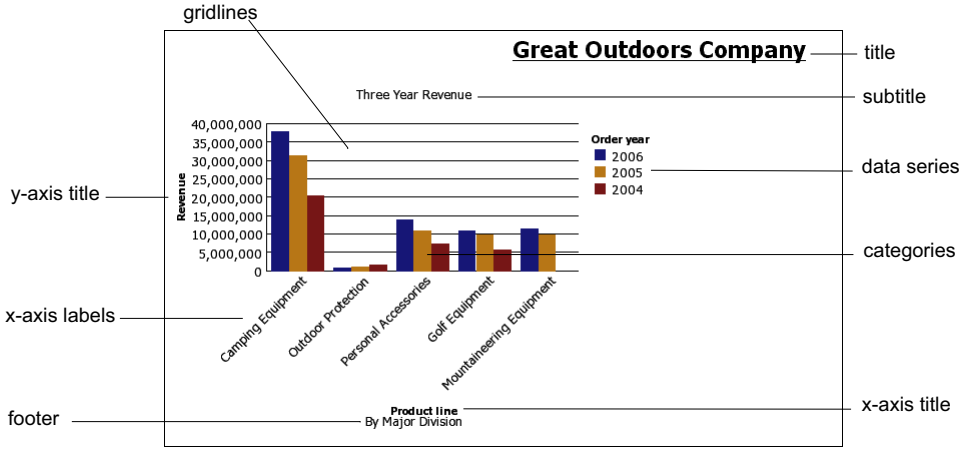
For information about different chart types, see "Chart Types" (p. 215).

You can also find information about how to create a chart in the Report Studio Tour.

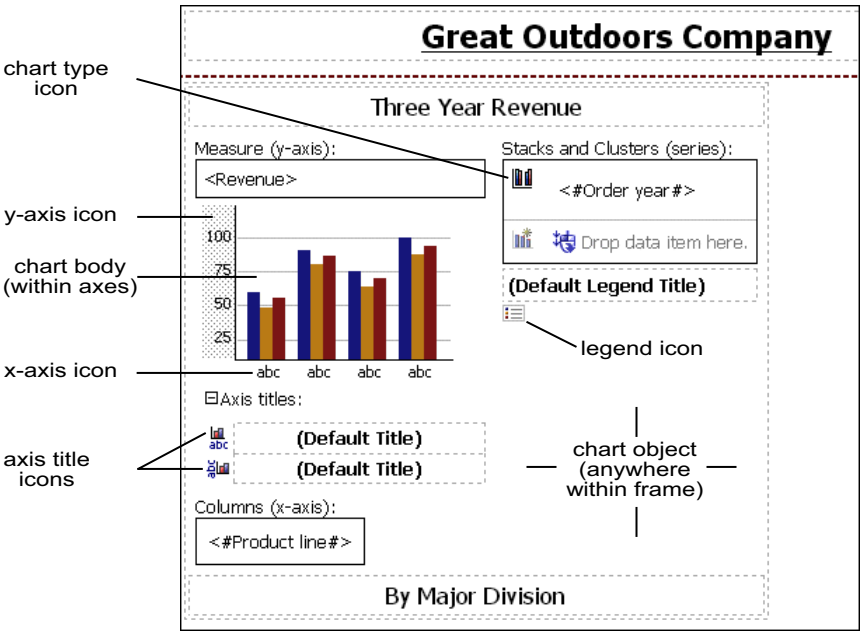
To create charts, you drag items from the **Insertable Objects** pane to the following drop zones:

- measures
- columns, bars, lines, areas, or points to represent data series
- groups, clusters, or stacks to represent categories of related data series

The following illustrates a typical chart as it appears in Cognos Viewer.



The following illustrates the same chart as it appears in the Report Studio interface. The y-axis icon is selected, and the **Axis titles** icon is expanded.



Example - Create a Column Chart to Plot Order Trends

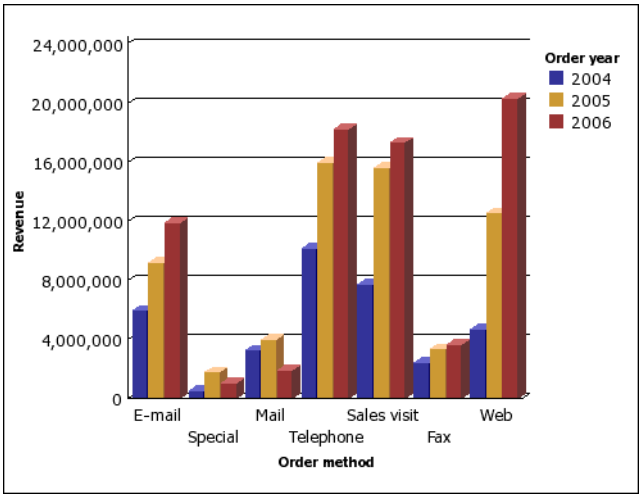
You are a report author at The Great Outdoors Company. You are requested to create a chart that shows the contribution that each order method makes to revenue. You decide to create a column chart because it emphasizes the values of each order method for each year.

Steps

1. In Cognos Connection, go to the **GO Sales and Retailers** package.
2. Click the **Report Studio** link.
   Report Studio starts.
3. From the **File** menu, click **New**.
4. Click **Chart** and click **OK**.
5. In the **Chart group** pane, click **Column**.
6. In the **Chart type** pane, click **Column with 3-D Visual Effect**.

7. Click **OK**.
8. In the **Insertable Objects** pane, on the source tab, expand **Orders** and add the following data:
   - Drag **Revenue** to the **Measure** (**y-axis**) drop zone.
   - Drag **Order year** to the **Series** drop zone.
   - Drag **Order method** to the **Categories** (**x-axis**) drop zone.

9. Click the **Order Year** icon in the **Series** drop zone, and then from the **Data** menu, click **Sort Ascending**.
10. Save the chart:
    - In the **Name** box, type
      Order Trends
    - Leave the default destination folder as **Public Folders**, and click **OK**.
11. Click the run button on the toolbar, and view the report.



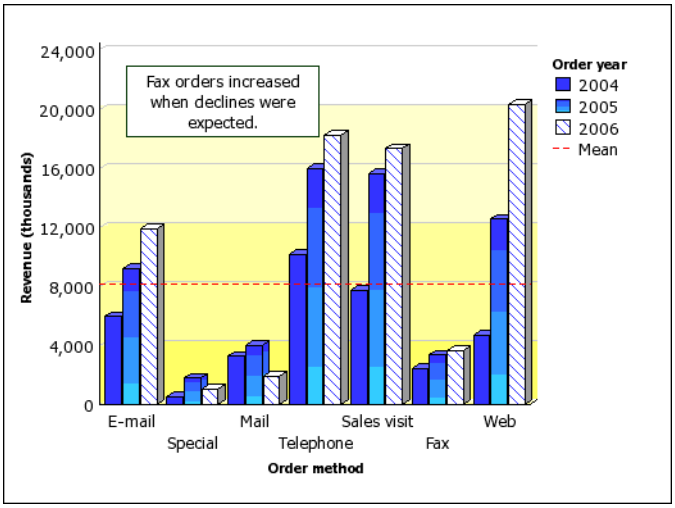
Customize a Chart

After you create a chart, you can customize it to suit your needs.

For example, the following chart was customized by:

- changing the palette
- changing the background
- adding a baseline
- adding a note
- formatting the axis values

You can make these and many other changes by changing the default properties of a chart.



You select the element in Report Studio to 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 chart to change appearance or provide information in response to expressions or conditions.

When you change a chart 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 charts. These properties are available when you select the chart object, unless specified otherwise in the **Action to perform in the Properties pane** column.

Goal	Action to perform in the Properties pane
Hide or show the title, subtitle, or footer	Under **Chart Titles**, change the **Title**, **Subtitle**, or **Footer** property.
Hide or show the legend, baselines (p. 61), markers, or notes (p. 62)	Under **Chart Annotations**, change the **Legend**, **Baselines**, **Markers**, or **Notes** property.
Hide or show the axes	Under **Axes**, change the **Y1 Axis**, **Y2 Axis**, or **Ordinal Axis** property.
Hide or show the axis title or axis line	Select the y-axis or the ordinal axis. Under **General**, change the **Axis Line** or **Axis Title** property.
Hide or show the gridlines	Select the y-axis or the ordinal axis. Under **General**, change the **Gridlines** or **Minor Gridlines** property.
Hide or show the border around the legend	Select the legend icon. Under **General**, change the **Borders** property.
Hide or show the border around the chart object	Under **Box**, change the **Border** property.
Hide or show the tooltips **Note:** Some versions of Acrobat Reader do not support tooltips.	Under **Chart Labels**, change the **Tooltips** property.
Change y-axis properties, such as range, scale interval, and so on	Select the y-axis. Under **General**, change the **Minimum Value**, **Maximum Value**, **Scale Interval**, or **Scale** property.

Goal	Action to perform in the Properties pane
Change the data format	Select the y-axis. Under **Data**, change the **Data Format** property.
Change ordinal axis properties, such as label truncation, skip interval, and so on	Select the ordinal axis. Under **General**, select the **Truncation**, **Allow Rotation**, **Allow Stagger**, or **Allow Skip** property.
Change the white space around the chart	Under **Box**, select the **Padding** or **Margin** property.
Change the color or pattern in the palette for columns, lines, and areas	Under **Color & Background**, select the **Palette** or **Conditional Palette** property.
Apply a palette to a series	Under **Color & Background**, select the **Series Color** property.
Change the default color or font (p. 60) for all chart elements	Under **Color & Background**, change the **Background Color**, **Foreground Color**, or **Fill Effects** property. Under **Font & Text**, change the **Font** or **Relative Alignment** property.
Override the default font or color for axes and chart values	Click the lock icon in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and change the **Font**, **Background Color**, **Foreground Color**, or **Fill Effects** property.
Resize the chart	Under **Positioning**, change the **Size & Overflow** property.
Change the 3-D appearance of a chart	Under **General**, change the **Depth** or **Visual Angle** property.
Insert a background image or watermark in the chart body, that is, the space between the axes	Click the lock icon in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and change the **Background Image** property.
Insert a background image or watermark in the chart object	Under **Color & Background**, change the **Background Image** property.
Go to another report	Under **Data**, change the **Drill Throughs** property.

Steps

1. Select the chart or chart element that you want to change:
   - To change general properties, such as size and color, click the chart object.
   - To change specific chart elements, such as a title or axis, click the element itself.

   Tip: To cancel a selection, press the Esc key.
2. In the **Properties** pane, click the property you want, and then make the desired changes.

   Ellipsis points (...) indicate that a dialog box provides further options.

   You may have to scroll to see all the properties.

Customize the Colors of a Chart

You can use the chart palette to control the colors of the columns, lines, or areas in a chart. For example, if you know that Telephone is the first in a data series of order methods and you want it to appear in blue, you use the palette to ensure that the first item in the series is blue.

You use a unique color, pattern, or gradient fill for each series in the chart. Patterns are useful when the report is printed on a black and white printer because colors do not print well in shades of gray.

Steps

1. Click the chart object.
2. In the **Color & Background** section of the **Properties** pane, click the ellipsis points (...) of the **Palette** property.
3. In the **Palette** dialog box, click the new palette entry button to define a new palette entry:
   - To apply color, click **Color**, specify the color properties, and click **OK**.
   - To apply a gradient, click **Gradient**, specify the **Direction**, **From color**, and **To color** properties, and click **OK**.
   - 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** drop-down list.
4. Click **OK**.
5. Repeat steps 4 and 5 for each chart series.

Tips

- To delete a palette definition, select the definition in the **Palette** box and click the delete button.
- 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.
- 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.

Example - Customize the Palette in the Order Trends Chart

You are a report author at The Great Outdoors Company. You have created a column chart that shows the contribution each order method makes to revenue.

You decide to give each column a distinctive look to better differentiate the values of each order method for each year.

Before you can try this example, you must create the chart in "Example - Create a Column Chart to Plot Order Trends" (p. 55).

Steps

1. Open the **Order Trends** chart.
2. Click the chart object.
3. In the **Color & Background** section of the **Properties** pane, click the ellipsis points (...) of the **Palette** property.
4. Click the **Color** link.
5. On the **Web Safe Colors** tab of the **Color** dialog box, click a medium blue color.
6. Click **OK**.

7. Click the new palette entry button, and then click **Gradient**.
8. In the **Direction** list, click **Down**.
9. Click the **From color** link.
10. On the **Web Safe Colors** tab of the **From Color** dialog box, click a medium blue color and click **OK**.
11. Click the **To color** link.
12. On the **Web Safe Colors** tab of the **To Color** dialog box, click a lighter blue color and click **OK**.
13. Click the new palette entry button, and then click **Pattern**.
14. In the **Pattern** box, click a diagonal line pattern.
15. Click the **Foreground color** link.
16. On the **Web Safe Colors** tab of the **Foreground Color** dialog box, click a blue color and click **OK**.

17. Click **OK** again to close the **Palette** dialog box.
    Now you must add borders to the diagonal line pattern.
18. Click the chart type icon in the **Series** drop zone.
19. In the **General** section of the **Properties** pane, change **Borders** from **Hide** to **Show**.
20. Save the chart.

Change a Chart Background

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. For more information, see "Customize a Chart" (p. 56).

Steps

1. Click the chart object.
2. Click the lock icon in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and in the **Properties** pane, under **Color & Background**, click the ellipsis points (...) of the **Fill Effects** property.
3. In the **Effect** list, choose a color, a gradient fill effect, or a pattern:
   - To apply a color, click the **Color** link 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 then 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. Click **OK**.

Tip: To remove a background fill effect, click **None**.

Example - Change the Background of the Order Trends Chart

You are a report author at The Great Outdoors Company. You created a column chart that shows the contribution that each order method makes to revenue.

You decide to give your chart a visually interesting background fill.

Before you can try this example, you must create the chart in "Example - Customize the Palette in the Order Trends Chart" (p. 59).

Steps

1. Open the **Order Trends** chart.
2. Click the chart object.
3. Click the lock icon in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and in the **Properties** pane, under **Color & Background**, click the ellipsis points (...) of the **Fill Effects** property.
4. In the **Effect** list, click **Gradient**.
5. Specify a gradient that goes from yellow to white:
   - In the **Direction** list box, click **Up**.
   - Click the **From color** link.
   - On the **Web Safe Colors** tab, click a light yellow color and click **OK**.
   - Leave white as the **To color**.
6. Click **OK**.
7. Save the chart.

Add a Baseline to a Chart

You can add one or more baselines to a chart.

Baselines are horizontal or vertical lines which 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.

You can add baselines based on

- a numeric value
- a query calculation or layout calculation
- a data minimum, maximum, mean, or percentile
- a percentage along the axis

Steps

1. Click the chart object.
2. In the **Chart Annotations** section of the **Properties** pane, click **Baselines**, and then click the ellipsis points (...).
3. Click the new button.
4. Click the desired type in the list and provide any necessary criteria, such as a value or percentage.
5. Click **OK**.

   A baseline icon appears in the **Markers, Notes, and Baselines** box.
6. To define the line style, click the baseline icon.
7. To define the text style, click the baseline text icon.
8. To add more baselines, repeat steps 1 to 6.

Tip: To delete a baseline, click its baseline icon and click the delete button.

Example - Add a Baseline to the Order Trends Chart

You are a report author at The Great Outdoors Company. You created a column chart that shows the contribution that each order method makes to revenue.

You decide to add a baseline to indicate the mean revenue for the Order Trends chart.

Before you can try this example, you must create the chart in "Example - Change the Background of the Order Trends Chart" (p. 60).

Steps

1. Open the **Order Trends** chart.
2. Click the chart object.
3. In the **Chart Annotations** section of the **Properties** pane, click **Baselines**, and then click the ellipsis points (...).
4. Click the new button.
5. Click the **Data Mean** type in the list, leave **Distance From Mean** at zero, and click **OK** twice.

   A baseline icon appears in the **Markers, Notes, and Baselines** box.
6. Click the baseline icon to define the line style.
7. In the **General** section of the **Properties** pane, change the **Line Color** to **Red**.
8. In the **General** section of the **Properties** pane, change the **Line Style** to **Dash**.
9. Save the chart.

Customize the Lines in a Chart

You can change the color, style, and weight of the lines in a chart. You can apply these changes to the x-axis and y-axis lines, as well as to the major and minor gridlines.

You can also

- specify whether to use a linear or a logarithmic scale for the y-axis

- specify a scale interval for the y-axis
- show or hide major and minor gridlines
- show regression lines on scatter charts (p. 221) and bubble charts (p. 221)

Steps

1. Click the y-axis icon or the x-axis icon of the chart.
2. In the **General** section of the **Properties** pane, click **Axis Line**, **Gridlines** or **Minor Gridlines**, then click the ellipsis points (...).
3. Click the color, weight, and style that you want.
4. Click **OK**.

Tips

- To specify a linear or logarithmic scale for the y-axis, in the **General** section of the **Properties** pane, change the **Scale** property.
- To specify a scale interval for the y-axis, in the **General** section of the **Properties** pane, change the **Scale Interval** property.
- To show minor gridlines, in the **General** section of the **Properties** pane, click **Minor Gridlines**, click the ellipsis points, select the **Show minor gridlines** check box, and type a number under **Number of minor gridlines**.
- To show regression lines, right click a scatter chart or bubble chart and, in the **Chart Annotations** section of the **Properties** pane, change the **Regression Line** property to **Show**.

Add a Marker to a Chart

Markers are notations along the axis of a chart that designate a point on the scale that may help you analyze or understand the data. The numeric position of the marker can be based on

- a numeric value
- a query calculation
- a layout calculation
- a statistical minimum
- a statistical maximum
- a data mean
- a data percentile
- a percent on the axis

Steps

1. Click the chart object.
2. In the **Chart Annotations** section of the **Properties** pane, click **Markers**, and then click the ellipsis points (...).

3. Click the new button and specify the **Numeric position type**, **Marker label**, **Marker shape**, **Numeric value**, and **Color** properties.
4. Click **OK** twice.

   The new marker appears in the **Markers, notes, and baselines** box.
5. Run the chart to view the marker.

Tip: To delete a marker, click the marker icon and text, and then click the delete button.

Add a Note to a Chart

Notes are pieces of text that appear in the chart. You can add notes to a chart to provide additional detail, such as explanatory information, or to highlight a notable value.

You can use text, an item label or value, or a report expression as the source for a note.

Notes overwrite whatever is under them. It is the responsibility of the report author to properly position them.

Steps

1. Click the chart object.
2. In the **Chart Annotations** section of the **Properties** pane, click **Notes**, and then click the ellipsis points (...).
3. Click the new button and click **OK**.

   A note icon appears in the **Markers, notes, and baselines** box.
4. Click the note icon to define size, position, and note border in the **Properties** pane.

   The location and size are statically set based on the number of pixels.
5. Double-click the note text icon to define the text.

Tip: To delete a note, click the note icon, and then click the delete button.

Example - Add a Note to the Order Trends Chart

You are a report author at The Great Outdoors Company. You created a column chart that shows the contribution that each order method makes to revenue.

You decide to add a note to draw attention to an unexpected result.

Before you can try this example, you must create the chart in "Example - Add a Baseline to the Order Trends Chart" (p. 61).

Steps

1. Open the **Order Trends** chart.
2. Click the chart object.
3. In the **Chart Annotations** section of the **Properties** pane, click **Notes**, then click the ellipsis points (...).

   The **Notes** dialog box appears.
4. Click the new button, and then click **OK**.

   A note icon appears in the **Markers, notes, and baselines** section.
5. Click the note icon.
6. In the **General** section of the **Properties** pane, change **Bottom Position** to 285 and **Left Position** to 110.

   The location and size are statically set based on the number of pixels.
7. In the **General** section of the **Properties** pane, click **Note Border**, then click the ellipsis points (...).
8. Select the **Note border** check box, click a line color and line style, and then click **OK**.
9. Run the report to view the note.

   If necessary, change the position again.
10. Double-click the note text icon and type

    Fax orders increased when declines were expected.
11. Save the chart.

Example - Format the Axis Values of the Order Trends Chart

You are a report author at The Great Outdoors Company. You created a column chart that shows the contribution that each order method makes to revenue.

You decide to format the y-axis values to make them easier to read.

Before you can try this example, you must create the chart in "Example - Add a Note to the Order Trends Chart" (p. 63).

Steps

1. Open the **Order Trends** chart.
2. Click the y-axis icon.
3. In the **Data** section of the **Properties** pane, click **Data Format**, and then click the ellipsis points (...).

4. Under **Format type**, click **Number**.
5. Under **Properties**, click **Scale**, in the drop-down list click **-3**, and click **OK**.
6. Expand **Axis titles**.
7. Double click the y-axis title icon and type
   Revenue (thousands)
8. Save the chart.

Example - Create a Drill-through Chart

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 are a report author at The Great 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 two reports, a target list report that contains the details for the item, and a source chart that shows the product line revenue.

For more information about using drill-though reporting in Report Studio, see "Set Up Drill-through Access in a Report" (p. 168).

Steps to Create the Target Report

1. In the Cognos 8 **Welcome** page, click the **Report Studio** link.
2. Select the package **GO Sales and Retailers**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.

5. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:
   - **Product line**
   - **Product type**
   - **Product name**
   - **Description**
   - **Introduction date**
   - **Product image**

   Tip: You can find these data items in the **Products** and **Orders** folders.

   Now you must create a filter to uses 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:
   [gosales_goretailers].[Products].[Product line]=?p_PL?
8. Click **OK**.
9. Save the report as **Product Line Details**.

Steps to Create the Source Chart

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. In the **Insertable Objects** pane, on the source tab, expand **Orders**.
7. Drag **Revenue** to the **Measure (y-axis)** drop zone.
8. Drag **Order method** to the **Series** drop zone.
9. Expand **Products** and drag **Product Line** to the **Categories (x-axis)** drop zone.
10. From the **Data** menu, click **Drill Behavior**.

11. In the **Basic** tab, in the **Report Drill Capabilities**, select the **Allow drill through from a package** check box.
12. Click **OK**.
13. Right-click the chart object and click **Drill Throughs**.
14. Click the **New Drill Through** button.
15. Under **Report**, select the **Product Line Details** report you created previously and click **Open**.
16. Under **Action**, select **Run the Report**.
17. Under **Format**, select **HTML**.
18. Click the edit button.

    Any existing drill-through parameters appear. You see the parameter you created for **Product Line Details**.
19. For item **p_PL**, under **Method**, click **Pass data item value**, and under **Value**, click **Product line**.
20. Click **OK** twice.
21. Save the chart as **Product Revenue**.
22. Click **Run**.

When the report is run, the list will show the product lines as clickable links. When a product line is clicked, the second report will be run for that product line.

Specify the Chart Label Text and Value Text Shown

You can specify which labels and values to use when generating the text shown on the chart.

When you show all the labels and values on some chart types, such as scatter charts, bubble charts, and polar charts, the text shown may be too long. For example, a bubble chart may show the values and the labels for the series, the categories, the x and y measures, and the bubble size.

Steps

1. Click the chart object.
2. In the **Chart Labels** section of the **Properties** pane, click **Values**, and then click the ellipsis points (...).
3. Click the labels and values that you want to show.
4. Click **OK**.

Specify the Properties of a Gauge Chart

The default gauge chart uses a band divided into thirds that runs from green to red. You can change the properties to improve the presentation of your data. For example, you may want change the colors in the band, or you may want to use a number instead of a percentage as a threshold.

Steps

1. Open the gauge chart that you want.
2. Click the chart object.
3. In the **Color & Background** section of the **Properties** pane, click the ellipsis points (...) of the **Gauge Palette** property.
4. Specify the following properties:
   - To change the color of a palette entry, under **Palette** select the color, click **Color**, specify the color properties, and click **OK**.
   - To change the boundary value of a palette entry, under **Palette** select the boundary value and specify the color properties, and click **OK**
   - To define a new palette entry, click the new palette entry button.
   - To choose a threshold style, click **Palette**, and click a style in the drop down list.
5. Click **OK**.

Tips

- 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.
- To copy and paste a gauge palette, open the gauge palette dialog box and click the copy icon to copy the palette to the clipboard. Close the palette dialog box. Select another chart, open the palette dialog box, and click the paste icon to paste the palette from the clipboard.

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, ensure that you are using a dimensionally modeled data source.

For more information about using drill-up and drill-down reporting in Report Studio, see "Create a Drill-up/Drill-down Report" (p. 172).

Steps

1. Open a chart that uses a dimensionally modeled data source.
2. From the **Data** menu, select **Drill Behavior**.
3. On the **Basic** tab, in the **Report Drill Capabilities,** select the **Allow drill up and 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 parameter 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.
4. Click **OK**.

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.

Example - Show Values in the Chart Legend

You are a report author at The Great Outdoors Company. You are requested to create a chart that shows the quantity of items sold for each product line by order method. To show how much the quantity represents in revenue, you want to show the revenue for each order method in the legend.

Steps

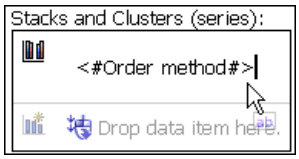
1. In Cognos Connection, go to the **GO Sales and Retailers** package.
2. Click the **Report Studio** link.

   Report Studio starts.
3. From the **File** menu, click **New**.
4. Click **Chart** and click **OK**.
5. In the **Chart group** pane, click **Column**.
6. In the **Chart type** pane, click **Column**.
7. Click **OK**.

8. In the **Insertable Objects** pane, on the source tab, expand **Orders**.

9. Drag **Quantity** to the **Measure** (**y-axis**) drop zone.
10. Drag **Order method** to the **Series** drop zone.
11. Expand **Products** and drag **Product Line** to the **Categories** (**x-axis**) drop zone.
12. From the **Structure** menu, clear **Lock Page Objects**.

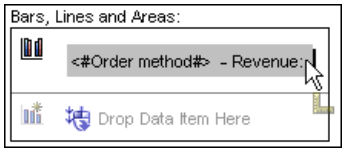
    If you do not clear **Lock Page Objects**, you will be unable to insert a text item in the legend.
13. In the **Insertable Objects** pane, on the toolbox tab, drag a text item into the legend next to **Order method**.



14. In the **Text** box, type

    - Revenue:

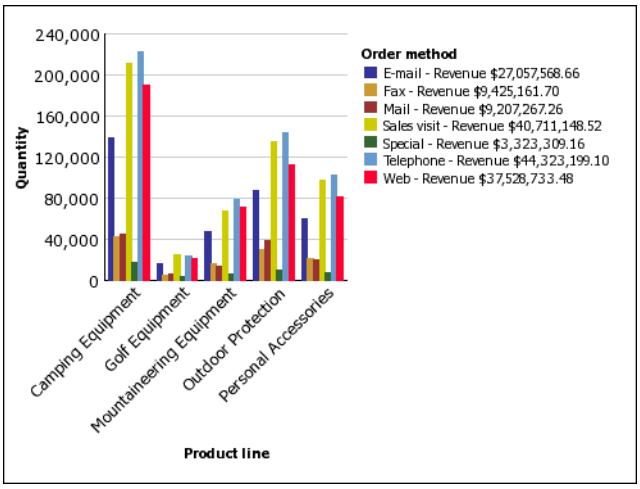
    You must insert a trailing space after the colon.
15. In the **Insertable Objects** pane, on the source tab, drag **Revenue** to the right of the text item.



16. Click the run button on the toolbar, and view the report.

The revenue is listed beside each legend item.

Tip: To reposition the legend in the chart, click the legend icon, change the **Absolute Position** property to **Yes**, and specify the **Left Position** and **Top Position** values.



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 it refers to. You must first add the desired data item to the query before you can define its property list.

For more information, see "Specify the List of Properties for a Layout Object" (p. 118).

Steps

1. Open the chart that you want to customize.
2. Click the query explorer button.
3. Click the query icon.
4. In the **Insertable Objects** pane, drag the desired item to the **Data Items** window to add it to the query.

5. Click the page explorer button to return to the chart.
6. In the **Insertable Objects** pane, drag the desired item to the layout object.
7. In the **Properties** pane, under **Data**, click the click the ellipsis points (...) to open the **Properties** dialog box.
8. Select the data item to define.
9. Click **OK**.

Example - Customize the Axis Titles

You are a report author at The Great Outdoors Company. You are requested to create a chart that shows the total revenue for the report in the horizontal axis title.

Charts contain several titles, such as axis titles, report titles and subtitles, and the legend title.

By default, the axis titles are managed for you. To customize an axis title, you drag text items or data items to the axis title area in the chart. You can use combinations of text, data items, and report expressions in titles.

Steps

1. In Cognos Connection, go to the **GO Sales and Retailers** package.
2. Click the **Report Studio** link.

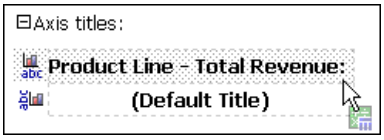
   Report Studio starts.
3. From the **File** menu, click **New**.
4. Click **Chart** and click **OK**.
5. In the **Chart group** pane, click **Column**.
6. In the **Chart type** pane, click **Column**.
7. Click **OK**.

8. In the **Insertable Objects** pane, on the source tab, expand **Orders**.
9. Drag **Revenue** to the **Measure** (**y-axis**) drop zone.
10. Drag **Order method** to the **Series** drop zone.
11. Expand **Products** and drag **Product line** to the **Categories** (**x-axis**) drop zone.
12. Expand **Axis titles**.
13. Click the horizontal axis title icon.

    Tip: You can use the **Properties** pane to change the axis title properties, such as font & text, positioning, color & background, and so on.
14. In the **Insertable Objects** pane, on the toolbox tab, drag a text item into the horizontal axis text.
15. In the **Text** box, type

    Product Line - Total Revenue:

    You must insert a trailing space after the colon.
16. In the **Insertable Objects** pane, on the toolbox tab, drag a query calculation to the right of the text.



17. In the **Create Calculation** dialog box, type

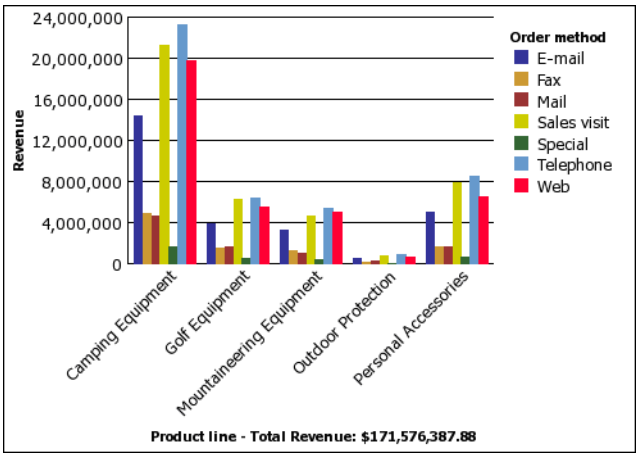
Total Revenue for Report

18. Click **OK**.
19. In the **Expression Definition** box, type the expression:

    total ([Revenue] for report)
20. Click the run button on the toolbar, and view the report.

The title shows the calculated total revenue for **Product line**.

Tip: You can use conditional formatting (p. 159) to conditionally style the titles.



Example - Create a Conditional Palette

You are a report author at The Great Outdoors Company. You are requested to create a chart that shows one pie slice for each order method. You also want to highlight the pie slice for the order method that has the highest quantity.

You create a condition that shows you which order method has sold a quantity greater than 550,000.

Steps

1. In Cognos Connection, go to the **GO Sales and Retailers** package.
2. Click the **Report Studio** link.

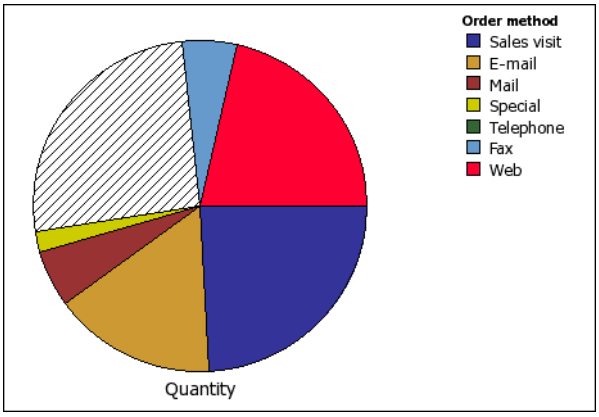
   Report Studio starts.
3. From the **File** menu, click **New**.
4. Click **Chart** and click **OK**.
5. In the **Chart group** pane, click **Pie, Donut**.
6. In the **Chart type** pane, click **Pie**.
7. Click **OK**.

8. In the **Insertable Objects** pane, on the source tab, expand **Orders**.
9. Drag **Quantity** to the **Measure** drop zone.
10. Drag **Order method** to the **Pie Slices** drop zone.

    You want to create a condition that shows you which order method has sold a quantity greater than 550,000.
11. Select the pie chart.
12. In the **Color & Background** section of the **Properties** pane, open the **Conditional Palette** properties dialog box.
13. Under **Variable**, select **<New Boolean Variable>**.
14. In the **New Variable** dialog box, type

    HighQuantity
15. Click **OK**.

16. In the **Expression Definition** box, type the expression:
    [Query1].[Quantity] > 550000
17. Click **OK** to close the expression editor. Leave the **Conditional Palette** dialog box open.
    You want to use a pattern to highlight the best performing order method.
18. Under **Effect**, click **Pattern**.
19. Under **Pattern**, click a pattern.
20. Click **OK**.
21. In the **Box** section of the **Properties** pane, change the **Borders** property from **Hide** to **Show**.
22. Run the report.

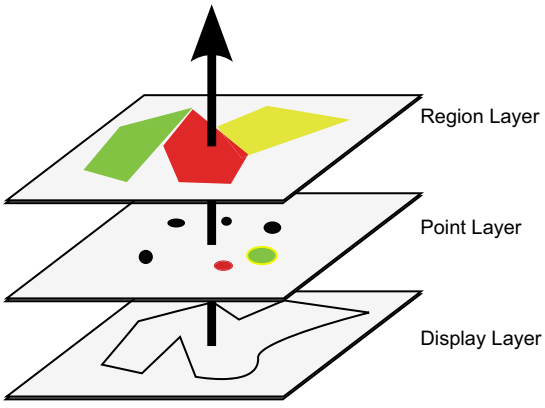


Maps

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 can be colored to represent the level of revenue. To edit the maps or create additional maps use Map Manager. For more information see "Edit a Map" (p. 78).

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.

A map in Cognos 8 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 on one layer and information related to cities on another level.



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 on a map of the world, choose Country as the region layer and then specify that the color of each country is based on the revenue value for that country. Regions 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.

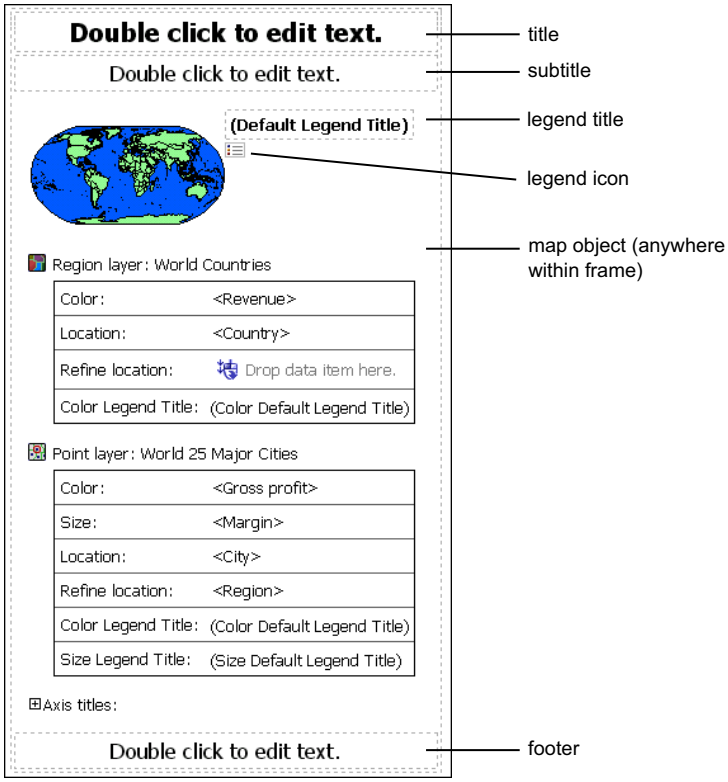
- Display layer

  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 a Map Report

The following shows the parts of a map as they appear in the Report Studio interface.



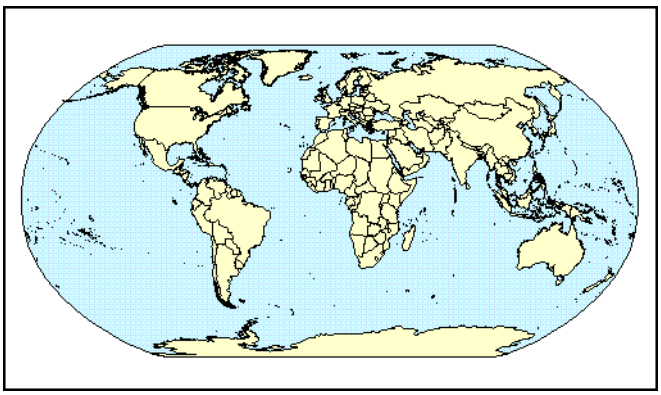
Example - Create a Map Report

You are a report author at The Great Outdoors Company. 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.

Steps

1. In Cognos Connection, go to the **GO Sales and Retailers** package.

2. Click the **Report Studio** link.
   Report Studio starts.
3. From the **File** menu, click **New**.
4. Click **Map** and click **OK**.
5. In the **Choose Map** dialog box, in the **Maps** pane, expand the folder **World** and click **World**.
6. In the **Region Layers** box, click **Countries + Territories**.
7. In the **Point Layers** box, click **None**.
8. In the **Display Layers** box, click **Oceans**.
   Tip: You can select multiple display layers or deselect a display layer by pressing Ctrl+right-click.
9. Click **OK**.
   Tip: You can return to the **Choose Map** dialog box at any time by double-clicking the map background.
10. Click the run button on the toolbar, and view the map.

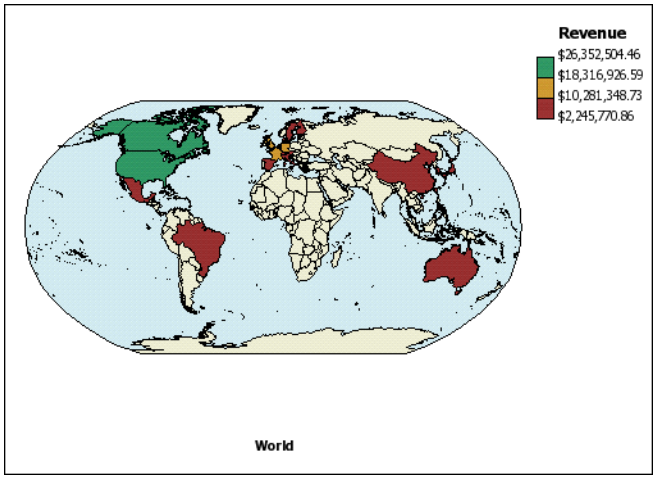


Example - Define Data Values for the Region Layer

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.

Steps

1. In the **Insertable Objects** pane, expand **Orders**.
2. Drag **Revenue** to the **Color** drop zone.
3. In the **Insertable Objects** pane, expand **Countries**.
4. Drag **Country** to the **Location** drop zone.
5. Select the map object by clicking the map background.
6. Run the report.



Ignore Data with no Features

You can specify that you do not require a match for each item in the data source. The report can then run successfully even if the map does not contain a match for each item in the data source for a particular object.

For example, an error message appears and the report does not run if you are using cities in the point layer and a city in the data source does not have the same name in the map file. If you ignore data with no features, then the data that has a match in the map file will be plotted and the data that does not have a match will be omitted.

Steps

1. Click the background of the map.
2. In the **Data** section of the **Properties** pane, click the **Ignore Data with No Features** property.
3. Set the value to **Yes**.

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 named United States and the layer in the map file labels the same country USA, then there is a mismatch that must be corrected. 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 can not 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 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*.

Steps to Create an Alias

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, click **Dictionary**, then click the ellipses points (...).

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 that you want to search the map file for.

   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 that you want to search, and click **Search**.
7. In the **Matching features** box, click the label that you want to match your data source to 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 in your data source is named USA, type USA as the alias.

   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 in your data source. For more information see "List Reports" (p. 43).

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.

Steps to Set the Color and Size of Points

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 that you want shown on the map.

   For example, on a map of the world you may want the points to represent cities.
3. Click **OK**.
4. In the **Insertable Objects** pane, drag an object 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 **Insertable Objects** pane to the **Color** drop zone.
5. In the **Insertable Objects** pane, drag an object to the **Size** drop zone in the **Point Layer**.

Steps to Set the Location of Points

1. In the **Insertable Objects** pane, drag an object 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 is not.
2. 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 Another Color to the Region or Point Layer

You can add colors for regions or points and specify values to determine when those colors are shown.

Steps

1. In the report, click the **Region Layer** or **Point Layer**.
2. In the **Color & Background** section of the **Properties** pane, click **Palette**, and then click the ellipses points (...).

3. Click the new button, and click **Color**.
4. A new color is added to the list of 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.
7. Click **OK**.

Specify the Size of Points

The size of the points used on a map is defined in the **Properties** pane.

Steps

1. In the point layer, click the measure in the **Size** drop zone.
   For example, if the measure is Revenue, click it.
2. In the **General** section of the **Properties** pane, click **Minimum Size**, and select or type the minimum point size for points on the map.
3. Click **Maximum Size**, and select or type the maximum point size for points on the map.

Add Titles, Subtitles, Footers, and Axis Title

You can add map titles, subtitles, and footers. In addition, the axis title is shown below the map. By default, the axis title shows the name of the map specified in the map file.

Steps

1. Select the map object.
2. In the **Chart Titles** section of the **Properties** pane, click the **Title, Subtitle, Footer,** or **Axis Title** property and click **Show**.
3. Double-click in the box that appears on the report and type the text that you want.
   By default, the axis title uses the name of the map.

Add Legend Titles

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.

Steps to Change the Legend Title for the Entire Map

1. If the legend title is not showing, click the legend icon. In the **General** section of the **Properties** pane, set the **Legend Title** to **Show**.
2. In the report, double-click the legend title and type the title you want.

Steps to Change the Legend Title for the Region Color, Point Color, or Point Size

- By default the legend titles are taken from the object you've selected from the data source. To change a title, do one of the following:
  - In the **Insertable Objects** pane, 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.

Steps to Add a Note

1. Select the map object.
2. In the **Chart Annotations** section of the **Properties** pane, click **Notes**, and then click the ellipsis points (...).

3. Click the new button, and click **OK**.
   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, click the **Text** property, and then click the ellipsis points (...).
6. Type the text that you want to appear in the note.

Steps to Position a Note and Add a Border

1. In the report, click the note icon.
2. 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.
3. Use the **Note Border** property to specify a border around the note.
4. Run the report to view the note.
   If necessary, change the position again.

Customize a Map

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.

Goal	Action to perform in the Properties pane
Hide or show the title, subtitle, footer, or axis title	Under **Chart Titles**, change the **Title**, **Subtitle**, **Footer**, or **Axis Title** property.
Hide or show the legend	Under **Chart Annotations**, change the **Legend** property.
Hide or show map labels	Select the region or point layer. Under **Chart Labels**, change the **Labels** property.
Hide or show values on the map	Select the region or points layer. Under **Chart Labels**, change the **Values** property.
Hide or show the border around the legend	Select the legend icon. Under **Box**, change the **Borders** property.
Change the border around the map object	Under **Box**, change the **Border** property.
Hide or show the tooltips **Note:** Some versions of Acrobat Reader do not support tooltips.	Under **Chart Labels**, change the **Tooltips** property.
Change the amount of white space around the map	Under **Box**, change the **Padding** or **Margin** property.
Change the default colors for all map elements	Under **Color & Background**, change the **Background Color**, **Foreground Color**, or **Fill Effects** property.

Goal	Action to perform in the Properties pane
Change the font and the alignment of text	Under **Font & Text**, change the **Font** or **Relative Alignment** property.
Resize the map	Under **Positioning**, change the **Size & Overflow** property.
Change the font for the legend	Select the legend icon. Under **Font & Text**, change the **Font** property.
Change the format of values in the legend	Select the value in the region or point layer. Under **Data**, change the **Data Format** property.

Steps

1. Select the map object or map element that you want 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, click the property you want, and then make the desired changes.

   Ellipsis points (...) indicate that a dialog box provides further options.

   You may have to scroll to see all the properties.

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.

Steps to Set Up a Filter in the Target Report

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 you want to use.

   For example, if you want the target report to open when Canada is clicked in the source report, expand **Countries** and double-click **Country**.
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, if are using the GO Sales and Retailers package and want the report to open when Canada is clicked in the source report, the expression would be as follows:

   ```
   [gosales_goretailers].[Countries].[Country]='Canada'
   ```
6. Save the target report.

Steps to Set Up a Drill-Through Link in the Source Report

1. Open the source report.
2. Select the region layer or the point layer.
3. In the **Data** section of the **Properties** pane, click **Map Drills**, and then click the ellipsis points (...).
4. In the **Map Drills** dialog box, click the new button.
5. In the **Search String** box, type the name of the feature you want to use for the drill-through link, select it from **Matching Features** box, then click **OK**.

   For example, if you want a report to open when you click on Canada, search on all or part of the word **Canada** then select **Canada**.
6. In the **Drill Throughs** dialog box, click the new button.
7. Click the ellipses points (...) beside the **Report** box, and select the target report.

8. Click **OK**.
9. Run the report.

When you click the selected feature in the source report, the target report will open.

Edit a Map

Administrators and modelers use a Windows utility named Map Manager to import maps and update labels for maps in Report Studio. For map features such as country and city names, administrators and modelers can define alternative names to provide multilingual versions of text that appears on the map.

With Map Manager you can edit the labels in maps to be consistent with the object names in your data base.

For instructions, see the Map Manager *Installation and User Guide*.

Additional Maps

Additional maps are available from the Cognos Global Customer Services Web site. Much of the mapping data on the Web site is derived from MapInfo Corporation, the Cognos preferred provider for location intelligence. MapInfo provides a wide variety of global data including

- detailed accurate roads
- political, postal, and census boundaries
- industry-specific data, such as communication system boundaries, insurance risk related data, and business prospect data

The Cognos Global Customer Services Web site provides a list of data providers and descriptions of the map data that Cognos provides as part of Cognos 8.

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 Cognos 8. MapInfo provides solutions that can be directly integrated with Cognos 8. 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.

Business application	Benefits
Target marketing	Learn who your best clients are and find more like them.
Network optimization and site location analysis	Put stores near your customers and look for gaps in geographical coverage.
Routing and work force optimization	Reduce the number of trucks you need and make your drivers more efficient.
e-government	Provide citizens with self-service opportunities.
Sales territory creation	Create balanced sales territories.
Economic development	Plan the development of your community.
Communications network planning	Avoid costly mistakes by putting cell towers in the right locations. Identify the locations of clients in your service area.

You can contact MapInfo for both data and location intelligence solutions through their Web site: www.mapinfo.com.

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 you want 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.

Convert a List into a Repeater

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

Steps

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 Great Outdoors Company, which sells sporting equipment. You are requested to create mailing labels for all of the company's retailers.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Repeater Table** and click **OK**.
5. Click the repeater, click the select ancestor button in the **Properties** pane title bar, and click **Repeater Table**.
6. In the **Properties** pane, specify properties for the repeater:
   - For the **Across** property, type **2**.
   - For the **Down** property, type **5**.
   - Double-click the **Table Properties** property, select the **Fixed size** check box, and click **OK**.
7. In the **Insertable Objects** pane, on the toolbox tab, drag the **Table** object to the repeater.

   The **Insert Table** dialog box appears.
8. In the **Number of columns** box, type **1** and click **OK**.
9. 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 **Browse**, click **logo.jpg**, and click **OK**.
   - In the **Position** box, click the align top right button.
   - In the **Tiling** box, click **Do not tile** and click **OK**.
   - Double-click **Size & Overflow**, and in the **Height** box, type **175**, and click **OK**.
10. Click the table, click the select ancestor button in the **Properties** pane title bar, and click **Table**.
11. In the **Properties** pane, specify properties for the table:
    - Double-click **Border**.

- In the **Color** box, click **Black**.
- In the **Style** box, click **Solid line**.
- In the **Width** box, click **1 pt**.
- Under **Preview**, click the all borders button and click **OK**.
- Double-click **Font**, and under the **Size** box, click **8pt**, and click **OK**.

12. In the **Insertable Objects** pane, on the toolbox tab, drag the **Block** object to the repeater 8 times to create 8 blocks.

13. Drag the **Text Item** object to the first block:
    - In the **Text** dialog box, type **To:** and click **OK**.
    - Click the text item.
    - Click the font button, click **Bold**, and click **OK**.
14. Click the first block, and, in the **Properties** pane, specify properties for the first block:
    - Double-click **Padding**, type **35** in the right box, click **mm** as the unit, and click **OK**.
    - Click **Horizontal Alignment**, and click **Center**.

15. In the **Insertable Objects** pane, on the source tab, expand **Retailers**.
16. Expand the **Mailing address** folder, and drag the seven data items to the remaining seven blocks.
17. Shift+click the seven blocks to select them, and in the **Properties** pane, specify properties for the seven blocks:
    - Double-click **Padding**, type **25** in the left box, click **mm** as the unit, and click **OK**.
    - Click **Horizontal Alignment** and click **Left**.

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

Chapter 3: Formatting a Report

Format your report to make it more readable and to reflect company standards. When you format a report in Report Studio, the formatting is stored in a layout.

You can also format a report based on conditions (p. 159).

When formatting a report, you can

- add a header or footer
- add borders
- add text
- specify the font
- add color
- insert an image
- add a bookmark
- insert other objects
- align an object
- use tables to control where objects appear
- apply padding
- set margins
- reuse a layout object
- add a page
- reorder columns
- rename a column
- swap columns and rows
- set object properties
- set report properties
- create and modify classes

Recommendation - 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.

The challenge in laying out a report is in performing a mental translation of a desired layout to the layout objects available in Report Studio. Use the following steps to help you perform this translation:

❑ 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 is similar to the page header, except the information 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 (p. 88).

❑ 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 (p. 91).

❑ Decide which data frames to use to lay out the data.

Choose a list, crosstab, chart, repeater, or text frame.

Report Layout Guidelines

To work efficiently in Report Studio, follow these guidelines when formatting reports:

- 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.

For more information about laying out reports using these guidelines, see "Try It Yourself - Create an Invoice" (p. 195).

The Page Structure View

When you add objects to a report, you usually work in the layout. 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.

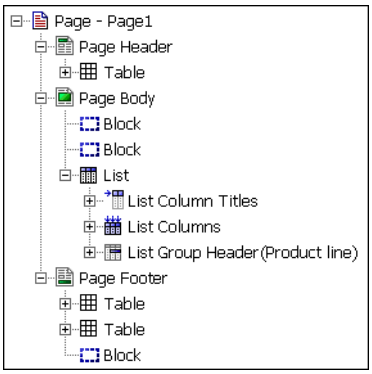
- to quickly move objects from one area of a page to another

  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**.



Add a Header or Footer

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.

Pages

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.

Lists

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.

Type	Description
List page header	Adds a header that appears at the top of the list on every page in which list data appears.
Overall header	Adds a header that appears once at the top of the list.
Group or section header	Adds a header that appears for each group of a grouped column (p. 43) or each section (p. 107).
Group or section footer	Adds a footer that appears for each group of a grouped column (p. 43) or each section (p. 107).
Overall footer	Adds a footer that appears once at the bottom of the list.
List page footer	Adds a footer that appears at the bottom of the list on every page in which list data appears.

In addition, you can modify list headers and footers. You can

- split a header or footer into multiple cells that span the list (**Structure** menu, **Split List Row Cell**)
- merge multiple cells (**Structure** menu, **Merge List Row Cells**) in a header or footer
- add rows (**Structure** menu, **Insert List Row Cells Above** and **Insert List Row Cells Below**) to a header or footer

You can also add section headers and footers (p. 107), and you can add footers by adding a summary (p. 108).

Steps

1. To add a page header or footer, from the **Structure** menu, click **Page Header & Footer**, select the appropriate check boxes, and click **OK**.
2. To add a list header or footer, from the **Structure** menu, click **List Headers & Footers**, select the appropriate check boxes, and click **OK**.
3. To add objects to a header or footer, drag the object that you want from the **Insertable Objects** pane 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 (p. 94).

Add Borders

You can add borders to objects in a report such as a column, a header, a footer, or to the whole report to improve appearance.

Steps

1. Click the object to which you want 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.
3. Select the values you want 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.
4. 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 button.
5. Click **OK**.

Add Text

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 (p. 163).

Steps

1. In the **Insertable Objects** pane, click the toolbox tab.
2. Drag the **Text Item** object to the report.

   The **Text** dialog box appears.
3. Type the text that you want.

   Tip: You can also paste text from another part of the report.
4. Click **OK**.

   If you click **OK** without typing any text, Report Studio inserts the string **Double click to edit text**.

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.

Specify the Font

Specify the font in which you want text in a report to appear.

Steps

1. Click the object for which you want to specify the font.

   Tip: To specify the default font for the report, click the page.
2. Click the font button.
3. Specify the font properties that you want.

   Tip: Type a list of fonts 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**, 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.
4. Click **OK**.

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 Cognos 8 tools. You can modify default values by modifying classes (p. 101).

Add Color

You can add background and foreground color to objects in the report.

Steps

1. Select the object to which you want to add color.

   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, select **Background Color** or **Foreground Color**, and then click the ellipsis points (...).
3. To apply an existing color, click the **Web Safe Colors** tab or **Named 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 in hexadecimal.
5. Click **OK**.

Insert an Image

You can insert an image in a report. You can insert images in other objects, such as a block or table cell, directly in the report page, or as the background image of another object.

Steps

1. In the **Insertable Objects** pane, click the toolbox tab.
2. Drag the **Image** object to the report and then select it.
3. In the **Properties** pane, double-click the **URL** property.
4. In the **Image URL** dialog box, type the URL of the image you want to insert or click the **Browse** button 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, see "Troubleshooting" (p. 205).
5. Click **OK**.

Insert a Background Image

You can insert a background image for objects in a report. For example, use a background image to add a watermark to a page.

Steps

1. Select the object for which you want to add a background image.
2. In the **Properties** pane, double-click the **Background Image** property.
3. In the **Image URL** box, type the URL of the image you want to insert or click the **Browse** button to go to the location containing the image.

   The image appears in the **Preview** box.
4. In the **Position** box, choose how you want to align the image in the object.
5. In the **Tiling** box, click the tiling option you want.
6. If the object has scroll bars and you do not want the background image to move while scrolling, select the **Keep background from scrolling** check box.
7. Click **OK**.

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 use bookmarks so that users can move to specific rows.

You can also drill through to another report (p. 168).

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.

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.

Steps

1. In the **Insertable Objects** pane, click the toolbox tab.
2. Drag **Bookmark** to the location you want, which can be anywhere in the report.
3. Click the bookmark, and in the **Properties** pane, click the **Source Type** property and then click a source type.

Source type	Description
Text	Creates a static bookmark using a text value. For example, a list has sections, and you want users to jump from each section to the top of the report.
Report Expression	Creates a dynamic bookmark whose values are derived from an expression that you define.
Data Item Value	Creates a dynamic bookmark that has data as possible values. This is useful for creating a context-based bookmark. 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.
Data Item Label	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. 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.
Member Caption	In a crosstab, creates a dynamic bookmark that uses member captions as possible values.
Cell Value	In a crosstab, creates a dynamic bookmark that uses cell values as possible values.

4. Click the property that appears under the source type you chose and specify the bookmark value.

   For example, if the source type is **Data Item Value**, click the **Data Item Value** property and click the data item you want to use.
5. Right-click the object that you want to use to jump to the bookmark and click **Drill Throughs**.

   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.

6. Click the new drill through button.
7. Click the **Bookmark** tab.

8. Click **Source type** and click one of the source types described in step 4.

   Click the source type you want 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.
9. If you clicked one of the following source types, specify the value to use to jump to the bookmark.
   - For **Text**, click the ellipsis points (...) next to **Text**, and type a text value.
   - For **Data Item Value** or **Data Item Label**, click the **Data item** list, and choose a data item.
   - For **Report Expression**, click the ellipsis points (...) next to **Report expression**, and define the expression.
10. Click **OK**.

Example - Add a Table of Contents to a Report

You are a report author at the Great Outdoors Company, which sells sporting equipment. You are requested to create a report that shows all products sold by the company. To make the report easier to read, you divide the report into sections for each product line. You add bookmarks so that users can jump to each product line and back to the top of the report.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.

4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:
   - **Product line**
   - **Product type**
   - **Product name**
   - **Description**
   - **Production cost**

   Tip: You can find these items in the **Products** folder.
6. Click the **Product line** column, and then click the create section button.

7. From the **Structure** menu, click **Lock Page Objects**.
8. In the **Insertable Objects** pane, on the toolbox tab, drag **Bookmark** to the left of **Product line**.
9. Click the bookmark, and in the **Properties** pane, click the **Source Type** property and click **Data Item Value**.

10. In the **Properties** pane, click the **Data Item Value** property and click **Product line**.
11. In the **Insertable Objects** pane, on the toolbox tab, drag **List** to the report header.
12. Click the report header.
13. Click the container alignment button, and click align top left.
14. In the **Insertable Objects** pane, on the source tab, drag **Product line** to the new list.

15. Right-click **Product line** and click **Drill Throughs**.
16. Click the new drill through button.
17. Click the **Bookmark** tab.
18. Click **Source type** and then click **Data Item Value**.
19. Click **Data Item** and then click **Product line**.

20. Click **OK**.
21. In the **Insertable Objects** pane, on the toolbox tab, drag **Text Item** to the left of the **Product line** bookmark.
22. In the **Text** dialog box, type **Top** and click **OK**.

23. Right-click the **Top** text item and click **Drill Throughs**.
24. Click the new drill through button.
25. Click the **Bookmark** tab.
26. Click **Source Type** and then click **Text**.
27. Click the ellipsis points (...) next to the **Text** box.
28. Type **Top** and click **OK**.
29. In the **Insertable Objects** pane, on the toolbox tab, drag **Bookmark** to the report header.
30. Click the bookmark.
31. In the **Properties** pane, click the **Source Type** property and click **Text**.
32. In the **Properties** pane, double-click the **Label** property and type
    Top
33. Click **OK**.
34. Save the report.
35. In Cognos Connection, click the run with options button for the report.
36. Under **Delivery**, click **Save the report**.
37. Click **Run** and then click **OK**.
    Tip: The report may take a few minutes to run.
38. Under **Actions**, click **View the output versions for this report** button for the report.
39. Under **Formats**, click **HTML** to view the report output.
    The report opens in Cognos Viewer.

In the report header, all product lines appear in a list as links. Clicking a product line brings you to the corresponding product line section in the second list. You can return to the top of the report by clicking the **Top** link next to the section heading.

Product line
Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories

Top Camping Equipment

Product type	Product name	Description	Production cost
Cooking Gear	TrailChef Canteen	Aluminum canteen. Rugged fleece-lined cover with belt clips, removable shoulder sling and small pocket for water purification tablets bottle. Holds 2 liters	$9.22
Cooking Gear	TrailChef Cook Set	All you will ever need on the trail. Pot gripper and nylon carrying bag included. 1.5 and 2 liter pots with individual lids. Weight: 750 g.	$34.97
Cooking Gear	TrailChef Cup	Tin cup. Holds 0.4 liters. Weight: 60 g	$5.00

Insert Other Objects

In addition to text and images, the toolbox tab in the **Insertable Objects** pane contains other objects that you can add to the report layout.

Object	Description
Query Calculation	Adds a calculated column (p. 113).

Object	Description
Layout Calculation	Adds a calculation in the layout that contains run-time information, such as current date, current time, and user name.
Block	Adds an empty block, 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.
Table	Adds a table, a container in which you can insert other objects. This is useful for controlling where objects appear.
Hyperlink	Adds a hyperlink so that users can jump to another place, such as a Web site.
Row Number	Numbers each row of data returned when the report is run. **Note:** You can add row numbers only to lists and repeaters.
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 (p. 213), or JavaScript. HTML items appear only when you run the report in HTML format. **Note:** You cannot include <form> tags in HTML items. When you upgrade to the next version of Cognos 8, the report upgrade processes do not account for the use of undocumented and unsupported mechanisms or features such as JavaScript that refers to 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 support only a restricted set of well-formed XHTML.
Layout Component Reference	Adds a reference to another object. Useful when you want to reuse an object.
Conditional Blocks	Adds an empty block that you can use for conditional formatting (p. 159).
Field Set	Adds an empty block that has a caption. This is similar to the **Block** object, but with a caption.
Hyperlink Button	Adds a hyperlink in the form of a button.
Metric Studio Diagram	Adds a Metric Studio history chart as an image. For information about adding a Metric Studio diagram to a report, see the Metric Studio *User Guide*.
As of Time Expression	Adds the As of Time expression (p. 118).
Crosstab Space	Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge.

Object	Description
Crosstab Space (with Text)	Inserts a cell with a text item on a crosstab edge. Allows for the insertion of non-data cells on an edge.

The toolbox tab contains other objects when you work in Query Explorer (p. 128) and in Condition Explorer (p. 158). You can also add your own objects to the toolbox, such as a company logo that you use often. For more information, see the *Administration and Security Guide*.

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 *Administration and Security Guide*.

Step

- To add an object, drag or double-click it from the toolbox tab.

For information about each report type, see "Types of Reports" (p. 43). For information about each prompt control and prompt buttons, see "Adding Prompts" (p. 119).

Elements Supported in Rich Text Items

Rich text items support all XHTML character entities, such as   (non-breaking space), and the following elements:

- div
- span
- ul
- ol
- 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.

```
<div style="font-size:14pt; text-decoration:underline">List:</div>
<ul style="list-style-type:circle">
   <li style="color:green">Item <span style="font-weight:bold">A</span></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 Great Outdoors Company, which sells sporting equipment. You want to insert a 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.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the top right corner of the page.
3. In the **Welcome** dialog box, click **Open an existing report**.
4. In the **Open** dialog box, click the **GO Template Samples** folder and then double-click **GO Cover Page**.

   The GO Cover Page report opens.

5. In the **Insertable Objects** pane, on the toolbox tab, drag the **HTML Item** object to the report.
6. In the report, select the **HTML Item** you just added.

7. In the **Properties** pane, double-click the **HTML** property.
8. In the **HTML** dialog box, type the following:

```
<OBJECT classid="CLSID:6BF52A52-394A-11D3-B153-00C04F79FAA6">
    <PARAM NAME="URL" VALUE="/c8/webcontent/samples/images/GO.wmv"/>
</OBJECT>
```

9. Click **OK**.

When you run the report, the multimedia file plays in Windows Media Player.

Align an Object

You can specify an alignment for objects in a report to determine where it appears.

Tables can also be used to determine where objects appear in a report.

Steps

1. Select the object that you want to align.

   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. Click the container alignment button, and then click one of the available alignment options.

Use Tables to Control Where Objects Appear

In addition to the container alignment button, 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.

Steps

1. In the **Insertable Objects** pane, click the toolbox tab.
2. Drag **Table** to the report.

   The **Insert Table** dialog box appears.
3. In the **Number of columns** and **Number of rows** boxes, type the number of columns and rows you want the table to have.
4. To have the table span the width of the report page, select the **Maximize width** check box.
5. To add borders to the table, select the **Show borders** check box.
6. Click **OK**.

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.

Steps

1. Click the table, list, or crosstab to which you want to apply a table style.
2. From the **Table** menu, click **Apply Table Style**.
3. In the **Table styles** box, click the table style that you want.

   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, select the **Set this style as the default for this report** check box to set the style as the default for all lists and crosstabs.

   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 tables styles cannot be set as the default.
6. Click **OK**.

Apply Padding

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.

Steps

1. Select the object to which you want 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 you want.

   Note: Specifying padding using % as the unit of measure is not supported when producing reports in PDF.
4. Click **OK**.

Set Margins

You can set margins for objects in a report.

Steps

1. Select the object for which you want to set margins.
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 you want.

   Note: Specifying margins using % as the unit of measure is not supported when producing reports in PDF.
4. Click **OK**.

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.

Steps

1. Click the object that you want 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, in the **Name** property, type a value beginning with a letter to uniquely identify the object and press the Enter key.

3. To reuse the object in another report, open that report.
4. In the **Insertable Objects** pane, on the toolbox tab, drag the **Layout Component Reference** object to the location where you want to reuse the object.
5. In the **Component Location** box, do the following:
   - Click **This report** to reference an object in the current report.
   - Click **Another report** to reference an object in another report, click the ellipsis points (...), and open the report you want.

6. In the **Available components to reference** box, click the object you want and click **OK**.
7. Click the referenced object.
8. In the **Properties** pane, click the **Embed** property and 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.
   - 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.

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 you want 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.

Steps

1. In the parent object that you want to reuse, click the child object you want to change.
2. In the **Properties** pane, in the **Name** property, type a value beginning with a letter to uniquely identify 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 **Component Overrides** property.
5. In the **Component Overrides** dialog box, select the child object you want to change and click **OK**.

   The child object in the copy of the parent object is replaced by the following text:

   Drag & drop object to override component child.
6. Drag the object that you want 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.

Steps

1. Open the report that you want 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 the **Reload Components** button 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 the **Update All Component Copies** button to refresh all copied objects.
6. Click **Close**.

Add a Page

Reports may contain pages with a variety of content. Pages may be added to create a multiple-page report. You may want to do this for a variety of reasons. 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. The following pages contain data.

You can also create page sets to control the rendering of pages in a report.

Steps

1. Pause the pointer over the page explorer button and click **Report Pages**.

   Tip: Click **Prompt Pages** to add a prompt page (p. 119).
2. In the **Insertable Objects** pane, on the toolbox tab, drag **Page** to the work area.
3. To change the order of pages, click a page and drag it above or below the other existing pages.
4. To associate a query to the page, in the **Properties** pane, click the **Query** property and click the query you want.

   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 (p. 82).
6. Add the objects you want to the page.

Create Page Sets

Create 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.

Steps

1. Pause the pointer over the page explorer button and click **Report Pages**.
2. Create the pages you want to appear in the report.
3. In the **Insertable Objects** pane, drag the **Page Set** object to the **Report Pages** pane.
4. In the **Properties** pane, click the **Query** property and click the query you want to associate to the page set.
5. Organize the pages in the report by dragging report pages to the page set you want.
6. Insert the page containing details in the **Detail Pages** folder.

   Detail pages are the pages that repeat based on the items you group by in the following step.

   Tip: You can insert multiple detail pages into the same page set, and link them using a master-detail relationship (p. 175).
7. Define the grouping structure for the page set:
   - Click the page set.
   - In the **Properties** pane, double-click the **Grouping & Sorting** property.
   - In the **Data Items** pane, drag the data item you want to group by to the **Groups** folder in the **Groups** pane.

   - To sort the data (p. 106) within each group, in the **Data Items** pane, drag the data item you want to sort by to the **Sort list** folder, and then click the sort order button to specify the sort order.
   - Click **OK**.
8. Repeat steps 2 to 6 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 occur within pages containing product line information.

Note: Grouping an item for a page set is not the same as grouping a column in the layout (p. 43). 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.

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

Steps

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 5 to 7 to create other links.

   Tip: To delete a link, click it and press the Delete key.
8. Click **OK**.

For more information about master-detail relationships, see "Create a Master-Detail Relationship" (p. 175).

Example - Preparing a Product List Report

You are a report author at the Great 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.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.

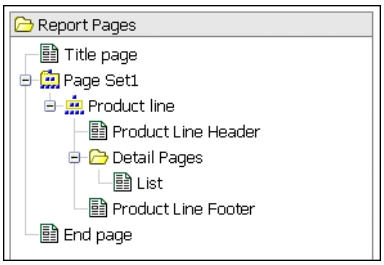
4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:

- **Product line**
- **Product type**
- **Product name**
- **Description**
- **Production cost**

Tip: You can find these data items in the **Products** folder.

6. Group the **Product line** and **Product type** columns.
7. Pause the pointer over the page explorer button and click **Report Pages**.
8. In the **Insertable Objects** pane, drag **Page Set** to the work area and associate it to the query **Query1**.
9. In the **Insertable Objects** pane, drag **Page** to the work area four times.
10. For each page, click the page, and in the **Properties** pane, double-click the **Name** property and name it as follows:
    - **Title page**
    - **End page**
    - **Product Line Header**
    - **Product Line Footer**
11. For each page just created, add the objects you want.

    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.



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

Reorder Columns

Change the order of columns in a report to rearrange information.

Steps

1. Click the column you want to reorder in the report.

You can click the column heading or a column.

2. Drag the column to a new location.

   A black bar indicates where you can drop the column.

Rename a Column

Change the column title to provide a more meaningful name. By default, when you run a report, the column title is taken from one of the following:

- if defined, the **Label** property of the data item
- if defined, the label of the data item in the model
- the **Name** property of the data item

Consequently, the column title you see in the layout may not be the column title you see when you run the report. For example, if you modify the **Name** property of the data item, the column title in the layout will change to the new name. However, when you run the report, the column title that appears will be, if defined, the data item label in the report or the data item label in the model. If neither of these are defined, only then will the modified name be used as the column title.

Steps

1. Click the column heading you want to change.
2. In the **Properties** pane, click the **Source Type** property and click the source type you want to use for the column title:
   - To use a static text item, click **Text**, and then type the name you want in the **Text** property that appears.
   - To use a data item value, click **Data Item Value**, and then click the data item you want to use in the **Data Item Value** property that appears.

     Using a data item value is useful when you have a master-detail relationship. For example, you create a list with sections using a master-detail relationship (p. 107), and you want the section name to appear in the column title of one of the columns. Insert the data item used as section headings into the column title, and specify the source type as **Data Item Value**. When you run the report, for each section, the column title will contain the section heading combined with the data item label or name.
   - To use a label that is different from the label in the model, click **Data Item Label**, click the data item you want to use in the **Data Item Label** property that appears, and type the name you want in the **Label** property.

     Tip: The **Label** property is under the **Data Item** group.
   - To use a report expression, click **Report Expression**, double-click the **Report Expression** property that appears, type the expression you want, and click **OK**.
3. Press Enter.

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.

Step

- From the **Structure** menu, click **Swap Rows and Columns**.

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

Setting Object Properties

For each object you insert in a report, you can define its properties by specifying values in the **Properties** pane. Some properties, such as borders, color, and padding, are discussed elsewhere. Other properties include

- pagination
- spacing and breaking
- text flow and justification
- size and overflow
- floating
- table properties

Some properties apply only to some objects. For more information about objects and properties, see "Report Studio Object and Property Reference" (p. 351).

Many object properties are Cascading Style Sheets (CSS) attributes. For information about a particular property that is a CSS attribute, see the CSS documentation on the World Wide Web Consortium (W3C) Web site (http://www.w3.org).

Note: Some properties do not apply to all types of report output. For example, the **Direction** text flow property works with HTML output but does not apply to PDF output.

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.

Option	Description
Keep with header	Keeps all headers on the same page with the number of detail rows specified.
Keep with footer	Keeps all footers on the same page with the number of detail rows specified.
Reset page count	Resets the page count after a page break to the value specified.
Reset page number	Resets the page number after a page break to the value specified.
Repeat every page	If the report renders multiple pages, this object is repeated on every page.
Allow text to break across pages	Allows text in a row to break across pages.
Repeat table rows on page break	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.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Pagination** property.
3. Specify how you want pages to break and be numbered by specifying the options that you want.

Specify Text Properties

You can specify text properties by choosing any of these options.

Option	Description
Line Height	Sets the distance between lines of text in an object.
Letter Spacing	Sets the amount of additional space between letters in an object.
Text Indent	Sets the indentation of the first line of text in an object.
Word Break	Sets line-breaking behavior within words.
Break Words when Necessary	Sets whether to break words when the content exceeds the boundaries of an object.
Enforce Stricter Line-Breaking Rules for Japanese Text	Sets line-breaking rules for Japanese text.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Spacing & Breaking** property.
3. Specify the text properties that you want.

Specify Text Flow

You can specify text flow properties by choosing any of these options.

Option	Description
Direction	Sets the reading order of an object, such as right to left.
Writing mode	Sets the direction and flow of content in an object.
Bi-directional	Sets the level of embedding in an object.
Type	Sets the type of alignment used to justify text in an object.
Kashida space	Sets the ratio of kashida expansion to white space expansion when justifying lines of text in the object. This property is used in Arabic writing systems.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Text Flow & Justification** property.
3. Choose how you want text to flow by specifying the options that you want.

Specify the Height and Width

You can specify the height and width of objects. In addition, if the object is a block, you can specify how to handle content overflow. Specify the height and width by choosing any of these options.

Option	Description
Height	Sets the height of the object.
Width	Sets the width of the object.
Content is not clipped	If the contents of the block exceed the height or width of the block, the block automatically resizes to fit the contents.
Content is clipped	If the contents of the block exceed the height or width of the block, the content is clipped. **Note:** The clipped content still exists. It is just not visible in the block.
Use Scrollbars only when necessary	If the contents of the block exceed the height or width of the block, scrollbars are added to the block.
Always use scrollbars	Scrollbars are added to the block.

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.

In addition, when you use a percentage to specify the size of an object, the percentage is relative to the object's parent. There are some cases where 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.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Size & Overflow** property.
3. Specify the options that you want.

Control How Objects Flow Around Other Objects

You can control how objects flow around other objects by choosing any of the following options.

Option	Description
Float	Sets how other objects flow around the object.
Allow floating objects on both sides	Allows other objects to flow on both sides. The Float property must be set.
Move below any floating object on the left side	If there are other objects to the left of the object, the object moves below those objects. The Float property must be set.
Move below any floating object on the right side	If there are other objects to the right of the object, the object moves under those objects. The Float property must be set.
Move below any floating object	Moves the object under any other object in which the Float property was set.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Floating** property.
3. Specify how you want other objects to flow around the object by specifying the options that you want.

Specify Borders and Cell Size in Tables

You can specify the borders and cell size in lists, crosstabs, and tables by choosing any of these options.

Option	Description
Collapse borders	Collapses adjacent cell borders into a single border.
Cell spacing	If borders are not collapsed, sets the amount of space between cell borders.
Show empty cell borders	If borders are not collapsed, shows borders of cells that are empty.
Fixed size	Makes all cells the same fixed, size.

Steps

1. Click the object you want.
2. In the **Properties** pane, double-click the **Table Properties** property.
3. Specify the borders and cell size by specifying the options that you want.

Setting Report Properties

You can change the default report property settings.

- To preserve the original appearance of reports created in ReportNet, use the **Use 1.x report style** option.

  For more information, see "Create and Modify Classes" (p. 101).
- For reports with multiple query frames, to render the default number of rows of each query frame on each HTML page, use the **Page break by data container for interactive HTML** option.

  For more information, see "Setting Object Properties" (p. 98).

Create and Modify Classes

Create your own classes or modify existing classes in a report to format objects across a report according to your particular needs. In 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 its parent objects.

Classes you create or modify can be applied only to the current report. To create or modify classes for all reports, you must modify the default layout style sheet. In addition, some classes can be used to format Query Studio reports.

Steps

1. To work with classes that were used in Cognos ReportNet(R), do the following:
   - From the **File** menu, click **Report Properties**.

- Select the **Use 1.x report styles** check box.

Use 1.x report styles when you are working with reports created in ReportNet and you want to preserve their original appearance.

2. Pause the pointer over the page explorer button and click **Classes**.
3. To create a new class, in the **Insertable Objects** pane, drag **Class** to the **Local Classes** pane.
4. To modify an existing class, in the **Global Class Extensions** pane, click the class you want.

   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 global classes to make the same change to more than one global class.
5. 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.

   A diamond symbol appears on the global class icon to indicate that the class was modified.
6. Apply the class to the objects you want:
   - Pause the pointer over the page explorer button, and click a report page.
   - Click an object to which you want to apply the style.
   - In the **Properties** pane, double-click the **Class** property.
   - To not apply any class, click the **None** radio button.
   - To apply a local class, click the **Local Class** radio button and then click the class you want.
   - To apply a global class, click the **Global Class** radio button and then click the class you want.
   - Click **OK**.

Modify Classes to Format Query Studio Reports

Some global classes are specific to Query Studio or can be applied to Query Studio reports. You can modify the following classes to format Query Studio reports.

Class	Description
List column body cell	Style used to format list data.
List column title cell	Style used to format list column headings.

Steps

1. Create a Query Studio template (p. 40).
2. Modify the global classes that you want.
3. Save the template.

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*.

Modifying the Default Layout Style Sheet

In addition to creating or modifying classes in a report, you can create and modify classes that will apply to all reports. Default styles are stored in a style sheet named GlobalReportStyles.css. For information about modifying the style sheet, see the *Administration and Security Guide*.

Chapter 4: Working with Data

Make a report easier to read and easier to understand by specifying what data appears in the report and how the data looks. In Report Studio, you can

- filter data
- sort data
- create sections
- add a summary
- create a calculation
- format data
- specify the list of properties for a layout object
- show data for a specific time period
- define a prompt
- work with queries

Filter Data

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

If an expression is used in multiple reports, or by different report authors, ask your modeler to create the expression as a standalone object in the model and include it in the relevant package.

For information about creating filters in the package, see the Framework Manager *User Guide*.

If you are working with a dimensional data source, you can also define slicers. If you are filtering non-numeric data, you must select members from the data tree instead of typing member names (p. 234). If you are filtering a multiple-fact query, you should understand the differences between using a detail filter or summary filter (p. 129).

Steps

1. To add a filter that was created in the package, in the **Insertable Objects** pane, on the source tab, drag the filter you want to the report.

   You must drag the filter to an object that is linked to a query. If there is more than one query defined in the report, drag the filter to an object linked to the query you want.

2. 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. If there is more than one query defined in the report, click an object that is linked to the query you want.

   Tip: You can also define filters in Query Explorer (p. 128).

3. In the **Filters** dialog box, decide what type of filter you want 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 **Available Components** box, define the filter expression:
   - To include data items not shown in the report, on the source tab, double-click data items.
   - To include 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.

- To include data items from a specific query (p. 128) in the report, on the queries tab, double-click data items.
- To to add functions, summaries, and operators to the filter expression, on the functions tab, double-click elements.

  Note: You can insert only functions that return a boolean value. 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.
- To include a value that is derived from a parameter, on the parameters tab, double-click a parameter.

  Parameters are used to define prompts (p. 119), drill-through reports (p. 168), and master-detail relationships (p. 175).

Tip: You can also type the filter expression directly in the **Expression Definition** box. When typing date values, the date must be in the YYYY-MM-DD format.

For more information about creating expressions, see "Using the Expression Editor" (p. 233).

6. Click the validate expression button and click **OK**.
7. In the **Usage** box, specify whether the filter is required, optional, or not to be used.
8. If you created a detail filter, in the **Application** box, click one of the following options:
   - To apply the filter to individual records in the data source, click **Before auto aggregation**.

     For example, you want to filter out individual orders of a specific product type within a product line 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 product types within a product line that generated less than ten thousand dollars in revenue.

   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 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.

   Tip: If you are using a dimensional data source, after aggregation is always applied, because by definition, dimensional data is aggregated.
9. If you are filtering a summary, click the ellipsis points (...) under **Scope** and select the grouping level for which you want 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.
10. Click **OK**.

Define a Slicer

Use slicers to create dimensional filters that reduce the data included in measure rollups. For example, a slicer is a filter that is applied to the cells but not the rows or columns in a crosstab.

A slicer member set is an expression that returns a set of members from the same dimension. You can create multiple slicer member sets to filter across two or more dimensions. You cannot create multiple slicer member sets against the same dimension.

You can add a child member of a report item as a slicer. For example, you create a crosstab with Product line as rows, Year as columns, and Revenue as the measure. You drag Sleeping Bags to the **Slicer** pane. When you run the report, all product lines appear in the crosstab, but a value appears only for the Camping Equipment row because Sleeping Bags is a child of Camping Equipment.

Steps

1. Pause the pointer over the query explorer button and click the query to which you want to add a slicer.
2. Choose whether to create a slicer with one or more than one member:
   - To create a slicer with a single member, in the **Insertable Objects** pane, on the source tab, drag the member that you want to the **Slicer** pane.
   - To create a slicer with more than one member, on the toolbox tab, drag **Slicer Member Set** to the **Slicer** pane. Then, drag the members that you want 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:

     ```
     set([Fax],[Telephone])
     ```
3. To add members from another dimension, repeat step 3 to create a separate slicer member set.

Filtering Data Using an SAP BW Data Source

If you are working with an SAP BW data source, you must consider additional things 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, although the aggregated values are correct. Siblings are characteristic values with the same parent of the filtered member.

Each level in an SAP BW hierarchy has an item with the same name as the level, and has 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 is 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.

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 in the day the order was booked. If the application defaulted the time component as it stored rows, the query 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 dates defaulting the time can be dangerous if the application changes and starts to capture actual times. To avoid this problem, you can

- truncate 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 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 specifying **Short** for the **Date Style** property.

You can then apply filters to the DATEONLY column that would return the correct results. If you create a parameter in 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 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 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 filter to ignore the time. In the 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)
```

Sort Data

Organize data values in a report by specifying a sort order. Data can be organized in ascending or descending order based on the values in any data item.

If you are using a dimensional data source, you cannot sort data items from different dimensions that are intermixed. For example, you cannot sort on Data_Item1_Dimension1, Data_Item1_Dimension2, Data_Item2_Dimension1. If you do, an exception error occurs.

For SAP BW, each level in a hierarchy has an item with the same name as the level, and has a role of _businessKey. Such items are known as level identifiers.

For SAP BW, only the level identifier of all the time-related characteristics, such as 0CALDAY and 0CALMONTH, should be used 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.

Steps

1. Click the data item on which you want to sort.

2. From the **Data** menu, click **Sort Ascending** or **Sort Descending**.

An up arrow or down arrow appears beside the data item to indicate that a sort order has been set.

Tip: To remove a sort order, click the column, and from the **Data** menu, click **Don't Sort**.

When you specify a sort order for more than one column, the columns are sorted in the order that 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, followed by B and then C. You can change the order in which the columns are sorted.

Perform Advanced Sorting

You can perform advanced sorting in a report. For example, 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.

Steps

1. Click a column.

   In a crosstab, click the row or column for which you want to perform advanced sorting.
2. From the **Data** menu, click **Advanced Sorting**.
3. If you are sorting a list and you want to sort a column within a group, do the following:
   - In the **Groups** pane, under the **Groups** folder, expand the folder of the grouped column (p. 43) that you want.
   - In the **Data Items** pane, drag the data items you want to sort to the **Sort List** folder.

     Tip: You can also drag data items from the **Detail Sort List** folder.

   - Click the sort order button to specify ascending or descending order.
4. If you are sorting a list and you want 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 you want to sort on 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.
6. Click **OK**.

Create 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 (p. 43). The difference is that section headers appear outside the list, crosstab, chart, or repeater. In addition, you can group data items only in lists.

Steps

1. Click the column that you want to show as 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.

   Sections are created, and a section header appears. The data container that contains the column used to create sections is embedded in a list.

3. To add or remove section headers and footers, from the **Structure** menu, click **List Headers & Footers**, select or clear the appropriate check boxes, and click **OK**.

   The section header and footer check boxes appear under **Overall header**.

Remove Sections

To remove sections and reinsert the data item used to create the sections to the data container, do the following:

Steps

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. In the **Insertable Objects** pane, click the data items tab.
8. Drag the data item that was used as a section header back into the data container.

9. 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.

For information about the page structure view, see "The Page Structure View" (p. 82).

Summarizing Data

In a report, there can be detail and summary aggregation. The detail aggregation specifies how a data item is totaled at the lowest level in a report. In lists, the detail aggregation specifies how the values that appear in the rows are totaled. In crosstabs, the detail aggregation specifies how the values in the cells are totaled. For example, the 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.

Product line	Product type	Revenue
Camping Equipment	Cooking Gear	$5,401,132.08
	Tents	$47,456,718.10
	Sleeping Bags	$11,688,944.32
	Packs	$14,627,996.26
	Lanterns	$10,539,200.16
Golf Equipment	Putters	$2,310,367.24
	Golf Accessories	$811,516.72
	Irons	$8,801,464.98
	Woods	$13,982,116.64

The summary aggregate specifies how data items are totaled in the headers and footers of a list and in the total rows and columns of a crosstab.

Revenue	2004	2005	2006	Total(Order year)
Camping Equipment	$20,471,328.88	$31,373,606.46	$37,869,055.58	**$89,713,990.92**
Golf Equipment	$5,597,980.86	$9,598,268.88	$10,709,215.84	**$25,905,465.58**
Mountaineering Equipment		$9,642,674.54	$11,248,676.06	**$20,891,350.60**
Outdoor Protection	$1,536,456.24	$988,230.64	$646,428.04	**$3,171,114.92**
Personal Accessories	$7,144,797.52	$10,955,708.04	$13,793,960.30	**$31,894,465.86**
Total(Product line)	**$34,750,563.50**	**$62,558,488.56**	**$74,267,335.82**	**$171,576,387.88**

OLAP Data Sources

Summary options depend on the data source type 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 aggregate that is used is specified in the data source itself. As a result, you cannot specify a different detail aggregation in a report. The report always renders the type of aggregate that is specified in the data source.

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 a data item represents, see "Add Data to a Report" (p. 31).

You can specify detail and summary aggregation in different ways by using any of the following:

- aggregation properties that are specified in the model (p. 109).
- the **Auto Group & Summarize** property (p. 109).
- the aggregate button in the toolbar (p. 109).
- aggregation properties for a data item (p. 110).
- aggregate functions in data item expressions (p. 110).
- the aggregation mode (p. 111).
- the solve order (p. 114).

Use the Aggregation Properties Set in the Model to Specify Detail or Summary Aggregation

When working with relational or Dimensionally Modeled Relational (DMR) data sources, you can use the aggregation properties specified for the query item in the 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 Report Studio (p. 230).

Set the Auto Group & Summarize Property

Set the query property **Auto Group & Summarize** to specify whether 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.

Steps

1. Pause the pointer over the query explorer button and click the query that you want.
2. In the **Properties** pane, set the **Auto Group & Summarize** property to the value that you want:
   - 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**.

Use the Aggregate Button to Add Simple Summaries

You can add simple summaries to the groups in a report by using the aggregate button in the toolbar. This button provides a subset of the aggregate functions available in Report Studio. In addition, a **Custom** option is available so that you can add your own aggregate function in the expression of the data item (p. 110).

The aggregate button sets the rollup aggregate property (p. 110) for the data item to the selected summary aggregate, 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.

Steps

1. Open the report that you want.
2. Click the column to which you want to add a summary.
3. From the **Data** menu, click the kind of summary that you want (p. 227).
4. If you want to change the summary label, do the following:
   - Click the label.
   - In the **Properties** pane, click the **Source Type** property and then click the source type that you want to use to define the label.

     For example, click **Data Item Value** to produce a dynamic label for the summary based on data item values.
   - Depending on the source type you chose in the previous step, click the property below **Source Type** and specify the label.

     For example, if clicked **Data Item Value** as the source type, click the **Data Item Value** property and click the data item that you want to use to define the label.

In lists, the summary appears as a footer (p. 82). 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 (p. 48).

Tips

- If you want to change a summary, click it, and in the **Properties** pane, under **Data Item**, click **Rollup Aggregate Function** (p. 110) 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.

Specify Aggregation Properties for a Data Item

For each data item in a report, you can specify detail and summary aggregation properties so that you can easily manage summaries without having to create complex data item expressions (p. 110).

Steps

1. Click the data item for which you want to set detail or summary aggregation.
2. In the **Properties** pane, set the **Aggregate Function** or the **Rollup Aggregate Function** property to the function that you want (p. 227).

Use Aggregate Functions in Data Item Expressions

You can use aggregate functions in data item expressions. The aggregate functions in the Expression Editor (p. 233) that have the same name as the aggregate functions available through the **Aggregate Function** and **Rollup Aggregate Function** properties (p. 110) operate the same way. For example, setting the **Rollup 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 aggregate 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 an aggregate 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 aggregate functions.
- You want to use aggregate functions that require more than one parameter, such as percentile.
- You require aggregate expressions that are not available in the aggregation properties, such as a `for` clause.

For example, your report uses Product line, Product type, and Quantity. You want to calculate the percentage of the parent quantity 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 aggregates at different group levels. You use aggregate functions in the data item expression with a `for` clause to specify the group level, as follows:

total([Quantity] for [Product type]) / total([Quantity] for [Product line])

Steps

1. Click the data item that you want.
2. In the **Properties** pane, double-click the **Expression** property.
3. Add the aggregate functions that you want to the expression for the data item.

Specify the Aggregation Mode

If you are working with a dimensional or a Dimensionally Modeled Relational (DMR) data source, you must specify the aggregation mode that you want Report Studio to use when aggregating data in crosstabs and charts (p. 111). The aggregation mode sets the aggregation clause to be used in the data item expression.

Steps

1. From the **Tools** menu, click **Set Options**.
2. Click the **Edit** tab.
3. Click **Aggregation mode** and select the aggregation mode that you want to use.

Aggregation mode	Description
Within detail	Aggregates the visible details
Within aggregate	Aggregates the visible aggregates at the next lower level of detail
Within set	Aggregates the member sets

4. Click **OK**.

Aggregating Values in Crosstabs and Charts

In crosstabs and charts, aggregated values are calculated using one of the following `aggregate` expressions:

- `aggregate(`*`measure`*` within set `*`set expression`*`)`
  Aggregates the member values from the data source within the current content.
- `aggregate(`*`measure`*` within detail `*`data item`*`)`
  Aggregates the lowest level of details in the report.
- `aggregate(`*`measure`*` within aggregate `*`data item`*`)`
  Aggregates each level of details in the report.

You decide which `aggregate` expression is used by setting the aggregation mode (p. 111).

For example, in the following crosstab, if you specified **Total** as the summary, these aggregate expressions are produced for each aggregation mode:

		<#Revenue#>	<#Revenue#>
<#Quarter#>	<#Month#>	<#1234#>	<#1234#>
	<#Month#>	<#1234#>	<#1234#>
<#Total(Quarter)#>		**<#1234#>**	**<#1234#>**

- 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.
- Total ([Revenue] within aggregate [Quarter])

  This expression totals the month values visible in the report at the intersecting product line into quarters. At the bottom right corner, it does the same, but starting with the aggregate over all product lines for each month.

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 month and quarter values are used. The result is the same.

		Revenue
2005 Q 3	2005/Sep	$5,300,599.08
2005 Q 4	2005/Oct	$11,778,348.92
	2005/Nov	$3,353,399.48
	2005/Dec	$4,645,069.28
2006 Q 1	2006/Jan	$3,119,227.22
	2006/Feb	$3,045,478.88
	2006/Mar	$6,867,919.18
2006 Q 2	2006/Apr	$8,306,586.26
	2006/May	$5,129,112.24
	2006/Jun	$6,927,447.24
2006 Q 3	2006/Jul	$5,078,789.96
	2006/Aug	$5,180,166.38
	2006/Sep	$4,892,742.38
2006 Q 4	2006/Oct	$15,887,237.28
	2006/Nov	$3,609,599.42
	2006/Dec	$6,240,652.88
Total(Quarter)		**$171,576,387.88**

Different results appear when you start filtering, changing aggregation types, or use set expressions or unions.

For example, if you replace month with an expression that returns only the first month in each quarter (January, April, July, and October), then the visible values you see in the crosstab are for only those months. If you then total the quarter values from the data source using the `within set` aggregation mode (total ([Revenue] within set [Quarter]), the result will include all months.

		Revenue
2004 Q 1	2004/Jan	$1,446,711.10
2004 Q 2	2004/Apr	$3,919,558.66
2004 Q 3	2004/Jul	$2,790,291.12
2004 Q 4	2004/Oct	$6,528,354.60
2005 Q 1	2005/Jan	$2,588,270.16
2005 Q 2	2005/Apr	$7,736,095.72
2005 Q 3	2005/Jul	$4,669,535.84
2005 Q 4	2005/Oct	$11,778,348.92
2006 Q 1	2006/Jan	$3,119,227.22
2006 Q 2	2006/Apr	$8,306,586.26
2006 Q 3	2006/Jul	$5,078,789.96
2006 Q 4	2006/Oct	$15,887,237.28
Total(Quarter)		**$171,576,387.88**

If you total the visible month values using the `within detail` aggregation mode (total [Revenue] within detail [Quarter]), then you are totaling the values from only the first month in each quarter. Consequently, you get different results.

		Revenue
2004 Q 1	2004/Jan	$1,446,711.10
2004 Q 2	2004/Apr	$3,919,558.66
2004 Q 3	2004/Jul	$2,790,291.12
2004 Q 4	2004/Oct	$6,528,354.60
2005 Q 1	2005/Jan	$2,588,270.16
2005 Q 2	2005/Apr	$7,736,095.72
2005 Q 3	2005/Jul	$4,669,535.84
2005 Q 4	2005/Oct	$11,778,348.92
2006 Q 1	2006/Jan	$3,119,227.22
2006 Q 2	2006/Apr	$8,306,586.26
2006 Q 3	2006/Jul	$5,078,789.96
2006 Q 4	2006/Oct	$15,887,237.28
Total(Quarter)		**$73,849,006.84**

In most cases, you should use the `within detail` aggregation mode because the results are easiest to understand, and they are the same as for footers in a grouped list report. In more complex cases, you may consider the `within aggregate` aggregation mode. The `within set` aggregation mode should be reserved for reports with a purely dimensional focus. For example, when there are no detail or summary filters defined in the report.

Create a Calculation

Create a calculated column to make a 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.

If an expression is used in multiple reports, or by different report authors, ask your modeler to create the expression 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" (p. 94).

Steps

1. In the **Insertable Objects** pane, click the toolbox tab.
2. To create a calculation that will be applied to data, do the following:
   - Drag **Query Calculation** to the report.

     The **Create Calculation** dialog box appears.
   - In the **Name** box, type a name for the calculation and click **OK**.
3. To create a calculation in the layout that contains run-time information, such as current date, current time, and user name, drag **Layout Calculation** to the report.
4. In the **Available Components** box, define the calculation:

   - To include data items not shown in the report, on the source tab, double-click data items.
   - To include 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.
   - To include data items from a specific query (p. 128) in the report, on the queries tab, double-click data items.
   - To add functions, summaries, and operators to the filter expression, on the functions tab, double-click elements.

   - To include a value that is derived from a parameter, on the parameters tab, double-click a parameter.

     Parameters are used to define prompts (p. 119), drill-through reports (p. 168), and master-detail relationships (p. 175).

     Tip: You can also type the expression directly in the **Expression Definition** box.

     For more information about creating expressions, see "Using the Expression Editor" (p. 233).
   - To include data items not shown in the report, on the source tab, double-click data items.

5. Click the validate expression button and click **OK**.

Multiple 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 for the 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 is 0. In crosstabs, if more than one data item has the same solve order, column items are calculated first and then row items.

For example, a crosstab contains the columns Actual revenue, Sales target, and a third column that calculates the percentage by which actual revenue exceeds target revenue. The crosstab contains a row for each Sales 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, Cognos 8 calculates the column percentage first, and the summary row is calculated 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:

	Actual revenue	Sales target	*Percentage by which Actual Exceeds Target*
2004	$43,025,310.66	$15,051,551.00	*186%*
2005	$86,096,447.76	$29,408,080.00	*193%*
2006	$102,253,419.64	$36,068,756.00	*183%*
Total(Sales year)	**$231,375,178.06**	**$80,528,387.00**	**187%**

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:

- outermost calculation on the columns
- innermost calculation on the columns
- outermost calculation on the rows
- innermost calculation on the rows

If there are solve order ties, column items are calculated before row items, and outer nesting items before inner nesting items.

Resolve Multiple Calculations for Crosstabs and Charts

If an intersection point in a crosstab or chart contains a value that is derived from multiple calculations (p. 114), 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 Aggregation Function** property to **Calculated** (p. 227) 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 Aggregation Function** property set to **Calculated**, the normal rules for solve order precedence apply.

Solve Order in Upgraded Reports

In previous versions of Cognos 8, you could specify the solve order for objects such as crosstab node members (p. 48). Solve order is now specified for data items. When you upgrade a report, solve orders specified in the report are moved to data items.

Steps

1. Click the data item for which you want 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 data container, and you want the data item to have a different solve order in each container, you must use a separate data item.

   Tip: In crosstabs, to specify the solve order for the default measure (p. 50), click the crosstab and, in the **Properties** pane, click the select ancestor button and click **Crosstab**.

2. In the **Properties** pane, type an integer value in the **Solve Order** property.

   To specify the solve order for the default measure, click **Default Measure Solve Order**.

Resolve Multiple Calculations when Querying Dimensional Data Sources

Dimensional data sources may include data source calculations. Because Cognos 8 cannot predict the presence of such calculations, specifying the **Solve Order** property may be unable to resolve the solve order correctly. You can suppress potentially meaningless results when querying dimensional data sources that include data source calculations.

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 Cognos 8 cannot determine the presence of data source calculated members ahead of time, it cannot determine the correct solve order automatically.

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.

Steps

1. Pause the pointer over the query explorer button and click the query that contains the data item for which you want to suppress values.
2. In the **Data Items** pane, click the data item.
3. In the **Properties** pane, in the **Calculation Intersection** property, click **Show "Not applicable"**.

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.

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 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.

You can also format data based on a condition (p. 159) or specify the format for a particular object.

When using a Cognos cube as a data source, mixed currency values use the asterisk character (*) as the unit of measure. Mixed currency values occur when you calculate values with different currencies.

The same behavior occurs for SAP BW data sources.

Steps

1. From the **Data** menu, click **Layout Data Format**.
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. In the **Currencies** dialog box, select the currencies you want and click **OK**.

   For example, you may have one column whose values are in euros and another column whose values are in US dollars.
4. Set the properties that you want.

   If you added currencies in step 4, click each one and set the properties you want. 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 required depend on the authoring language specified for the report. For more information, see "Using Patterns to Format Data" (p. 437).

   If you specify 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 only be changed with caution.
5. Click **OK**.

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 you are working in is applied to values for which no format is specified.

Specify the 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.

Steps

1. Click the object.
2. In the **Properties** pane, double-click the **Data Format** property.
3. Under **Format type**, click the format type you want 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.
5. Click **OK**.

Specifying the Number of Decimals

When specifying the number of decimals, Cognos 8 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. That is,

- if the digit immediately after the precision to be displayed is greater than 5, the number is rounded up.
- if the digit immediately after the precision to be displayed is less than 5, the number is rounded down.
- if the digit immediately after the precision is a 5, then the number is rounded down when the preceding digit is even and rounded up when the preceding digit is odd.

For example, if you specify no decimals, 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.

Locale-sensitive Properties

Report Studio contains an extensive library of customized properties adapted to users from different regions who speak different languages. If, for example, 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 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**

Specify the List of Properties for a Layout Object

Specify the list of properties 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 in its definition. If the data item does not appear in the list, you must reference it in order to make the layout calculation work.

Steps

1. Click the layout object that you want.

   Tip: For a list of layout objects for which you can specify a list of properties, see the **Properties** property in "Report Studio Object and Property Reference" (p. 351).
2. In the **Properties** pane, double-click the **Properties** property.
3. Select the data items you want and click **OK**.

Show Data for a Specific Time Period

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.

Steps

1. In the **Insertable Objects** pane, on 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 that you want and click **OK**.

   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 that you want to use.

   If you do not specify a time, the default time 12:00:00.000 AM is used.

Adding Prompts

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, which are based on parameterized filters, form the questions you want to ask users. Prompt controls provide the user interface in which the questions are asked. Finally, parameter values provide the answers to the questions.

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

You can also create prompts in the package. For more information, see the Framework Manager *User Guide*.

Use the Build Prompt Page Tool

Use the **Build Prompt Page** tool to quickly add prompts to a report. Report Studio creates the prompts for you in a default prompt page.

Steps

1. Click the column you want users to be prompted on.

   To create multiple prompts or a cascading prompt, click more than one column.
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** prompt buttons

You can add more objects or modify existing elements. For example, you can change the prompt control chosen by Report Studio.

Build Your Own Prompt and Prompt Page

Create your own prompt and prompt page to control how they appear in a report.

Steps

1. Pause the pointer over the page explorer button and click **Prompt Pages**.
2. In the **Insertable Objects** pane, on the toolbox tab, drag **Page** to the **Prompt Pages** box.
3. Double-click the page you just created.
4. In the **Insertable Objects** pane, on the toolbox tab, drag one of the following prompt controls to the prompt page.

Prompt control	Description
Text Box Prompt	Retrieves data based on a value that you type. Use this control when users know exactly what value they want to enter, such as a name or account number.

Prompt control	Description
Value Prompt	Retrieves data based on values that you select from a list. Use this control to show the list of possible values from which users can choose. **Note:** The maximum number of items that can appear in a list is 5000.
Select & Search Prompt	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.
Date Prompt	Retrieves data based on a date that you select. Use this control when you are filtering a date column. With this control, users can retrieve data for a specific day, a set of days, or a range of days.
Time Prompt	Retrieves data based on a time that you select. Use this control to restrict a report to a particular time or time range. For example, you can use this control to see how many orders are received after business hours. You can then use this information to determine the number of staff needed to work after hours.
Date & Time Prompt	Retrieves data based on a date and time that you select. Use this control when you are filtering a datetime or timestamp column. This control is useful for specifying ranges. For example, you can retrieve all orders received from Monday at 12:00 a.m. to Friday at 5:00 p.m.
Interval Prompt	Retrieves data based on a time interval that you specify. Use this control to retrieve data that is related to the passage of time. For example, you can use this control to retrieve a list of products that were returned 30 or more days after they were purchased.
Tree Prompt	Retrieves data based on values you 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" (p. 128).
Generated Prompt	Selects a prompt control based on the data type of the data item. This control acts like a placeholder. When you run the report, the control is replaced by the appropriate prompt control. For example, if you are prompting date values, the control is replaced by a date & time prompt.

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 create a new parameter, define the expression by selecting a data item from the package and the operator you want to use.

     Tip: Make the prompt optional by selecting the **Make the filter optional** check box. When selected, users will not be required to choose a value when the report is run.
   - Go to step 8.
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 you want to use.

     If you are creating a tree prompt, you must choose **in** in the **Operator** box.

     You can use parameters for other purposes besides filtering data. For example, you can 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. When selected, users will not be required to choose a value when the report is run.
   - Click **Next**.
   - If you created a parameterized filter and you have more than one query defined in the report, select the query you want to filter and then 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 points (...) beside **Values to use** and click the data item you want to prompt on.

     Tip: You can choose a data item that is different than what users will see when they are prompted (see the following step). For example, you want to prompt on Product line. The package contains the data items Product line and Product line code. You can choose Product line code as the values to use for faster data retrieval.
   - Click the ellipsis points (...) beside **Values to display** and click the data item that you want users to see when they are prompted.
   - To create a cascading prompt, click the **Cascade Source** box, and select the parameter that represents the cascade source.
7. Click **Finish**.

The prompt control is added to the prompt page. A prompt page is like a report page. You can insert graphics, text, and so on, and apply any formatting you want.

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 Returned for a Specific Time Interval

You are a report author at the Great Outdoors Company, which sells sporting equipment. You are requested to create a report that shows products returned for a time interval specified by the user.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper-right corner of the page.

3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the source tab, double-click the following data items to add them to the list:
   - **Order number**
   - **Product name**
   - **Return quantity**

   Tip: You can find these items in the **Orders** and **Products** folders.
6. In the **Insertable Objects** pane, on the toolbox tab, drag **Query Calculation** to the right of **Return quantity** in the list.
7. In the **Name** box, type the following and click **OK**:

   Time Interval
8. In the **Expression Definition** box, type the following and click **OK**:

   [gosales_goretailers].[Orders].[Return date]-[gosales_goretailers].[Orders].[Order date]
9. From the **Data** menu, click **Filters**.
10. Click the add button.
11. In the **Expression Definition** box, type

    [gosales_goretailers].[Orders].[Return date]-[gosales_goretailers].[Orders].[Order date] > ?p1?
12. 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.
13. Pause the pointer over the page explorer button and click **Prompt Pages**.
14. In the **Insertable Objects** pane, drag **Page** to the **Prompt Pages** pane, and then double-click it.
15. In the **Insertable Objects** pane, on the toolbox tab, drag **Interval Prompt** to the work area.

    The **Prompt Wizard** dialog box appears.
16. Click **Use existing parameter**, and then click **p1**.
17. Click **Finish**.
18. Run the report.

    An interval prompt appears.
19. In the **Days** box, type a value and click **Finish**.

    Tip: You can also type values for the **Hrs** and **Mins** boxes.

A list report appears, showing all products that were returned after the time interval you specified. For example, if you type 30 days, the list will show products that were returned more than 30 days after the order date.

Order number	Product name	Return quantity	Time Interval
9328	Aloe Relief	8	45 days 3 hours 5 minutes 40 seconds
8270	Aloe Relief	2	47 days 14 hours 36 minutes 54 seconds
7344	Aloe Relief	4	58 days 17 hours 44 minutes 6 seconds
5557	Bear Edge	2	35 days 20 hours 52 minutes 48 seconds
5133	Bear Edge	4	35 days 20 hours 52 minutes 49 seconds
8996	Bear Edge	4	37 days 3 hours 9 minutes 14 seconds
6317	Bear Edge	2	37 days 4 hours 11 minutes 16 seconds
7077	Bear Edge	2	42 days 15 hours 37 minutes 27 seconds
7099	Bear Survival Edge	4	31 days 15 hours 36 minutes 26 seconds
9329	Bear Survival Edge	2	34 days 20 hours 52 minutes 18 seconds
6756	Bear Survival Edge	4	35 days 8 hours 19 minutes 30 seconds
2182	Bear Survival Edge	8	41 days 11 hours 28 minutes 43 seconds
8435	Bear Survival Edge	10	48 days 16 hours 40 minutes 30 seconds
6701	Bear Survival Edge	2	53 days 21 hours 55 minutes 22 seconds
1363	BugShield Extreme	8	50 days 20 hours 51 minutes 48 seconds
5105	BugShield Lotion	2	47 days 13 hours 31 minutes 48 seconds
8109	BugShield Natural	6	57 days 5 hours 13 minutes 19 seconds
8301	BugShield Natural	6	58 days 2 hours 7 minutes 10 seconds
6036	Calamine Relief	2	44 days 17 hours 43 minutes 5 seconds

Create a Parameter to Produce a Prompt

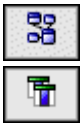
Report Studio can automatically generate prompted reports based on parameters you create. When you run the report, Report Studio can generate a prompt page for each parameter not associated to an existing prompt page, depending on whether the prompt run option (p. 35) is selected or not.

You also define parameters when you want to create a drill-through report (p. 168), and you can use parameters to define master-detail relationships (p. 175).

Steps

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 or data items tab to select the data item you want 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 the Expression Editor" (p. 233).
5. Type a name after the operator to define the prompt parameter. A question mark must precede and follow the name.
6. Click **OK**.
7. In the **Usage** box, click one of the following:
   - To create a required prompt, click **Required**.
   - To create an optional prompt, click **Optional**.
   - To specify not to use the prompt, click **Disabled**.
8. Click **OK**.

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 by their very nature 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 you want using the **Run options** tab in 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.

If you are using the sample reports that come with Report Studio, the reports Consumer Trends, GO Media, Product Line by Year-prompt, and Sales Representative Contact List contain prompt controls on report pages.

Steps

1. In the **Insertable Objects** pane, on the toolbox tab, drag a prompt control to the location where you want it to appear.

   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 (p. 126).
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 setting its **Type** property to **Finish**.

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
- specify that a prompt requires user input
- allow users to select multiple values
- show or hide prompt status
- specify default selections
- specify prompt values
- add a prompt button
- create a cascading prompt

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, Report Studio selects the prompt control interface. You can change the prompt control interface to something different, 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.

Steps

1. Click the prompt control you want to modify.
2. In the **Properties** pane, click the **Select UI** property and click the interface you want.

Specify That a Prompt Requires User Input

You can specify that a prompt requires user input before the report can run.

Steps

1. Click the prompt control you want to modify.
2. In the **Properties** pane, click the **Required** property and click **Yes**.
3. Pause the pointer over the page explorer button and go to 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** and then click **OK**.

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 or a range of values 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.

Steps

1. Click the prompt control that you want to modify.
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 point on the page explorer button and go to 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. If there is more than one query defined in the report, click an object that is linked to the query you want.
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, `[gosales_goretailers].[Products].[Product_line] in ?Product line?` allows users to select multiple product lines.
   - If you are creating a range prompt, change the operator to `in_range`.

     For example, `[gosales_goretailers].[Orders].[Margin] in_range ?Margin?` allows users to specify a margin range.

7. Click **OK** twice.

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.

Steps

1. Click the prompt control you want to modify.
2. In the **Properties** pane, click the **Hide Adornments** property and click **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.

The values you specify must exactly match the corresponding values in the database. If you are specifying values for a date prompt control, you must type the value in the format Y-M-D when prompting a date column. When prompting a date-time column, type the value in the format Y-M-D hh:mm:ss.ms.

Steps

1. Click the prompt control that you want to modify.
2. In the **Properties** pane, double-click the **Default Selections** property.

3. Click the add button, and then choose whether you want to define a single value or a range of values:
   - Click **Simple Selection** to define a single value.
   - Click **Range Selection** to define a range of values.
4. If you chose to define a single value, type the value you want as the default selection.
5. 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. Click **OK**.
7. Repeat steps 4 to 6 to specify other default selections.

   You can specify other default selections only if the **Multi-Select** property for the prompt is set to **Yes**.
8. Click **OK**.

Specify Prompt Values

Provide your own values in a prompt

- to show something different from what is in the database
- to improve performance by not accessing the database
- to provide text for optional prompts, such as Select a value
- to restrict the number of values available

For example, you have a prompt in which users choose a country. For the database value United States, you want USA to appear in the prompt.

Steps

1. Click the prompt control that you want to modify.
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 you want to add.
5. In the **Display** box, type the value that will appear in the prompt.

6. Click **OK**.
7. Repeat steps 4 to 7 to add other prompt values.
8. To link a prompt value to a condition (p. 159), do the following:
   - Click the **Variable** drop-down list and choose the variable you want to use or create your own.

     For information about creating variables, see "Add a Variable" (p. 158).
   - Click the **Value** drop-down list and click one of the possible values for the variable.
   - Click the static value you want to link to the variable and click the edit button.
   - In the **Display** box, type the value that you want to appear in the prompt for that particular value of the variable.
   - Repeat this procedure for each additional value.
9. Click **OK**.

Add a Prompt Button

Add prompt buttons so that users can submit selected items, cancel a report, or navigate between pages.

When you are building prompts and prompt pages (p. 119), you may have to add a prompt button 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.

Steps

1. Pause the pointer over the page explorer button and click the report or prompt page that contains the prompt control for which you want to add a prompt button.
2. In the **Insertable Objects** pane, on the toolbox tab, drag **Prompt Button** to the work area.
3. Click the prompt button and in the **Properties** pane, click the **Type** property. Choose one of the following actions.

Goal	Action
Cancel the report	**Cancel**
Go to the previous prompt page	**Back**
Go to the next prompt page	**Next**
Run the report	**Finish**
Reprompt the user **Tip:** Reprompting users is useful when you have cascading prompts.	**Reprompt**

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 cascading source. When users select a product line, they see only the product types related to the selected product line.

Steps

1. To make the cascading source a required prompt, click it and in the **Properties** pane, set the **Required** property to **Yes**.
2. Click the prompt control that you want 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, and click **OK**.

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:

- In the **Insertable Objects** pane, on the toolbox tab, drag **Prompt Button** to the report.
- Click the prompt button and in the **Properties** pane, click the **Type** property and click **Reprompt**.

- To change the text in the prompt button, in the **Insertable Objects** pane, on the toolbox tab, drag **Text Item** to the prompt button and type the text you want.

For more information about how to create cascading prompts, see the Report Studio Tour.

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 (p. 119) 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.

Steps

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 that you are prompting on.
3. In the **Expression Definition** box, type one of the following functions and click **OK**.

Goal	Function
Show the hierarchical structure of all members in the hierarchy. If this function is not used, all members are shown in a flat list.	rootmembers(*data_item*)
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 2004, 2005, and 2006 in the tree. If x=2, you will see 2004 Q1, 2004 Q2, and so on.	descendants(rootmembers(*data_item*, x)
Show the children of a member. For example, 2004 Q1, 2004 Q2, 2004 Q3, and 2004 Q4 appears for the member 2004.	children(*member*)

Working with Queries

Queries specify what data appears in the report. In Report Studio, you create and modify queries using Query Explorer. Query Explorer provides an alternative way to modify existing reports or 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: By default, queries run sequentially. Your administrator must enable the concurrent query execution feature. For more information, see the *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

Relating Queries to Layouts

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 don't want to show. The query and layout portions of a report must be linked to produce a valid report.

Report Studio automatically links query and layout. For example, when you use Report Studio and the list report template, query and layout are automatically linked. In addition, you can create a query and manually link it to a data container. In the layout, when you drag a data container to the work area, you can specify which query to use by clicking the **Query** property in the **Properties** pane. You then add the items in the query that you want to see in the layout by dragging them from the data items tab in the **Insertable Objects** pane.

Working with Dimensional Data Sources

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.

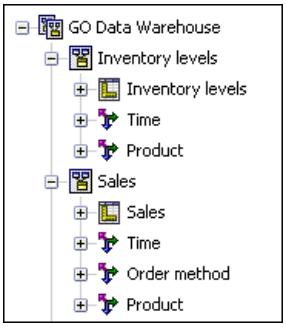
When performing multi-cube queries using dimensional data sources, the following restrictions apply:

- Only basic operators (+, *, /,-) are available for cross-cube calculations.
- Inner joins are not supported.
- All joins for multi-cube queries are outer joins.
- You cannot sort or filter on a conformed dimension (query subject).
- Conformed dimensions are created in Framework Manager.
- 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.

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, as well as on the level of granularity and on the additive nature of the data.

The following data source has characteristics that affect results if you use a multiple-fact query with Inventory levels and Sales. 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 available for changing a query to obtain the results that you want.

Conformed Dimensions

Individual queries on Inventory levels and Sales by Quarter and Product yield the following results.

Quarter	Product	Opening inventory	Closing inventory	Quantity
200501	Husky Rope 50	720	727	608
	Aloe Relief	1,128	1,236	1,256
	Course Pro Umbrella	1,077	1,217	902
200501		**975**	**1,060**	**2,766**
200502	Husky Rope 50	667	632	1,246
	Aloe Relief	988	874	1,710
	Course Pro Umbrella	969	1,025	2,158
200502		**874**	**843**	**5,114**

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, Time and Product.

Product and Time apply to both Inventory levels 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 200501, but there were no sales of this product in the same time period, Quantity would show a null in the Husky Rope 50 row.

Non-conformed Dimensions

If a non-conformed dimension is added to the query, the nature of the results returned by the query is changed.

Quarter	Product	Order method	Opening inventory	Closing inventory	Quantity
200501	Husky Rope 50	Telephone	720	727	254
	Husky Rope 50	Mail	720	727	70
	Husky Rope 50	E-mail	720	727	50
	Husky Rope 50	Web	720	727	98
	Husky Rope 50	Sales visit	720	727	136
	Aloe Relief	Telephone	1,128	1,236	400
	Aloe Relief	E-mail	1,128	1,236	138
	Aloe Relief	Web	1,128	1,236	378
	Aloe Relief	Sales visit	1,128	1,236	340
	Course Pro Umbrella	Fax	1,077	1,217	72
	Course Pro Umbrella	Telephone	1,077	1,217	140
	Course Pro Umbrella	E-mail	1,077	1,217	202
	Course Pro Umbrella	Web	1,077	1,217	80
	Course Pro Umbrella	Sales visit	1,077	1,217	224
	Course Pro Umbrella	Special	1,077	1,217	184
200501			**975**	**1,060**	**2,766**

Order Method 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 200501 is 2,766 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.

Quarter	Product	Order method	Opening inventory	Closing inventory	Quantity
200501	Husky Rope 50		720	727	
	Aloe Relief		1,128	1,236	
	Course Pro Umbrella	Fax	1,077	1,217	72
200501			**975**	**1,060**	**72**
200502	Husky Rope 50	Fax	667	632	82
	Aloe Relief	Fax	988	874	94
	Course Pro Umbrella	Fax	969	1,025	314
200502			**874**	**843**	**490**
200601	Husky Rope 50	Fax	722	741	132
	Aloe Relief	Fax	1,175	1,233	40
	Course Pro Umbrella	Fax	1,463	1,764	130
200601			**1,120**	**1,246**	**302**
200602	Husky Rope 50	Fax	725	714	86
	Aloe Relief	Fax	1,114	1,024	36
	Course Pro Umbrella		1,520	1,388	
200602			**1,120**	**1,042**	**122**
Summary			**1,022**	**1,048**	**986**

The filter is applied only to one subject area, Sales. Because Order method does not exist in Inventory levels, all products still appear in the report. For example, Course Pro Umbrella was in inventory in 200602. Because there were no sales using the fax order method for this product in 200602, Order method 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.

Quarter	Product	Order method	Opening inventory	Closing inventory	Quantity
200501	Course Pro Umbrella	Fax	1,077	1,217	72
200501			**1,077**	**1,217**	**72**
200502	Husky Rope 50	Fax	667	632	82
	Aloe Relief	Fax	988	874	94
	Course Pro Umbrella	Fax	969	1,025	314
200502			**874**	**843**	**490**
200601	Husky Rope 50	Fax	722	741	132
	Aloe Relief	Fax	1,175	1,233	40
	Course Pro Umbrella	Fax	1,463	1,764	130
200601			**1,120**	**1,246**	**302**
200602	Husky Rope 50	Fax	725	714	86
	Aloe Relief	Fax	1,114	1,024	36
200602			**919**	**869**	**122**
Summary			**989**	**1,025**	**986**

The summary for quantity is 986 using either filter method, which shows the results are based on the same data set.

In 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 "Filter Data" (p. 103).

Add Queries to a 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.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. In the **Insertable Objects** pane, drag one of the following objects to the work area.

Object	Description
Query	Adds a query.
Join	Adds a join relationship.
Union	Adds a union operator.
Intersect	Adds an intersect operator.
Except	Adds an except (minus) operator.
SQL	Adds SQL.
MDX	Adds MDX.

3. In the **Properties** pane, specify the properties you want.

   For example, if you added a join, double-click the **Join Relationships** property to define the join.

4. Double-click a query.
5. In the **Insertable Objects** pane, on the source tab, drag data items that you want in the query 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, if you want to filter on Product line code but you want users to see Product line in the layout, you must add both data items to the query.

6. To create a new data item in the query, in the **Insertable Objects** pane, on the toolbox tab, drag **Data Item** to the **Data Item** pane.
7. To add a filter, in the **Insertable Objects** pane, on the toolbox tab, drag **Filter** to the **Detail Filters** pane or to the **Summary Filters** pane, and define the filter expression (p. 103).

   Tip: You can also create a filter by dragging a data item from the source tab to the one of the filters panes and then completing the filter expression. If you are working with a dimensional data source, you can quickly filter data by dragging a member instead of the level to which the member belongs. For example, dragging the member 2006 from the Years level will filter data for the year 2006. This method is quicker than dragging the Years level and then completing the filter expression to specify to show data only for the year 2006.

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

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.

To combine two queries, the following conditions must be met:

- The two queries must have the same number of data items.
- The data types of the data items must be compatible and the data items must 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 perfectly.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. In the **Insertable Objects** pane, 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 the data items that you want.
4. On the toolbar, press the back button to return to the Queries work area.
5. Click the set operator that you added in step 3.
6. In the **Properties** pane, click the **Duplicates** property and choose whether you want 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. If you want Report Studio to automatically produce the list of projected data items, click **Automatically generated**.

   Report Studio generates the projection list using only one of the two queries being unioned.
9. To add, delete, move, or rename data items in the projection list, click **Manual** and make the changes you want.
10. Click **OK**.

11. Double-click the union query.
12. In the **Insertable Objects** pane, on the source tab, drag the data items that you want to add to the query to the **Data Items** pane.

The union query is complete. You can now link it 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 Great 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.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper-right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Blank** and click **OK**.

5. Pause the pointer over the query explorer button and click **Queries**.
6. In the **Insertable Objects** pane, 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. Double-click **Query2**.
8. In the **Insertable Objects** pane, on the source tab, drag the following data items to the **Data Items** pane:
   - from the **Products** folder, **Product line**

- from the **Orders** folder, **Revenue**

9. In the **Insertable Objects** pane, on 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.
10. In the **Expression Definition** box, type **'A'** and click **OK**.
11. In the **Properties** pane, click the **Name** property, delete the default name, and type
    Sort key

12. On the toolbar, press the back button to return to the Queries work area.
13. Double-click **Query3**.
14. In the **Insertable Objects** pane, on the source tab, drag the following data items to the **Data Items** pane:
    - **Order method**
    - **Revenue**

    Tip: You can find these items in the **Orders** folder.
15. Repeat steps 9 to 11 to create a Sort key data item in Query3 with 'B' as its definition.
16. On the toolbar, press the back button to return to the Queries work area.
17. Click the **Union** operator.
18. Double-click the **Projection List** property.
    The **Product line** item in the projection list contains both product lines and order methods.
19. Click **Manual**.

20. Click **Product line** and then click the edit button.
21. In the **Edit** box, type the following after **Product line**:
    & Order method
22. Click **OK** twice.
23. Double-click **Query1**.
24. In the **Insertable Objects** pane, on the source tab, drag the following data items to the **Data Items** pane:
    - **Product line & Order method**
    - **Revenue**
    - **Sort key**
25. With **Sort key** selected, in the **Properties** pane, click the **Pre-Sort** property and click **Sort ascending**.
    When you run the report, all product lines will appear first, followed by all order methods.

26. Pause the pointer over the page explorer button and click **Page1**.
27. In the **Insertable Objects** pane, on the toolbox tab, drag **List** to the work area.
28. Click the list.

29. Click the select ancestor button in the title bar of the **Properties** pane and click **List**.
30. Click the **Query** property and click **Query1**.
    The list is linked to the union query.
31. Double-click the **Properties** property.
32. 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.

33. In the **Insertable Objects** pane, on the data items tab, drag the following items from Query1 to the list:
    - **Product line & Order method**
    - **Revenue**
34. Run the report.

A list report with two columns is produced. All product lines and order methods appear in the first column.

Product line & Order method	Revenue
Camping Equipment	$89,713,990.92
Golf Equipment	$25,905,465.58
Mountaineering Equipment	$20,891,350.60
Outdoor Protection	$3,171,114.92
Personal Accessories	$31,894,465.86
E-mail	$27,057,568.66
Fax	$9,425,161.70
Mail	$9,207,267.26
Sales visit	$40,711,148.52
Special	$3,323,309.16
Telephone	$44,323,199.10
Web	$37,528,733.48

Create a Join Relationship

You can create a join relationship to join two queries.

In general, join relationships should be created in the Framework Manager model. Create a join relationship in Report Studio if what you are trying to do cannot be modeled in Framework Manager.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. In the **Insertable Objects** pane, 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 the data items that you want.

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. Click a data item in the left query, and then click a data item in the right query to create a link.
9. For each query, click **Cardinality**, and then click the cardinality that you want to use.
10. Click **Operator**, and then click the operator that you want to use.
11. To convert the join relationship to an expression, click the **Convert to expression** button.

    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. Click **OK**.

13. Double-click the join query.
14. In the **Insertable Objects** pane, on the source tab, drag the data items that you want to add to the **Data Items** pane.

The join query is complete. You can now link it to a data container in the layout.

Add Dimension Information to a Query

Add dimension information to a query to obtain the results that you want. 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, Cognos 8 creates a default dimension structure.

Dimension information is not intended to define the presentation of information, but to help query planning. It can be considered as a form of query hint. In addition, users will not be able to drill down (p. 172) on dimensions that you add to a report.

Steps

1. Pause the pointer over the query explorer button and click the query you want.
2. In the **Properties** pane, click the **Override Dimension Info** property and click **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, in the **Insertable Objects** pane, on the source tab, drag the data item to the **Dimensions** pane.

   Report Studio automatically generates the entire dimension.

5. To create a new dimension, in the **Insertable Objects** pane, on the toolbox tab, drag **Dimension** to the **Dimensions** pane, and then build it by adding the objects you want:
   - 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 "Working with Dimensional Data" (p. 32).
6. In **Insertable Objects** pane, on the source tab, drag the data items you want 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, in the **Insertable Objects** pane, on the source tab, drag the data item you want to the **Facts** pane.

Example - Create a Crosstab Report that Shows Empty Rows

You are a report author at the Great 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.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper-right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Crosstab** and click **OK**.

5. In the **Insertable Objects** pane, on the source tab, drag the following data items to the crosstab:
   - **Product line** as rows
   - **Order year** as nested rows
   - **Order method** as columns
   - **Revenue** as the measure

   Tip: You can find these items in the **Products** and **Orders** folders.
6. Right-click the crosstab and click **Go to Query**.
7. In the **Properties** pane, click the **Override Dimension Info** property and click **Yes**.

   The **Dimension Info** tab appears at the bottom of the work area.

8. Click the **Dimension Info** tab.
9. In the **Insertable Objects** pane, on the source tab, drag the following items to the **Dimensions** pane:
   - **Product line**
   - **Order year**

   Product line and **Order year** become separate dimensions in the query.

10. Pause the pointer over the page explorer button and click **Page1**.
11. Click **Order year**.
12. In the **Properties** pane, double-click the **Sort** property.
13. In the **Data Items** box, drag **Order year** to the **Sort List** box and click **OK**.
14. Run the report.

All order years appear for all product lines, even if no revenue was produced.

Revenue		Web	Telephone	Fax	Mail	Special	E-mail	Sales visit
Mountaineering Equipment	2004							
	2005	$1,922,374.36	$2,524,028.34	$662,612.52	$728,792.28	$248,532.04	$1,468,181.34	$2,088,153.66
	2006	$3,122,019.60	$2,855,944.64	$557,716.86	$244,454.22	$162,907.36	$1,828,950.12	$2,476,683.26
Personal Accessories	2004	$854,223.96	$2,058,122.44	$497,113.74	$694,807.24	$130,953.28	$1,132,334.20	$1,777,242.66
	2005	$2,142,070.60	$2,938,441.44	$598,985.66	$568,820.56	$337,283.82	$1,599,997.90	$2,770,108.06
	2006	$3,559,743.22	$3,468,271.58	$566,175.82	$407,107.52	$193,990.50	$2,309,879.90	$3,288,791.76
Golf Equipment	2004	$902,238.60	$1,567,379.72	$407,812.22	$546,070.02	$61,207.18	$983,890.06	$1,129,383.06
	2005	$1,793,605.92	$2,410,111.66	$508,769.58	$756,304.66	$229,453.08	$1,412,003.24	$2,488,020.74
	2006	$2,833,810.30	$2,454,580.78	$635,590.76	$377,717.62	$272,093.56	$1,461,270.32	$2,674,152.50
Outdoor Protection	2004	$235,137.74	$430,100.22	$90,874.96	$153,497.06	$21,108.06	$218,439.06	$387,299.14
	2005	$231,261.18	$246,964.74	$53,224.76	$45,023.92	$28,032.20	$156,742.04	$226,981.80
	2006	$175,267.48	$142,908.00	$32,387.72	$22,686.90	$11,343.26	$112,951.26	$148,883.42
Camping Equipment	2004	$2,713,489.04	$6,064,567.16	$1,454,618.22	$1,865,955.94	$323,438.84	$3,606,197.82	$4,443,061.86
	2005	$6,451,499.90	$7,880,932.52	$1,522,878.58	$1,906,834.58	$933,539.64	$4,596,732.88	$8,081,188.36
	2006	$10,591,991.58	$9,280,845.86	$1,836,400.30	$889,194.74	$369,426.34	$6,169,998.52	$8,731,198.24

Reference Package Items in Child Queries

When you create a child query in 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.

Steps

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 of the **Insertable Objects** pane.
4. Add the package items you want.

   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 you want in the **Expression Definition** box.
5. Click the up arrow 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.

Working with 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. You can

- view the SQL or MDX
- edit the SQL or MDX
- build a report using your own SQL or MDX
- choose the data source

When working with a relational data source, the SQL that is produced by Report Studio depends on the report output selected.

View the SQL or MDX

View the SQL or MDX to see what is passed to the database when you run a report.

Steps

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 you want.
   - 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 click the view tabular data button, which 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.

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 Report Studio uses. Cognos SQL is converted to native SQL before the query is executed.

Edit the SQL or MDX

You can edit the SQL or MDX for a query to suit your particular needs. For example, you may want to edit the SQL or MDX to improve performance or to troubleshoot problems.

If you did not originally build the query using SQL or MDX, you must convert the query to SQL or MDX before you can make changes. Converting a query to SQL or MDX is an irreversible process.

Steps

1. Pause the pointer over the query explorer button and click the query you want.
2. In the **Properties** pane, double-click the **Generated SQL/MDX** property.
3. Click **Convert**.
4. Make the changes you want.

   If you are working with MDX, you must be aware of the MDX syntax that Report Studio supports.
5. Click **Validate** to check for errors.
6. Click **OK**.

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 Report Studio supports.

Steps

1. From the **File** menu, click **New**.
2. Click **Blank** and click **OK**.
3. Pause the pointer over the query explorer button and click **Queries**.
4. In the **Insertable Objects** pane, drag **Query** to the work area.
5. In the **Insertable Objects** pane, do one of the following:
   - To build a 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 to choose the data source.
7. Click the data source you want and click **OK**.
8. If required, in the **Catalog** property, type the name of the catalog that you want to use.
9. Double-click the **SQL** or **MDX** property.
10. Type the SQL or MDX you want to use.
11. Click **Validate** to check for errors.
12. Click **OK**.
13. Double-click the query.

    If the SQL or MDX is valid, data items defined in the SQL or MDX appear in the **Data Items** pane.
14. Pause the pointer over the page explorer button and click a report page.
15. In the **Insertable Objects** pane, on the toolbox tab, drag the report type (p. 43) you want to create to the work area.

    For example, drag a list, crosstab, chart, or repeater.
16. Click the data container.
17. In the **Properties** pane, click the select ancestor button and then click the container you just created.

    For example, if you created a list, click **List**.
18. Click the **Query** property and click the query you want to use for the report type.
19. In the **Insertable Objects** pane, on the data items tab, drag the items you want from the query you chose in the previous step to the data container.

Choose the Data Source

If you converted a query to SQL or MDX or created a query by adding your own SQL or MDX, you must choose the data source against which you want to execute the query.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. Click the **Data Source** object of the query for which you want to choose the data source.
3. In the **Properties** pane, double-click the **Data Source** property.
4. Click the data source you want and click **OK**.

First-Rows Optimization

The SQL produced by 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. Report Studio supports the MDX grammar as specified in Microsoft's Data Access SDK version 2.8, with the following exceptions.

Report Studio does not support the following MDX functions. This is a partial list and applies only when you are working with the following data sources:

- cubes
- DB2 OLAP
- Cognos Consolidation
- Cognos Contributor

For these data sources, Cognos 8 uses a variation of the MS Analysis Services syntax.

AllMembers	Ignore	SetToArray
Ascendants	Is	SetToStr
CalculationCurrentPass	IsAncestor	StddevP
CalculationPassValue	IsGeneration	StdevP
Call	IsLeaf	StripCalculatedMembers
Correlation	IsSibling	StrToMember
Covariance	LinkMember	StrToSet
CovarianceN	LinRegIntercept	StrToTuple
Current	LinRegPoint	StrToValue
DataMember	LinRegR2	ToggleDrillState
DrillDownLevel	LinRegSlope	TupleToStr
DrillDownLevelBottom	LinRegVariance	UniqueName
DrillDownLevelTop	LookupCube	UserName
DrillDownMember	MemberToStr	ValidMeasure
DrillDownMemberBottom	Name	VarianceP
DrillDownMemberTop	NameToSet	VarP
DrillupLevel	NonEmptyCrossjoin	VisualTotals
DrillupMember	Predict	

Add a Query Macro

Add a query macro to allow run time changes to be made to SQL queries. 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 also add query macros to the Framework Manager model. For more information, including examples, see the Framework Manager *User Guide*.

Support for query macros in Report Studio includes the same capabilities as macros used in Framework Manager, including support for parameter maps defined within the Framework Manager model. 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.

Steps

1. Pause the pointer over the query explorer button and click **Queries**.
2. If the query to which you want to add 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.
   - Click **OK**.
3. If the query to which you want to add 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.
   - Click **OK**.

Tip: For a list of supported macro functions, see "Using the Expression Editor" (p. 233).

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.

The `prompt` and `promptmany` functions have the following mandatory and optional parameters. All argument values must be specified as strings.

Parameter	Description
Name (mandatory)	Name of the prompt. 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.

Parameter	Description
Datatype (optional)	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: • boolean • date • datetime • decimal • double • float • int • integer • interval • long • 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 • xsddatetime • xsddecimal • xsddouble • xsdduration • xsdfloat • xsdint • xsdlong • xsdshort • xsdstring • xsdtime For more information about which data types are supported, visit the Cognos Global Customer Services Web site (http://support.cognos.com).
DefaultText (optional)	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.
Text (optional)	Text that precedes any user-provided values, such as, 'and column1 = '.
QueryItem (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 (optional)	The closing parenthesis which is used most often for the `promptmany` function. It is also useful when the prompt is optional and is followed by hardcoded filters in the SQL statement.

Example - Select a Country Prompt

You are a report author at the Great Outdoors Company, which sells sporting equipment. You are requested to create a report that will prompt users to choose the country for which they want to see data.

The following code shows how you can use macros to create a 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')#
```

Notes

- Because the `<Datatype>` argument is not specified, valid prompt values must be strings, which is correct in this case.
- Because the `<DefaultText>` argument is not specified, the `CountryName` prompt is a required prompt.

Chapter 5: Working with 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 *Administration and Security Guide*.

You can

- open and save reports locally
- open a report from the clipboard
- remove upgrade messages
- open files from other Studios
- manage changes in the package
- add multiple items to a single column
- burst a report
- customize reports using conditions
- set up drill-through access
- drill-up/drill-down in the same report
- create a master-detail relationship

Open and Save a Report Locally

By default, the reports you create are stored on the Cognos 8 server. You can, however, open and save reports on your computer. This is useful if you want to send a report to a report author that is working in a different environment, or you want to save the report to a source code controlled directory on a local network or drive.

Steps

1. Obtain the LFA.dll from your Cognos 8 administrator.
   The DLL is located in the bin directory where Cognos 8 is installed.
2. Open a command prompt window.
3. Register the LFA.dll file by typing
   regsvr32 LFA.dll
4. In Internet Explorer, set your computer and the Cognos 8 server as trusted sites:
   - From the **Tools** menu, click **Internet Options**.
   - On the **Security** tab, click **Trusted sites**.
   - Click the **Sites** button.
   - In the **Add this Web site to the zone** box, type **http://localhost** and click **Add**.
     Tip: If the Cognos 8 server is not on the same computer as the browser, type **http://<computer name>** instead.
   - Repeat the above step for **http://<Cognos 8 servername>**.
   - Clear the **Require server verification (https:) for all sites in this zone** check box.
   - Click **OK** twice.
5. In Report Studio, from the **Tools** menu, click **Options**.
6. Select the **Allow local file access** check box and click **OK**.
   The menu items **(Local) Open** and **(Local) Save As** appear in the **File** menu.

7. Close and restart Report Studio.

You can now open and save reports on your computer. The first time you try to open or save a report locally, Internet Explorer asks you whether you want to allow an ActiveX control on the page to interact with other parts of the page. Click **Yes** to continue.

Open a Report from the Clipboard

You can open a report specification (p. 42) that was previously copied to the clipboard (**Tools** menu, **Copy To Clipboard**). This is useful for importing an XML report specification from outside the Cognos 8 environment.

Although Report Studio attempts to validate the report specification, it is your responsibility to ensure that it is correct. For more information, see the Cognos 8 SDK *Developer Guide*.

Step

- From the **Tools** menu, click **Open Report From Clipboard**.

Remove Upgrade Messages

When you open a report that was created in a previous version of Report Studio, any problems detected during the upgrade process appear in the **Upgrade Information** dialog box. You must fix these problems to complete the upgrade of the report.

After you fix the problems detected, information about the upgrade issues remains in the report specification. The report will not run until you verify that the problems have been fixed by clearing the upgrade messages.

Steps

1. From the **File** menu, click **Upgrade Information**.
2. Select the **Delete upgrade error information** check box.
3. Click **OK**.

You can now run the report.

Open Files from Other Studios

You can open reports, templates, or analyses that were created in Query Studio or Analysis Studio in Report Studio. All the capabilities of Report Studio are available to you, to change the formatting, layout, calculations, and queries.

You can also open a report, template, or analysis in Report Studio from Cognos Connection with the **Open with Report Studio** action.

Steps

1. From the **File** menu, click **Open**.
2. Click the report, template or analysis that you want.
3. Click **Open**.

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.

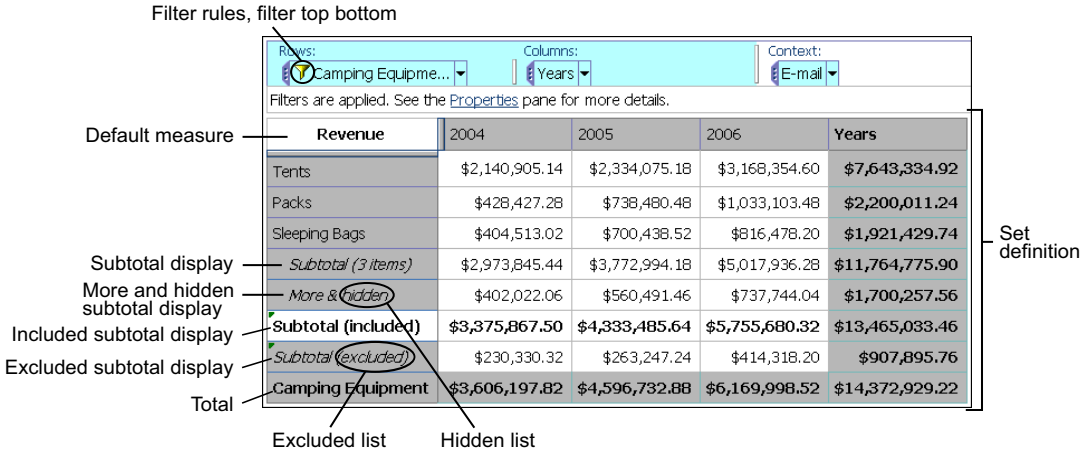
Components of an Analysis Studio Query Specification

Analysis Studio defines each group of rows and columns as a set. When an analysis is imported into Report Studio, the report will have one query which processes all the sets found on the crosstab. Each set is defined by 18 data items which segment and summarize the base set definition. In order to maintain reports converted from Analysis Studio, you must understand what each of these items represent and how they relate to each other.

The data items for a set specify the following:

- The set definition.
- The set segments, including which members were excluded and hidden individually.
- The filter rules for defining which members are to be retrieved.
- The calculations for Subtotal (N items), More & Hidden, Subtotal (included), Subtotal (excluded), and the total.

  For more information about the subtotals, see the Analysis Studio *User Guide*.
- Which subtotals should appear.
- Any user-defined calculations.



Data item	Definition	Dependencies
<measure name> (default)	Identifies which measure is to be the default measure for the crosstab or chart. May be empty if no default measure is specified.	
Set Definitions		
<set name> (base)	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.	
<set name> (level)	Identifies the level for a level-based set.	
<set name> (list)	Defines the list of members in a selection-based set.	
<set name> (depth N)	Defines the set of members at N, number of levels down.	
<set name> (named set)	References a predefined set.	

Data item	Definition	Dependencies
Set Segment Definitions		
<*set definition*> (hidden list)	Lists the members manually hidden using the **Hide** command in Analysis Studio. This set appears when the user tries 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 for showing the last two members if that is all that remained 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 those members that were manually excluded from the analysis. If no members were manually excluded, the expression defines an emptySet(). This set appears when the user tries to remove a member from the **Excluded items** list in the Properties pane.	<*set definition*>
Filters		
<*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*>
<*set definition*> (excluded filters)	Removes those members that were manually excluded from the results after applying the user-defined rules.	<*set definition*> (excluded list), <*set definition*> (filter rules)
<*set definition*> (filter top bottom)	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 Analysis Studio *User Guide*.	<*set definition*> (excluded filters) and totals for sets on the opposite axis
Subtotals and Related Conditions		
<*set definition*> (subtotal)	Aggregates the visible items shown for the set.	<*set definition*> (visible items set)
<*set definition*> (subtotal display)	Shows the subtotal if the number of included items is greater than the number of visible items.	<*set definition*> (visible items set), <*set definition*> (included set), <*set definition*> (subtotal)

Data item	Definition	Dependencies
<*set definition*> (more and hidden subtotal)	Calculates the **More & hidden** subtotal by subtracting the aggregation of the included members from the aggregation of those members manually hidden or hidden by exceeding the maximum display limit. Not available for selection-based sets.	<*set definition*> (included set), <*set definition*> (visible items set), <*set definition*>
<*set definition*> (more and hidden subtotal as set)	Converts the **More & hidden** subtotal member to a set for use in set operations.	<*set definition*> (more and hidden subtotal)
<*set definition*> (more and hidden subtotal as member)	Generically references **More & hidden** subtotal.	<*set definition*> (more and hidden subtotal)
<*set definition*> (more and hidden subtotal display)	Shows the **More & hidden** subtotal if the number of items hidden or clipped is greater than zero.	<*set definition*> (more and hidden subtotal)
<*set definition*> (included subtotal)	Calculates the value for **Subtotal (included)**. For selection-based sets, this data item references the summary data item for the set.	<*set definition*> (included set)
<*set definition*> (included subtotal display)	Shows **Subtotal (included)** if any members passed the filter criteria.	<*set definition*> (included set)
<*set definition*> (excluded subtotal)	Calculates the value for **Subtotal (excluded)**. This data item is not available for selection-based sets.	<*set definition*> (total), <*set definition*> (included subtotal), <*set definition*>
<*set definition*> (excluded subtotal display)	Shows **Subtotal (excluded)** if the number of members in the filtered set is less than the base set. The filter rule is filtering out something.	<*set definition*> (excluded subtotal), <*set definition*>
<*set definition*> (total)	Identifies the total for the set and 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.	<*set definition*>

Managing Changes in the Package

If changes were made to the package used to create a report, the report must be updated. When you open a report, 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

- the namespace name or query subject and data item names in the package have changed
- the name of the package has changed

Update Name References

If the namespace name or query subject and data item names in the package have changed, you must update reports created with the package to reflect the change.

The names of data items in a report are a concatenation of the namespace name, query subject name, and the data item name. For example, if you add Order number from the GO Sales and Retailers sample package to a filter expression, you will see `[gosales_goretailers].[Orders].[Order number]` in the expression. Similarly, package filter names are a concatenation of the namespace name and the filter name.

Steps

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, and click **OK**.
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, and click **OK**.

Change the Package Connection

If the name of the package used to create a report has changed, change the package connection to update the report.

At the same time, you can also change the authoring language for the report.

Steps

1. From the **File** menu, click **Report Package**.
2. In the **Package** box, click the package to which you want to link the report.
3. To change the authoring language, in the **Language** box, click a different choice.

   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**.

   If errors are found, the **Validation Errors** dialog box appears, showing what elements 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 (p. 128) to make all the changes.

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.

Steps

1. From the **Structure** menu, click **Lock Page Objects** to unlock the report.
2. In the **Insertable Objects** pane, drag the items that you want 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 Great 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. To reduce the number of columns in the list, you show email addresses, phone numbers, extensions, and fax numbers in a single column.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.

4. In the **New** dialog box, click **List** and click **OK**.
5. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:
   - **Country**
   - **City**
   - **Staff name**
   - **Position**
   - **Email**
   - **Work phone**
   - **Extension**
   - **Fax**

   Tip: You can find these data items in the **Sales branch address** and **Sales reps** folders.

6. Click the **Country** column, and from the **Structure** menu, click **Section**.
7. From the **Structure** menu, click **Lock Page Objects**.

   The report is unlocked.
8. Click **Country**, and in the **Properties** pane, double-click the **Font** property.
9. Change the font to **Arial Black**, **11 pt**, and **Bold**, and click **OK**.

10. In the **Insertable Objects** pane, on the toolbox tab, drag **Table** to the right of the **Work phone** text item in the **Work phone** column (not the column title), and create a table that has one column and three rows.
11. Drag the following data items to the table:
    - **Email** to the first row in the table
    - **Work phone** to the second row in the table
    - **Fax** to the third row in the table
12. Drag **Extension** to the right of **Work phone**.
13. In the **Insertable Objects** pane, on 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:**

14. Press Ctrl+click to select the **Extension**, **Email**, and **Fax** column titles, and click the delete button.
15. Click the **Work phone** column title.
16. In the **Properties** pane, click the **Source Type** property and click **Text**.
17. Double-click the **Text** property.
18. Type the following and click **OK**.

    Contact Information
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. Save the report.

When you run the report, contact information for each sales representative appears in a single column.

Sales Representatives Contact List

Australia

City	Staff name	Position	Contact Information
Melbourne	Alice Walter	Branch Manager	Email: AWalter@grtd123.com Work phone: +(61) 03 2982 4242 ext: 1845 Fax: +(61) 03 2982 4243
Melbourne	Dave Smythe	Level 1 Sales Representative	Email: DSmythe@grtd123.com Work phone: +(61) 03 2982 4242 ext: 1875 Fax: +(61) 03 2982 4243
Melbourne	Donald Ward	Level 2 Sales Representative	Email: DWard@grtd123.com Work phone: +(61) 03 2982 4242 ext: 1874 Fax: +(61) 03 2982 4243
Melbourne	Jake Cartel	Level 2 Sales Representative	Email: JCartel@grtd123.com Work phone: +(61) 03 2982 4242 ext: 1648 Fax: +(61) 03 2982 4243
Melbourne	John Sinden	Level 3 Sales Representative	Email: JSinden@grtd123.com Work phone: +(61) 03 2982 4242 ext: 1892 Fax: +(61) 03 2982 4243

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 distribution to recipients who each view only a subset of the data. For example, salespeople in different regions each need a report showing the sales target for each country. You use burst reports to send each salesperson the information they need. Burst reports can be distributed by email or saved to a directory for viewing in Cognos Connection.

To burst a report against a dimensional data source, see "Creating Burst Reports Using a Dimensional Data Source" (p. 156).

You cannot burst crosstab reports.

To create a burst report, you

- ❑ define the burst recipients
- ❑ specify burst groups
- ❑ set burst options
- ❑ enable bursting

Defining Burst Recipients

Define the recipients that 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 can

- create a calculated field in the report
- create a burst table in the data source

Create a Calculated Field

You can use a calculated field to dynamically create the burst recipients.

Steps

1. Pause the pointer over the query explorer button and click the query that will produce the data you want to distribute.

2. In the **Insertable Objects** pane, on 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, the following expression builds the list of sales representatives for The Great Outdoors Company. The expression concatenates the first letter of each representative's first name with their last name.

   ```
   lower(substring([gosales_goretailers].[Sales reps].[First name],1,1) +
   [gosales_goretailers].[Sales reps].[Last name])
   ```
4. To give the data item a more meaningful name, in the **Properties** pane, click the **Name** property, type a different name, and press the Enter key.

Creating a Burst Table

You can create a burst table to specify the list of recipients.

You can also use an existing table as the burst table.

Creating a burst table involves

- ❑ creating the table in the source database
- ❑ importing the table into a package

Create the Burst Table in the Source Database

Create a 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 that you want to burst on.

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 that will contain the email address of each recipient.

After you create the table, add the recipients you want to 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.

Recipient example	Recipient type
CAMID(":Canada")	Group
CAMID(":")/contact[@name='Silvano Allessori']	Contact
CAMID(":")/distributionList[@name='European Partners']	Distribution list
CAMID("LDAP_Local_ID:u:uid=gbelding,ou=people")	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
c8@cognos99.com	Email address

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 directory. You can obtain the path in Cognos Connection. Open the **Set properties** page for each recipient, and click the **View the search path** link. 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:

```
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 *Administration and Security Guide*.

Import the Table

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.

In Framework Manager, do the following:

- ❑ Open the package you want.
- ❑ Import the table.
- ❑ Define the relationship between the burst table and the table containing the data item that you want to burst on.

  For example, you are bursting on country code. You define a relationship between country code in the burst table and country code in the Country table.
- ❑ Save and publish the package.

For more information about importing tables and creating relationships, see the Framework Manager *User Guide*.

Specify Burst Groups

Specify burst groups to set how the report is distributed. Burst groups are defined by a data item that you create in the report or that you add from the burst table.

Steps

1. Pause the pointer over the query explorer button and click the query that will produce the data you want to distribute.

2. If you are creating a data item, do the following:
   - In the **Insertable Objects** pane, on the toolbox tab, drag **Data Item** to the **Data Items** pane.
   - In the **Expression Definition** box, type the expression that defines the burst key and click **OK**.

     For example, the following expression builds an email address for each sales representative in The Great Outdoors Company. The expression incorporates the calculated field previously created, named `userID` below, with `cognos99.com` as the domain name.

     ```
     [userID]+'@cognos99.com'
     ```

     Tip: To give the data item a more meaningful name, in the **Properties** pane, click the **Name** property, type a different name, and press Enter.

3. To specify a burst table column as the data item, do the following:
   - In the **Insertable Objects** pane, on the source tab, expand the burst table.
   - Drag the data item you want to use 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 what data item to burst on and who the recipients are.

Before you set burst options, ensure that the data item you intend to burst on is in the report and grouped. The grouped column will create the appropriate subsets of data. In addition, you must associate the burst key with this level of grouping.

Steps

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 you want to burst on.

   Tip: You can choose a query that does not appear in the layout. This is useful when you want to distribute the same report to all burst recipients.
4. In the **Label** box, click the data item to be used to label each burst report.
5. Click the edit button.
6. In the **Data Items** box, drag the data item that you want to burst on 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 use to burst the report.
   - Click **Email addresses** to distribute reports by email.
   - Click **Directory entries** to distribute reports to a directory that recipients can access in Cognos Connection.

     When a recipient logs into Cognos 8, he will see only the report that is specific to him.
   - Click **Automatic** to let Cognos 8 determine from the data item whether to email reports or send them to a directory.
10. If the report contains two nested data containers, such as a list and a chart, click the ellipsis points (...) beside **Master Detail Relationships**, and define the relationship between the containers.

    For information about master detail relationships, see "Create a Master-Detail Relationship" (p. 175).
11. Click **OK**.

Enable Bursting

When the report is ready to be distributed, enable bursting for the report in Cognos Connection.

Steps

1. Locate the report in Cognos Connection.
2. Under **Actions**, click **Run with options**.
3. Click the **Advanced options** link.
4. Select the **Burst the report** check box.
5. 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 directory, you can also send the report by email if the 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 it is entered in the recipient's personal information in Cognos Connection.
6. If the burst report contains a drill-through link (p. 168) to another report and you are distributing the report by email, do the following:
   - Click the **Edit the email options** link.

- Select the **Include a link to the report** check box and click **OK**.

  If you do not select the check box, the drill-through links in the burst report will not work.

7. Click **Run** to run the report.

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 Cognos Connection or access their email accounts, they will see only the data that is meant for them.

Creating Burst Reports Using a Dimensional Data Source

You can burst a report using a dimensional data source by using burst information stored in the data source. We assume that you do not want to append bursting information to existing dimensional data sources. The solution is to create a relational data source that contains the burst information.

To burst a report using a dimensional data source, do the following:

- ❑ In 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*.

- ❑ In Report Studio, create a master-detail report in which the master query drives the report of interest and the detail query contains the burst information.

  The master query must be grouped by a data item that you are bursting on and that has 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 one 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 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" (p. 175).

- ❑ Ensure that the detail query, which must be evaluated by Cognos 8 when the report is executed, is not visible:
  - Place a list that is based on the detail query in a conditional block for which the box type is set to **None**.
  - Link the master and detail queries using the following expression:

    `[Master Burst Key] = [Detail Burst Key]`

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 Great 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 do not need to see data for everyone, only the data that pertains to them. You create a list report that you burst to each sales representative.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.

5. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:
   - **Staff name**

- **Product line**
- **Product type**
- **Product name**
- **Quantity**
- **Revenue**

Tip: You can find these data items in the **Sales reps**, **Products**, and **Orders** folders.

6. Group the **Staff name**, **Product line**, and **Product type** columns.
7. Click the **Staff name** column, and then click the create header button.

   Staff name appears as a header in the list. You no longer need to keep the data item as a list column.

8. In the list, click **Staff name** and click the delete button.
9. Click **Revenue**, click the aggregate button, and click **Total**.

10. Pause the pointer over the query explorer button and click **Query1**.
11. In the **Insertable Objects** pane, on the source tab, drag **Email** from the **Sales reps** folder to the **Data Items** pane.
12. From the **File** menu, click **Burst Options**.
13. Select the **Make report available for bursting** check box.
14. Under **Burst Groups**, in the **Query** box, click **Query1**.
15. In the **Label** box, click **Staff name**.

16. Click the edit button.
17. In the **Data Items** box, drag **Staff name** to the **Groups** folder and click **OK**.
18. Under **Burst Recipient**, in the **Query** box, click **Query1**.
19. In the **Data Item** box, click **Email**.
20. In the **Type** box, click **Email addresses**.
21. Click **OK**.
22. Save the report.
23. Locate the report in Cognos Connection.

24. Under **Actions**, click **Run with options**.
25. Click the **Advanced options** link.
26. Select the **Burst the report** check box.
27. Select the **Send the report by email** check box.
28. Click **Run** and then click **OK**.

When sales representatives access their email accounts, they will see a report with only the data that is meant for them.

Customizing Reports 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 variable 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. You can also use conditional formatting to highlight exceptional data in a report.

All conditional processing is based on report variables. There are three types of report variables: string, boolean, and report language.

You can create and modify variables in the condition explorer. Alternatively, you can create a variable when you define conditional processing for a specific object, such as a style variable, or when you define a conditional layout. However, until you gain experience using variables, working in the condition explorer may help you to understand and organize the variables included in the report. All variables appear in the condition explorer regardless of where they were created. You must work in the condition explorer to modify an existing variable.

Add a Variable

Before you can add conditional formatting or conditional rendering to your report, you must add a variable. You can add a variable in the condition explorer or in the **Properties** pane.

Steps in the Condition Explorer

1. Pause the pointer over the condition explorer button and click **Variables**.
2. In the **Insertable Objects** pane, 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 dollars and the value No if revenue is greater than or equal to one million.

   ```
   [Revenue]<1000000
   ```

   For information about creating expressions, see "Using the Expression Editor" (p. 233).
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 dollars and the value low if revenue is less than or equal to one million dollars.

     ```
     if ([Revenue]>1000000) then ('high') else ('low')
     ```

     For information about creating expressions, see "Using the Expression Editor" (p. 233).

   - Click the add button under 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.
   - Click **OK**.

     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.
5. If you created a language-specific variable, in the **Languages** dialog box, select the languages you want to support and click **OK**.

   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 all of the different French languages available.

Steps in the Properties Pane

1. Select the object that you want.
2. In the **Properties** pane, under **Conditional**, double-click the conditional property that you would like to assign the variable to.

Goal	Conditional Property
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.	**Text Source Variable**
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.	**Render Variable**
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.	**Style Variable**
Specify a variable based on which objects inserted into a block can be conditionally rendered. Applies only to conditional block objects that you insert in a report (p. 88).	**Block Variable**

3. Click **Variable** and click one of the following variable types:
   - **<New language variable>**
   - **<New string variable>**
   - **<New boolean variable>**

   Tip: To use an existing variable, you can select it here.
4. In the **New Variable** dialog box, type the name of the variable in the **Name** box.

5. If you created a string variable, click the add button and type the string values you want to define, and click **OK**.
6. If you created a language variable, select the languages you want to support, and click **OK**.
7. In the **Expression Definition** box, define the condition and click **OK**.

Adding Conditional Formatting to a Report

Enhance a report by adding conditional formatting. For example, use conditional formatting to highlight exceptional data in a report.

With conditional formatting, you can

- hide and show objects
- highlight data

Hide and Show Objects

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 (p. 160).

Steps

1. Create a variable and define the condition that determines if the object is to be shown or hidden.

   Tip: Create a boolean variable to show and hide objects, as this type of variable has only two possible values.

2. In the **Insertable Objects** pane, on the toolbox tab, drag the **Conditional Blocks** object to the work area.
3. Click the conditional block.
4. In the **Properties** pane, double-click the **Block Variable** property.

5. Click **Variable**, click the variable you created, and click **OK**.
6. Click the **Current Block** property and click **Yes**.
7. In the **Insertable Objects** pane, drag the object you want to show or hide in 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 (p. 94) before you can add a data item to the block.

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.

Highlight Data

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.

Steps

1. Create a variable and define the condition that determines if the data will be highlighted.
2. In the work area, click the column that you want to highlight based on the condition you created.
3. In the **Properties** pane, double-click the **Style Variable** property.
4. Click **Variable**, click the variable you want to assign to the object, and click **OK**.
5. If you assigned a string variable, in the **Values** box, select the values you want 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 you want 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 one of the possible values for the variable other than the default value.

   Tip: When you select a value in Condition Explorer, the Explorer bar becomes green to indicate that conditional formatting is turned on, and that any changes you make to the report applies only to the variable value.

   For example, if you created a boolean variable, click the **Yes** value.
9. In the **Properties** pane, specify the formatting that you want to highlight the column with when the condition is satisfied.

   For example, click the **Border** property to create a thicker border around the column.
10. Repeat steps 9 to 10 for other possible values defined for the variable.

    Tip: When pausing the pointer over the condition explorer button, click (**No variable**) to view how the report looks when no variable is applied. Or you can triple-click the Explorer bar.

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.

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 showing or 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 "Report Studio Object and Property Reference" (p. 351).

Steps

1. Select the list column to be rendered conditionally.

   Tip: You must select the list column, not list column body or list column title. If the body or title is selected, as indicated in the **Properties** pane, use the select ancestor button to select the list column.

2. In the **Properties** pane, double-click the **Render Variable** property.
3. Click **Variable** and click the variable that will be used to determine if the column is to be rendered.
4. In the **Render for** box, select the values you want the condition to support and click **OK**.

   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 Great 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 will also ask whether the user wants to see a description for each order.

Steps

1. In the Cognos Connection **Welcome** page, click the **Public Folders** link.
2. Click the **GO Sales and Retailers** link, and then click the **Report Studio** link in the upper-right corner of the page.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.

5. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:
   - **Order date**
   - **Order number**
   - **Product name**
   - **Description**
   - **Quantity**
   - **Unit sale price**
   - **Revenue**

   Tip: You can find these data items in the **Products** and **Orders** folders.

6. Click **Order Date** and click the section button.

7. Group the **Order Number** column by selecting the column and clicking the group button.
8. Click **Revenue**, click the aggregate button, and click **Total**.

9. Change the title of the report to **New Orders**.
10. Pause the pointer over the page explorer button and select **Prompt Pages**.
11. Create a new prompt page by double-clicking **Pages** in the **Insertable Objects** pane.

12. Double-click the new prompt page.
13. In the **Insertable Objects** pane, on the toolbox tab, double-click **Text Item** and type the following text:

    Enter the start date, and select if descriptions will be shown.
14. Insert a 2 by 2 table into the prompt page using the insert table button, and moving the pointer until four squares are highlighted in a 2 by 2 pattern.
15. In the **Insertable Objects** pane, on the toolbox tab, drag **Text Item** into the upper-left cell and type the following text:

    Starting Date
16. In the **Insertable Objects** pane, drag a **Text Item** into the lower-left cell, and type the following text:

    Show Descriptions
17. In the **Insertable Objects** pane, drag a **Date Prompt** into the upper-right cell.

18. When prompted with the **Prompt Wizard, Choose a Parameter** window, select **Create a new parameter** and type **p_Date** in the space provided, then click **Next**.
19. When prompted in the **Create Filter** window, select **Create a parameterized filter** with the following entries:
    - For **Package item**, click the ellipsis points (...) and click **[gosales_goretailers].[Orders].[Order date]**.
    - For **Operator**, click **>**.
20. Click **Finish**.
21. In the **Insertable Objects** pane, drag a **Value Prompt** into the lower-right cell.
22. When prompted with the **Prompt Wizard, Choose a Parameter** window, select **Create a new parameter** and type **p_ShowDescn** in the space provided, and then click **Finish**.
23. Select the **Value Prompt**, and in the **Properties** pane, double-click **Static Choices**.
24. Click **Variable**, click **<New boolean variable>**, and when prompted, type the name **showDesc**.
25. Click the add button.
26. In the **Edit** dialog box, type **Yes** in both the **Use** and **Display** boxes.
27. Click the add button.
28. In the **Edit** dialog box, type **No** in both the **Use** and **Display** boxes.
29. Click **OK**.
30. Pause the pointer over the condition explorer button and click **Variables**.
31. Click the **showDesc** variable, and in the **Properties** pane, double-click **Report Expression**.
32. In the **Report Expression** dialog box, type the following in the **Expression Definition** window:

    ParamDisplayValue("p_ShowDesc") = 'Yes'
33. Click **OK**.
34. Pause the pointer over the page explorer button and click the report page.
35. Click the **Descriptions** column.
36. In the **Properties** pane, select the list column by clicking the select ancestor button and selecting **List Column** from the context menu.
37. In the **Properties** pane, double-click the **Render Variable** parameter and select the **showDesc** boolean variable you created in steps 24 to 33.
38. Click **Run**.

The report will prompt you for a date, and will then provide orders that occur after the date you entered. The report will also ask if the **Descriptions** column is to be shown, and the column will be rendered only if you choose **Yes** to this selection.

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.

Steps

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.
4. Click **OK**.

A layout is created for each value you selected. Pause the pointer over the page explorer button to navigate the different layouts. For each layout, click the **Report Pages** link to create a report page, or click the **Prompt Pages** link to create a prompt page (p. 119), and add the objects you want.

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" (p. 158).

Using Cognos 8 to Create Multilingual Reports

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 Cognos 8 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 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 Windows utility named Map Manager to import maps and update labels for maps in Report Studio. For map features such as country 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, the report author can specify that text, such as the title, appears in German when the report is opened by a German user. Report authors can also add translations for text objects, and create other language-dependent objects.

  For more information, see the Report Studio *User Guide*.

- ❑ Specify the language in which a report is viewed.

  You can use 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 Cognos Connection *User Guide*.

The data then appears in the language and with the regional settings specified in

- the user's Web browser options
- the run options
- the Cognos Connection preferences

Any text that users or authors add appears in the language in which they typed it.

Create a Multilingual Report in Report Studio

To create a multilingual report in Report Studio, do the following.

If you want the report to show data in different languages, the model must also be multilingual.

Steps

1. Create a report language variable.
2. In the work area, click the object that you want to modify based on one of the languages you selected.
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 you want 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 click one of the possible languages for the variable.

   Tip: When you select a value in Condition Explorer, the Explorer bar becomes green to indicate that conditional formatting is turned on, and that any changes you make to the report applies only to the variable value.
7. In the **Properties** pane, specify the formatting that you want 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 7 to 9 for all other languages specified for the variable.

   Tip: When pausing the pointer over the condition explorer button, click (**No variable**) to view how the report looks when no variable is applied. Or you can triple-click the Explorer bar.

When you run the report, the report objects to which you applied the variable are formatted according to the browser's language.

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 using prompt parameters, and then using information from the source report to provide answers to those prompts. An important part of defining a drill-through path is defining which values to use for each prompt. If a prompt is answered by the drill-through value passed to it, the prompt page does not appear.

In Cognos 8, you can drill through

- between reports created in different packages against different data source types

  For example, you can drill through from an analysis against a cube to a detailed report against a relational data source.
- from one existing report to another report using Report Studio (p. 168)
- between reports in Report Studio, Query Studio, or Analysis Studio, exploring data in many ways

  For more information, see the user guide for the studio you are working in.
- from Cognos Series 7 to Cognos 8 (p. 447)
- from Metric Studio to other Cognos 8 reports by passing parameters using URLs

  For more information, see the Metric Studio *User Guide*.

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.

Model-based vs. Report-based Drill-through Access

The target of drill-through access is always a saved report definition. The report can be created in Report Studio, Query Studio, or Analysis Studio.

Model-based drill-through refers to a drill-through path from any report created in the source package to any target report in Cognos Connection. The drill-through path is stored as a drill-through definition under the source package. All source reports in this package can access the drill-through definition.

For any target report that contains parameters, the target parameters must be mapped to the metadata in the source package. This ensures that the values from the source report are passed to the correct parameter values, and that the target report is filtered correctly.

A model-based drill-through can be performed between any combination of Report Studio, Query Studio, and Cognos Viewer reports. This type of drill-through can be used when you want to drill from Analysis Studio or Query Studio to any of the Report Studio, Query Studio, or Cognos Viewer target reports in the same, or different package.

Report-based drill-through refers to a drill-through path created and contained in a Report Studio source report. The drill-through path is associated with a specific data column, chart, or cross tab, and is available only when users select that area of the report. The drill-through definition, if it exists, appears as a hyperlink in the source report. Usually this area of the report shows the data to be passed as a parameter. However, other values can be passed as well, or instead.

The report-based drill-through is limited to Report Studio source reports and any target reports. This type of drill-through can be used when you want to pass the parameter results within a source report to the target report, the results of a report expression to a target report, or a URL link as a part of the drill-through.

Drilling Through to Different Report Formats

When users follow a drill-through path, they usually run the report filtered by the drill-through parameters. They then see the results in Cognos Viewer as an HTML Web page. However, there are other options.

Reports can be output as HTML Web pages, or as PDF, XML, CSV, or Excel 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 or Query Studio, the report can be run and opened in its studio instead of in 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. Drilling through to Report Studio is not supported, because Report Studio does not show any data.

You can define an application where a dashboard style report of high-level data can be drilled through to an Analysis Studio view to investigate problems. The Analysis Studio view can then be drilled through to a PDF report for printing.

Bookmark References

When you drill through, the values that you pass are usually, but not always, used to filter the report using prompts. Cognos 8 Business Intelligence supports bookmarks within PDF reports so that a user can scroll a report to view the relevant part based on a URL parameter. For example, you may define a report with one page per product and add a bookmark of the product number on each page. Report consumers can then select a bookmark to see the product that they want.

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 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 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) (p. 166). 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 Cognos 8, 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 and must also be from a conformed dimension (p. 167).

If the target parameter is a value, the source can be either a value or a member. The source can be a member if the level is mapped to the data item correctly, or if the business key from which the member is sourced matches the business key in the relational target. For more information, see "Conformed Dimensions" (p. 167).

Member Unique Names

The member unique name (MUN) is a unique identifier for a member in 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.

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, cube 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:

- 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 Framework Manager in the production environment.
- Resolve the non-unique keys in a dimension in the data source.

Ensure that there are no duplicate source values in all levels of a dimension before you build the PowerCube. We do not recommend using the tilde character (~) in the category codes.

For more information, see the section about uniqueness in the Cognos Series 7 *Step-by-Step Transformer*.

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 Cognos PowerCube as the source and the target. In the previously mentioned example of the Revenue and Inventory data stores, 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.

Business Keys

When performing a drill-through from a member to a relational value, the business key of the member is passed. This means that your relational target parameter must be set up using the business key data 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 Cognos PowerCubes, the business key value is the **Source** property defined for the level in Cognos Series 7 Transformer.

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 text that is familiar to the users, but filters on the business key.

Drilling Through on Dates Between PowerCubes and Relational Packages

Usually, drilling through from OLAP to relational packages requires that the target report parameter is set using the business key in the relational data. However, this method does not work well for dates. OLAP data sources typically view dates as members, such as Quarter 1 2006, while relational data sources view dates as ranges, such as 1/Jan/2006 to 31/March/2006.

A special feature exists for drilling through between PowerCubes and relational packages. Ensure that the target report parameter is set up using `in_range`. Here is an example:

```
[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 data level is not set to suppress blank categories. Enabling this option may cause the drill-through on dates to be unsuccessful because there are missing values in the range.

Drilling Through Between Packages

You can set up drill-through access between different 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.

Source data source	Target data source
OLAP	OLAP **Note:** OLAP to OLAP drill through is supported only if the data source type is the same, for example, MSAS to MSAS.
OLAP	Dimensionally modeled relational
OLAP	Relational **Note:** For more information, see "Business Keys" (p. 167).
Dimensionally modeled relational	Dimensionally modeled relational
Dimensionally modeled relational	OLAP
Dimensionally modeled relational	Relational
Relational	Dimensionally modeled relational
Relational	Relational **Note:** For more information, see "Business Keys" (p. 167).

Scope

Scope is specific to model-based drill-through access. It defines when the drill-through path is made available, or 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 the parameters that it passes. For example, if the drill-through path passes an employee number to an employee report, you make the drill-through path available only when the user's context includes an employee.

Scope is not required for report-based drill-through access where the drill path is associated with a specific report column, which serves as the scope for the drill path.

Set Up Drill-through Access in a Report

Set up drill-through access in a report to link two reports containing related information. You can then access related or more detailed information in one report (the target) by selecting a data item value from another report (the source). In addition, you can pass parameter values from the source report to the target report. This means that you can filter the target report using the same prompted filter values in the parent report.

You can also drill through within the same report by creating bookmarks (p. 85), and create drill-through definitions in the package. Users can use package drill-through definitions to navigate to a target report from an Analysis Studio analysis, a Query Studio report, or a Report Studio report. For more information, see the *Administration and Security Guide*.

If you have the Cognos 8 software development kit (SDK) (p. 42), you can use URLs to set up drill-through access to and from third-party sources.

You can also drill through from a map (p. 77).

If you are using an SAP BW data source for the target report, and if 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, ensure that you have a report that will serve as the source report and another report that will serve as the target report.

Steps

1. Open the target report.
2. Create a parameter that will serve as the drill-through column or that will be used to filter the report.

   For example, to drill through or filter Product line, create a parameter that looks like this:

   `[Product line]=?prodline_p?`

   Tip: Use the operators `in` or `in_range` if you want the target report to accept multiple values or a range of values.
3. In the **Usage** box, do one of the following:
   - To specify that users must click a value in the source report, click **Required**.

     If users do not click a value, they will be prompted to choose a value before the target report appears.
   - To specify that users do not need to click a value in the source report, click **Optional**.

     If users do not click a value, they are not prompted to choose one.
   - To specify not to use the parameter, click **Disabled**.

     Clicking a value in the source report has no effect on the target report.
4. To create other parameters, repeat steps 2 to 3.
5. Open the source report.
6. To allow drill-through from the package, do the following:
   - From the **Data** menu, click **Drill Behavior**.
   - Select the **Allow drill through from a package** check box and click **OK**.
7. Click the object that will serve as the drill-through object.

   For example, to pass a data item value to the target report, click a 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. This means that you can drill from outside the context of a query.
8. Click the drill throughs button.
9. Click the new drill through 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**.
10. On the **Target report** tab, click the ellipsis points (...) next to the **Report** box, and select the drill-through or target report.

11. Under the **Parameters** box, click the edit button.

    Each required and optional parameter defined in the target report appears in the **Parameters** dialog box.
12. For each parameter, click **Method** and do one of the following:
    - To specify not to pass any value, click (**Default**).

      That is, the default method is to not pass any value.
    - To specify not to use this parameter, click **Do not use parameter**.

      The target report is 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 that you want.

      Values for the selected data item are passed to the target report.
    - To pass values from a parameter, click **Pass parameter value**, click **Value**, and then click the parameter that you want.

      Values for the selected parameter are passed to the target report.
13. Click **OK**.

14. In the **Action** box, decide how the target report will be viewed when users click the drill-through column in the parent report:
    - To run the target report, click **Run the report**.

      Run the target report when you want to see the latest data.
    - To view the most recently saved output version of the target report, click **View the most recent report**.

      Saved output versions are created in Cognos Connection.
    - To use the default action specified for the report in Cognos Connection, click (**Default**).
15. If you chose to run the target report in the previous step, in the **Format** box, click the output format you want for your report.

    Tip: Click (**Default**) to run the report using the default format specified for the report in Cognos Connection.
16. If you want the target report to appear in a new window, select the **Open in new window** check box.
17. Click **OK**.

The drill-through object appears as a link with a plus sign (+) next to it. When users run the source report, they can click the object to drill through to the target report. If more than one target report is available, the **Go To** page appears with the list of available targets.

To view package drill-through definitions, users must right-click a value, click **Go To**, and then click **Related Links**.

Tip: If you run the target report directly, you are prompted to select a value for the report to run on.

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.

Steps

1. Right-click the drill-through object and click **Drill Throughs**.
2. If more than one drill-through exists for the object, in the **Drill Throughs** box, click a drill through.
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 **Values**, and click one of the variable's possible values.
5. In the **Source type** drop-down list, click the source type you want to use.
6. If the source type is **Text**, click the ellipsis points (...) beside the **Text** box and type the text you want.
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 points (...) beside 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.
10. Click **OK**.

When users run the parent 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 - Create a Drill-through Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that lists product sales by order method for each product line, and allows users to drill through from the sales report to view the product details for any item selected. You will create two reports, one that contains the details for the item, and another that lists the product sales for the order methods selected by users.

Steps

1. In the Cognos Connection **Welcome** page, click the **Report Studio** link.
2. Select the package **GO Sales and Retailers**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **List** and click **OK**.

5. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:
   - **Order method**
   - **Product line**
   - **Product type**
   - **Product name**
   - **Description**
   - **Introduction date**
   - **Product image**

   Tip: You can find these data items in the **Orders** and **Products** folders.
6. Group the **Order method**, **Product line**, and **Product type** columns.

7. Click the filters button.
8. Click the add button, and type the following in the **Expression Definition** box:

   [Product name]=?p_PN?
9. Click **OK**.
10. Repeat steps 8 to 9 to create the following filter:

    [Order method] in ?p_OM?
11. Change the title of the report to **Product Details**.
12. Save the report as **Product Details**.
13. From the **File** menu, click **New** to create a new report.
14. In the **New** dialog box, click **List** and click **OK**.
15. In the **Insertable Objects** pane, on the source tab, add the following data items to the list by double-clicking them:
    - **Order method**
    - **Product line**
    - **Product type**
    - **Product name**
    - **Quantity**
    - **Revenue**

    Tip: You can find these data items in the **Orders** and **Products** folders.
16. Group the **Order method**, **Product line**, and **Product type** columns.

17. Click the filters button.
18. Click the add button, and type the following in the **Expression Definition** box:

    [Order method] in ?p_OM?
19. Click **OK**.
20. Right-click the **Product Name** column and click **Drill Throughs**.

21. Click the new drill through button.

22. Under **Report,** click the ellipsis points (...) and select the **Product Details** report you created previously.
23. Click **Action** and click **Run the report**.
24. Click **Format** and click **HTML**.
25. Click the edit button.
26. For the item **p_OM**, under **Method**, select **Pass parameter value**, and select **p_OM** for the **Value**.
27. For the item **p_PN**, under **Method**, select **Pass data item value**, and select **Product name** for the **Value**.
28. Click **OK** twice.
29. Change the title of the report to **Product Revenue by Order Method**.
30. Save the report as **Product Revenue by Order Method**.
31. Click the run report button.

When the report is run, you are prompted to select one or more order methods. When you click **OK**, the list will show the product names as clickable links. When a product is clicked, the second report will run, showing the order methods and product selected in the first report.

Create a Drill-up/Drill-down Report

You can create a report that allows the reader to drill down to lower level data sets or drill up to higher-level data sets, provided you are using a Dimensionally Modeled Relational (DMR) data source. This will allow you to rerun the report for a different data set without having to regenerate the report.

An example of a dimensional hierarchy might be:

Years - Year - Quarter - Month

Drilling up and down allows you to view more general or more detailed information on your data within such a predefined dimensional hierarchy, without having to create different reports.

Before you begin, ensure that you are using a dimensional data source.

Steps

1. From the **Data** menu, click **Drill Behavior**.
2. On the **Basic** tab, in the **Report drill capabilities** box, select the **Allow drill-up and drill-down** check box.

   By default, Report Studio determines which items can be drilled on, based on the dimensional structure.
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.

Behavior name	Drill-up behavior	Drill-down behavior
Preserve	The data item's value will remain unchanged.	The data item's value will remain unchanged.
Empty Set	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.	The data item will be removed from the report.

Behavior name	Drill-up behavior	Drill-down behavior
Replace Item	The data item's value will change to become the parent of the item drilled on (if a lowest-level summary was drilled up on), or the grandparent (if a lowest-level detail of a dimension was drilled up on).	The data item's value will change to become the item drilled on.
Replace Expression	The data item's value will change to become the children of the parent of the item drilled on (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).	The data item's value will change to become the children of the item drilled on.
Ancestor	The data item's value will change to become the data value of the ancestor "n generations higher" in the dimensional hierarchy from the item drilled on. The number of generations or levels is determined by the **Depth** value.	The data item's value will change to become the data value of the ancestor "n generations higher" in the dimensional hierarchy from the item drilled on. The number of generations or levels is determined by the **Depth** value.
Depth Based Expression	The data item's value will change to become all data items "n generations higher" in the dimensional hierarchy from the item drilled on. The number of generations or levels is determined by the **Depth** value.	The data item's value will change to become all data items "n generations higher" in the dimensional hierarchy from the item drilled on. The number of generations or levels is determined by the **Depth** value.
Change Expression	The data item's value will change to become the value of some other data item in the query.	The data item's value will change to become the value of some other data item in the query.

5. Click **OK**.

The report will generate links for any item that can be drilled up or down on.

You can perform 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.

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, where member sets that are 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 no member set is defined for an item, the Cognos 8 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 a Member Set

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.

Steps

1. Pause the pointer over the query explorer button and click the query in which you want to create a member set.
2. In the **Properties** pane, click the **Define Member Sets** property and click **Yes**.
3. Click the **Member Sets** tab.
4. In the **Insertable Objects** pane, drag the items that you want 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.

   You can also nest member sets.

Example - Create a Drill-up/Drill-down Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that lists product sales for each product line. The user must be able to drill-down to retrieve data on the same criteria with a more detailed scope.

Steps

1. In the Cognos Connection **Welcome** page, click the **Report Studio** link.
2. Select the package **Great Outdoors Company**.
3. In the **Welcome** dialog box, click **Create a new report or template**.
4. In the **New** dialog box, click **Crosstab** and click **OK**.

5. In the **Insertable Objects** pane, on the source tab, add **Year** followed by **Years** to the columns section.
6. Add **General Manager** followed by **Product Line** to the rows section.
7. Insert **Products** as a total row in the crosstab.
8. Insert **Revenue** as the measure for the report.
9. From the **Data** menu, click **Drill Behavior**.
10. On the **Basic** tab, select the **Allow drill up and down** check box.
11. Click **OK**.
12. Change the title of the report to **Sales Report: Product Revenue by Period**.
13. Save the report.
14. Run the report.

When the report is rendered, the list shows the sales manager, the product lines, and even the years as links. When you right-click a link, a context menu allows you to drill-up or drill-down to run the report from a higher or lower level scope of that dimension. Notice how the **Years** and **Products** total columns are also updated.

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, on the page for the detail query.

Note that 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.

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.

Steps

1. To use a parent frame for the master query and a nested frame for the detail query, do the following:
   - In the **Insertable Objects** pane, on the toolbox tab, drag a **List**, **Repeater Table**, or **Repeater** to the report.
   - Click **OK** to create a new query.
   - Add a second data container to the object you previously 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.
   - Click **OK** to create a second query.
   - Add the data items you want to both data containers.

2. To associate a report page with the master query and use a data container on the page for the detail query, do the following:
   - Click anywhere in the report page.
   - In the **Properties** pane, click the select ancestor button and click **Page**.
   - Click the **Query** property and then choose a query from the drop-down list.
   - In the **Insertable Objects** pane, on the toolbox tab, drag a data container to the report.
   - Click **OK** to create the detail query.

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 when you want to filter values at a lower level in the detail query.
   - Pause the pointer over the query explorer button and click the detail query.
   - In the **Insertable Objects** pane, on the toolbox tab, drag the **Filter** object to the **Detail Filters** box.
   - In the **Expression Definition** box, create the parameter (p. 123).
   - Click **OK**.

4. Pause the pointer over the page explorer button, and click the report page.
   - Click anywhere in the report page.
   - In the **Properties** pane, click the select ancestor button and click **Page**.
5. Click the data container containing the details.
6. From the **Data** menu, click **Master Detail Relationships**.
7. Click the **New Link** button.

8. In the **Master Query** box, click the data item that will provide the primary information.
9. To link the master query to a data item, in the **Detail Query** box, click the data item that will provide the detailed information.
10. To link the master query to a parameter, in the **Parameters** box, click the parameter that will provide the detailed information.
11. Repeat steps 8 to 11 to create other links.

    Tip: To delete a link, click it and press Delete.
12. Click **OK**.

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.

Link Members from Two Dimensional 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" (p. 235).

For example, you may want to link the Americas member in the Sales territory level in the Great Outdoors cube with Americas in the GO Subsidiary level in the Great Outdoors Finance cube. To create the relationship, you must alias the levels that contain the members that you want to link.

Steps

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:

   `caption(`*`expression`*`)`

   For example, `caption([great_outdoors_company].[Sales Territory].[Sales Territory].[Sales territory])`
3. Click **OK**.
4. Repeat steps 1 to 3 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.

You can now create the master-detail relationship between the two levels.

Chapter 6: Try It Yourself Exercises

If you have some experience with Report Studio and want to improve your skills in report writing, this chapter is for you. Each topic gives you some guidelines on how to create each sample report. If you need help, links to more detailed instructions are available.

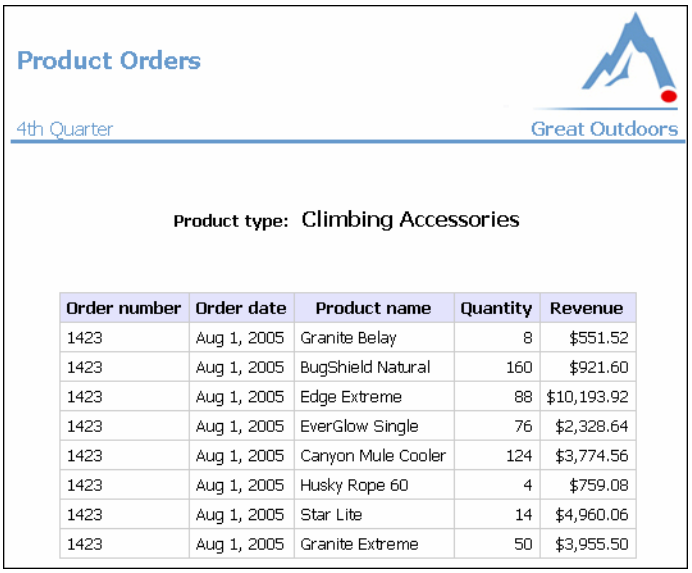
If you have not used Report Studio before, see the Report Studio Tour first to learn basic skills.

Try It Yourself - Create a List Report

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

In this topic, you learn how to create a list report that shows revenue for each product for the last quarter of 2002.

It should take 15 to 20 minutes to complete this topic, and your report will look like this.



Product Orders

4th Quarter

Great Outdoors

Product type: Climbing Accessories

Order number	Order date	Product name	Quantity	Revenue
1423	Aug 1, 2005	Granite Belay	8	$551.52
1423	Aug 1, 2005	BugShield Natural	160	$921.60
1423	Aug 1, 2005	Edge Extreme	88	$10,193.92
1423	Aug 1, 2005	EverGlow Single	76	$2,328.64
1423	Aug 1, 2005	Canyon Mule Cooler	124	$3,774.56
1423	Aug 1, 2005	Husky Rope 60	4	$759.08
1423	Aug 1, 2005	Star Lite	14	$4,960.06
1423	Aug 1, 2005	Granite Extreme	50	$3,955.50

Steps

1. Create a new list report that uses the sample package named **GO Sales and Retailers** and the sample template named **GO-list**.
2. Add the following data items to the report:
   - **Order number** (in **Orders**)
   - **Order date** (in **Orders**)
   - **Product type** (in **Products**)
   - **Product name** (in **Products**)
   - **Quantity** (in **Orders**)

   Tip: Use the source tab in the **Insertable Objects** pane.
3. Create this query calculation named **Revenue**:

   [gosales_goretailers].[Orders].[Unit price] * Quantity

   Tip: Use the toolbox tab in the **Insertable Objects** pane.

4. Group the **Product type** column to make the report easier to read. Then group the **Order date** column.
5. Make the **Product type** column appear as a section header in the report.
6. Bold **Product type** in the body of the report.
7. Remove the **Product type** column title without removing **Product type** from the body of the report.
8. Add a total to the **Revenue** column to view total revenue for each product type.
9. Create this tabular filter to view revenue for the last quarter of 2004.

   Order date between 2004-10-01 and 2004-12-31
10. Change the **Data Format** property for the **Order date** column to be date only, not date and time.
11. Change the title text placeholder to this text:

    Product Orders
12. Add the following text to the bottom cell, replacing the date object:

    4th Quarter
13. Format the text as Tahoma, 11 pt, and apply the Web-safe color #6699CC.
14. Run the report to view what it will look like for your users.

Need More Help?

- Add Data to a Report
- Add a Header or Footer
- Add a Summary
- Create a Calculation
- Filter Data
- Group Data
- Run a Report

Try It Yourself - Create a Crosstab

Use crosstab reports to compare information that uses one or more criteria. The values at the intersection points of rows and columns show summarized information.

In this topic, you learn how to create a crosstab that shows the cost of goods sold for each product line and product type by quarter.

It should take 15 to 20 minutes to complete this topic, and your report will look like this.

Product Line Profitability

Jun 16, 2005

		Quarter 1				
		Revenue	Cost of Goods Sold	Gross profit	Gross Profit Margin	Revenue
Camping Equipment	Cooking Gear	$275,929.98	$188,731.64	$87,198.34	$68.40	$240,230.24
	Lanterns	$577,074.10	$387,926.78	$189,147.32	$67.22	$434,906.52
	Packs	$768,014.96	$526,788.08	$241,226.88	$68.59	$701,010.02
	Sleeping Bags	$665,818.24	$454,255.48	$211,562.76	$68.23	$462,310.80
	Tents	$2,329,489.68	$1,913,626.60	$415,863.08	$82.15	$2,136,008.34
Golf Equipment	Golf Accessories	$39,167.78	$18,713.30	$20,454.48	$47.78	$45,637.44
	Irons	$431,098.78	$228,206.50	$202,892.28	$52.94	$412,999.48

Steps

1. Create a new crosstab that uses the sample package named **GO Sales and Retailers** and the sample template named **GO-crosstab**.
2. Add the following data items to the report:
   - **Order year** (in **Orders**) in the columns
   - **Product line** (in **Products**) in the rows
   - **Revenue** (in **Orders**) as the first measure
   - **Gross profit** (in **Orders**) as the second measure

   Tip: Use the source tab in the **Insertable Objects** pane.
3. Nest **Product type** (in **Products**) after **Product line** in the rows.
4. Nest **Order month** (in **Orders**) under **Order year** in the columns.
5. Change the **Expression** property for **Order month** to:

   mod([gosales_goretailers].[Orders].[Order month],4)+1

   Tip: Use the toolbox tab in the **Insertable Objects** pane.
6. Create this query calculation named **Cost of Goods Sold**:

   [Revenue]-[Gross Profit]
7. Change the **Data Format** property for **Revenue, Gross profit,** and **Cost of Goods Sold** to currency.
8. Change the blue text placeholder to this text:

   Profitability by Product Line
9. Run the report to view what it will look like for your users.

Need More Help?

- Add Data to a Report
- Create a Calculation
- Create a Nested Crosstab Report
- Crosstab Reports
- Run a Report

Try It Yourself - Create Charts

Use charts to reveal trends and relationships that are not evident in tabular reports.

Report Studio gives you flexibility in how you organize data in charts.

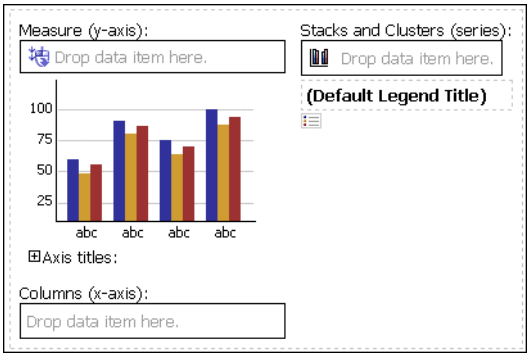
In this topic, you learn how to create charts that show different information.

It should take 15 to 20 minutes to complete this topic.

Let's take a look at the basic chart interface.

Steps to Create a Basic Column Chart

1. Open Report Studio with the **GO Sales and Retailers** package.
2. Create a new **Chart** report.
3. Click **OK** to open the default column chart.



You can click on the white space around the chart to select the whole chart, or you can click on an individual item, such as an axis icon or chart type icon, to find a given chart property.

The **Series** drop zone defines the items that appear in the legend. This is the legend edge. The **Category** (**x-axis**) drop zone defines the ordinal edge.

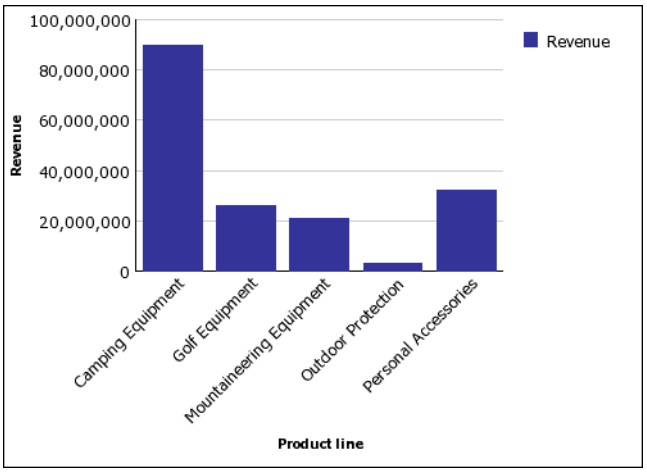
Let's chart some GO Sales and Retailers data.

Steps to Add Data to a Chart

1. Drag the following items to the chart:
   - **Revenue** (in **Orders**) to the **Measure** drop zone
   - **Product line** (in **Products**) to the **Category** (**x-axis**) drop zone

   Tip: Use the source tab in the **Insertable Objects** pane.

   Column, bar, line, area, and clustered charts are based on the same combination chart for flexibility in charting.
2. Run the report to see your chart.



3. Return to Report Studio.

We want to show the revenue value on each column in the chart.

Steps to Show Values in a Chart

1. Click the bar icon in the **Series** drop zone.
2. In the **Properties** pane, under **Chart Labels**, change the **Values** property to **Show**.
3. Run the report to see your chart.

   You can customize the font properties for the values shown in the chart, independent of the axis labels.
4. From the **Structure** menu, clear **Lock Page Objects** to unlock the chart objects.
5. Select the chart body by clicking between the axis.
6. In the **Properties** pane, increase the font size and change the foreground color to red.

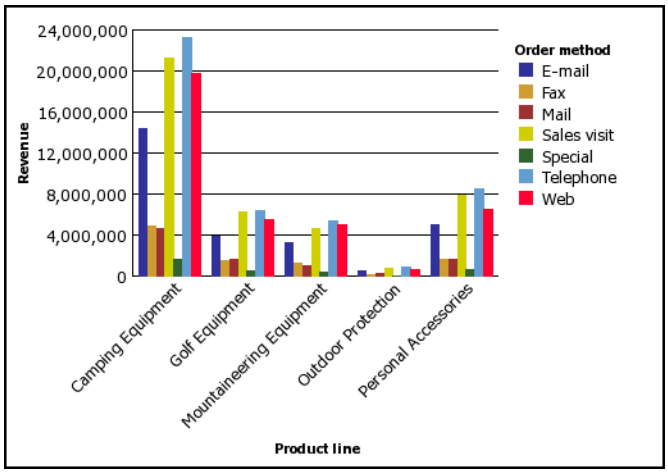
7. Run the report to see your chart.

   This makes the values easier to see. After we add more data, showing the values clutters the chart, so we will remove them from view again.
8. Return to Report Studio.
9. Click the bar icon in the **Series** drop zone.
10. In the **Properties** pane, under **Chart Labels**, change the **Values** property to **Hide**.

Let's group this chart based on **Order Method**.

Steps to Group Data in a Chart

1. Drag **Order method** (in **Orders**) to the **Series** drop zone.
2. Run the report to see your chart.

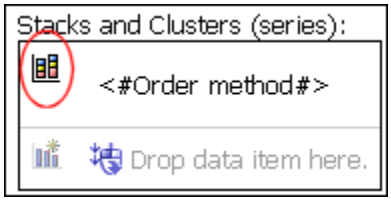


3. Return to Report Studio.

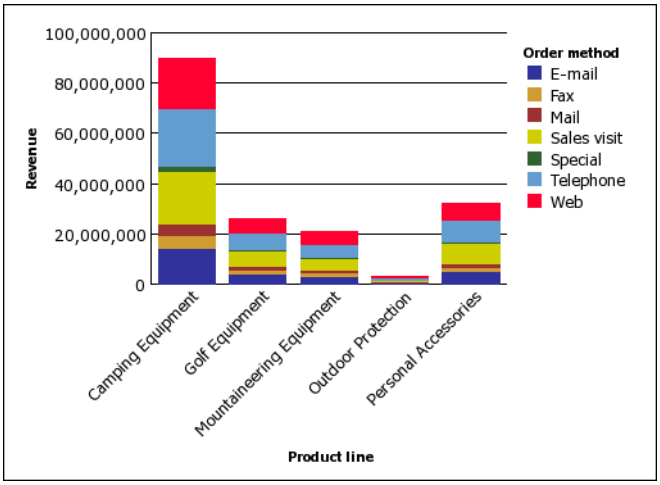
Let's see what other kind of charts we can make with this data.

Steps to Change the Chart Configuration

1. Click the **Order method** legend item icon.



   There are many properties to let you customize how the columns are shown.
2. In the **Properties** pane, set the **Grouping Type** property to **Stacked**.
3. Run the report to see your chart.



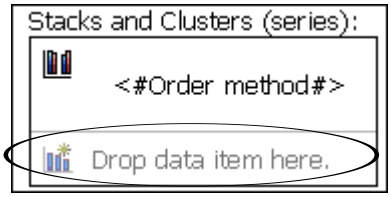
4. Return to Report Studio.

You can also use the properties in the **Properties** pane to show the data as a 100% stacked column chart, show or hide borders, and so on.

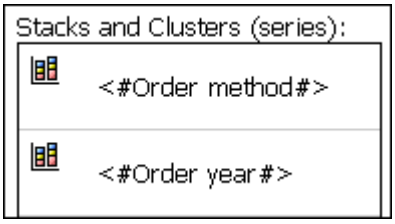
Let's add some more data to this chart.

Steps to Combine Sets of Data in Charts

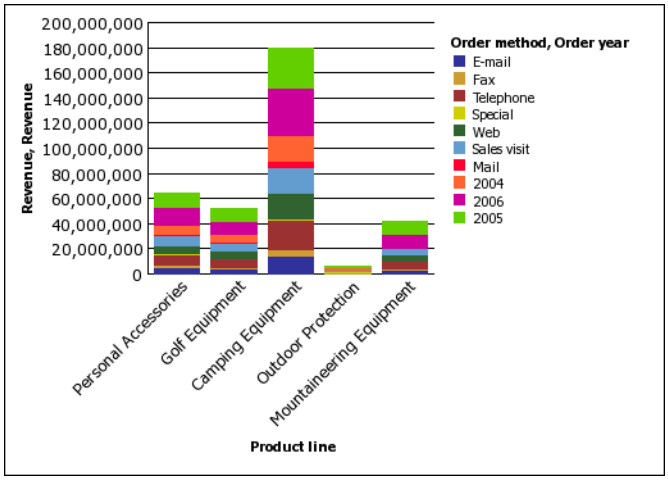
1. Add **Order year** (in **Orders**) to the **Series** drop zone.



This is how you union together the **Order method** and **Order year** items on the legend edge:

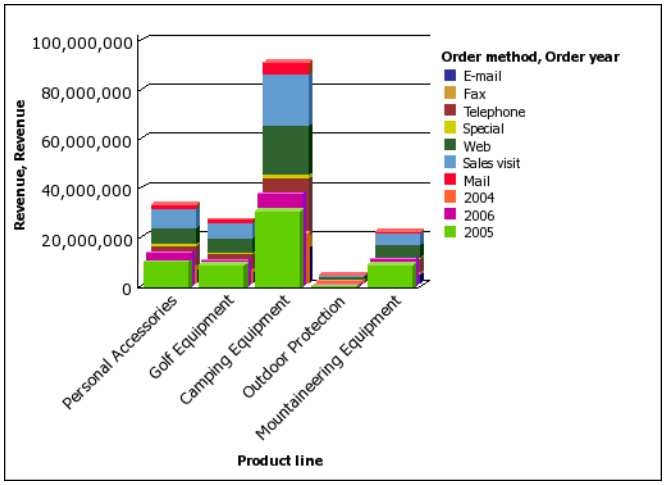


2. Run the report to see your chart.



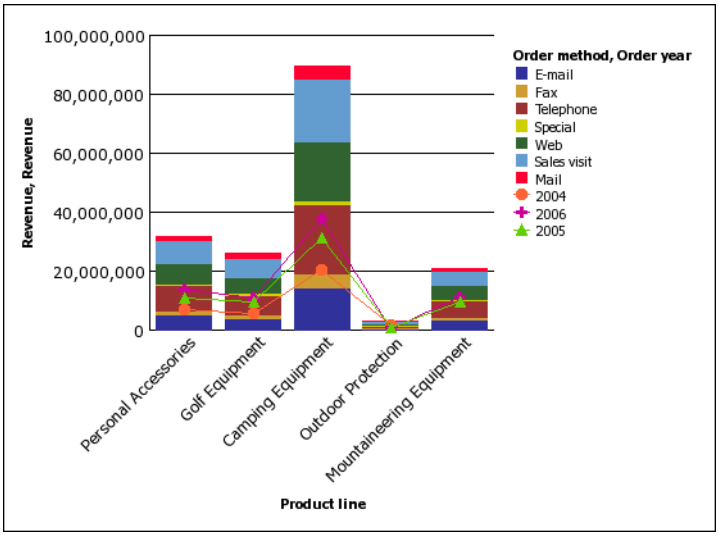
Report Studio places another set of data values on the stack.

3. Return to Report Studio.
4. Click the **Order year** legend item icon.
5. In the **Properties** pane, set the **Grouping Type** property to **Clustered**.
6. Run the report to see your chart.



You can combine stacked and unstacked charts.

7. Return to Report Studio.
8. Select **Order year**.
9. Set the **Chart Type** property to **Line**.
10. Run the report to see your chart.



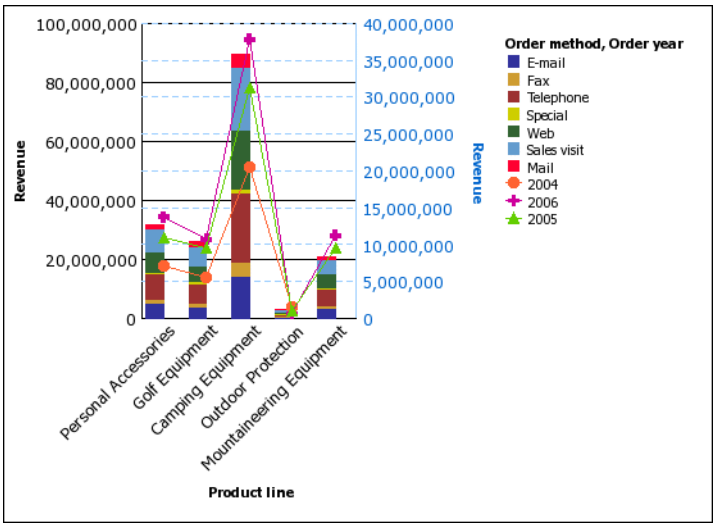
You can combine line and stacked charts. Let's improve the presentation of the **Order year** values by showing them on their own y-axis.

Steps to Change the Chart Presentation

1. Return to Report Studio.
2. Select **Order year**.
3. Set the **Axis Assignment** property to **Y Axis 2**.

   The chart image now shows two y-axes. There are two ways to view this second y-axis.
4. Select the chart background, and set the **Y2 Axis Position** property to **bipolar**.
5. Run the report to see your chart.

   The line chart appears under the stacked column chart.
6. Return to Report Studio.
7. Select the chart background, and set the **Y2 Axis Position** property to **dual**.
8. Run the report to see your chart.

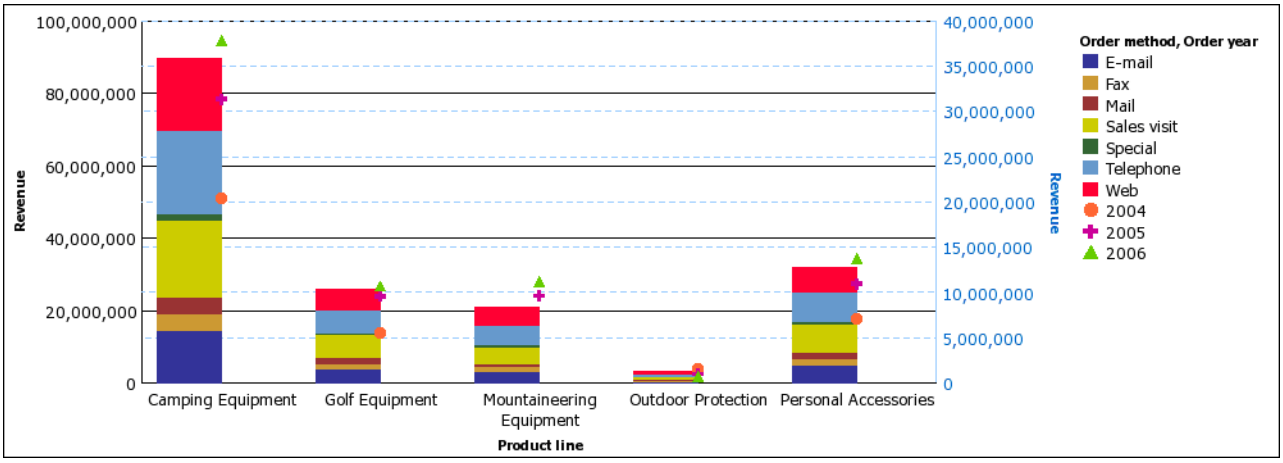


Now the chart is too narrow. We also prefer points instead of lines.

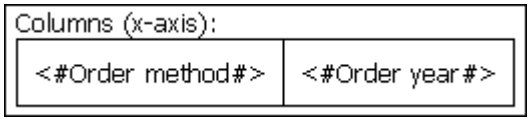
9. Return to Report Studio.
10. Select **Order year** and, in the **Properties** pane, change the **Line** property from **Yes** to **No**.

    You can also use the **Properties** pane to resize the points, restyle the lines, and so on.

11. Select the chart background, set the **Size & Overflow** property **Width** to 1000 px, and click **OK**.
12. Run the report to see your chart.

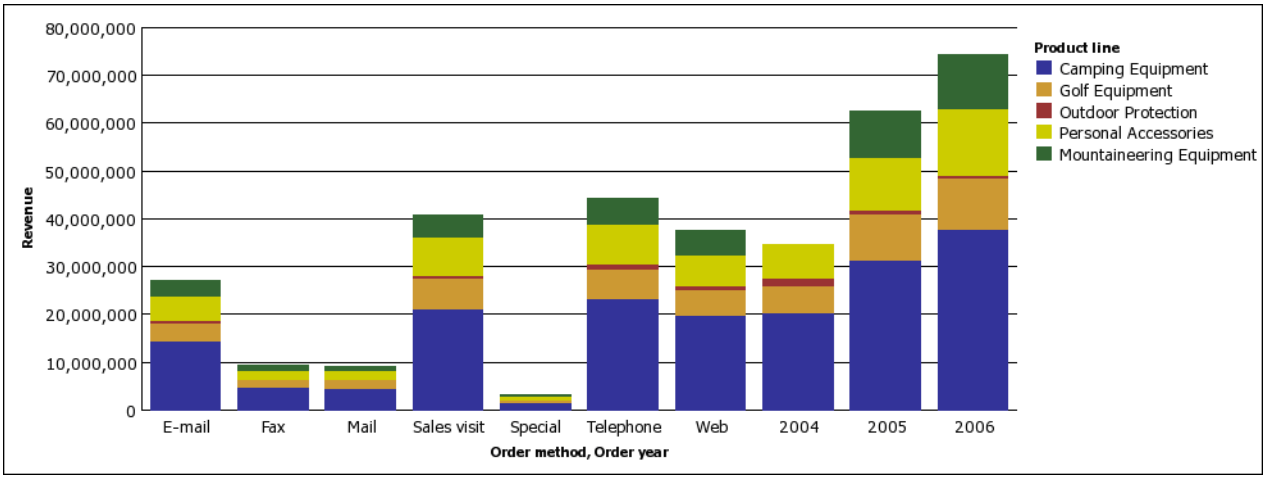


13. Return to Report Studio.
    There is another way to show the second axis.
14. Set the **Y Axis 2** property to **bi-polar**.
    You can pivot a chart like a crosstab.
15. Click the swap rows and columns button on the toolbar.
    There are now two ordinal edge entries: **Order method** and **Order year**.



You can union and nest data items on both edges equally. There is an additional drop zone on the legend edge because multiple legend entries are more common in charting than multiple ordinal edge entries.

16. Run the report to see your chart.

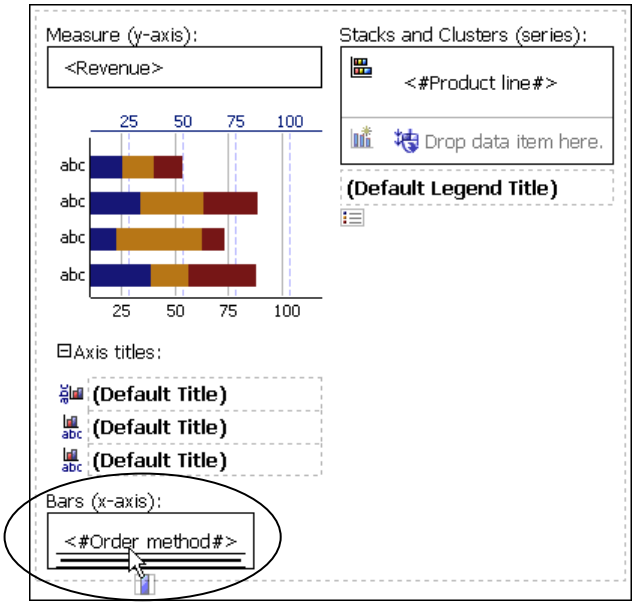


17. Return to Report Studio.
    You can view this as a horizontal bar chart.
18. Select the chart and set the **Chart Orientation** property to **Horizontal**.
19. In the **Size & Overflow** property, delete the **Width** value.
20. Run the report to see your chart.

You nest data to make charts even more flexible. For example, you can nest **Order Year** within **Order Method** to show the trend for each order method.

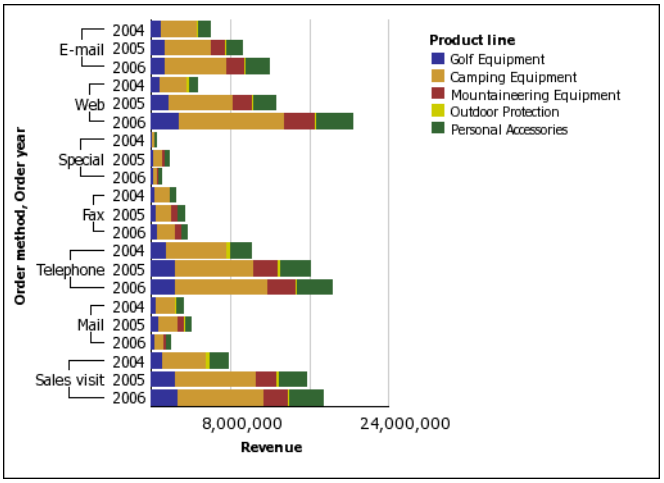
Steps to Nest Data

1. Return to Report Studio.
2. Drag **Order year** below **Order method** as shown below.



3. Click **Order year** in the **Category** (**y-axis**) drop zone, and from the **Data** menu, click **Sort Ascending**.

   There are several drop zones in this area. They allow you to nest and union data as you can in a crosstab.
4. Run the report to see your chart.



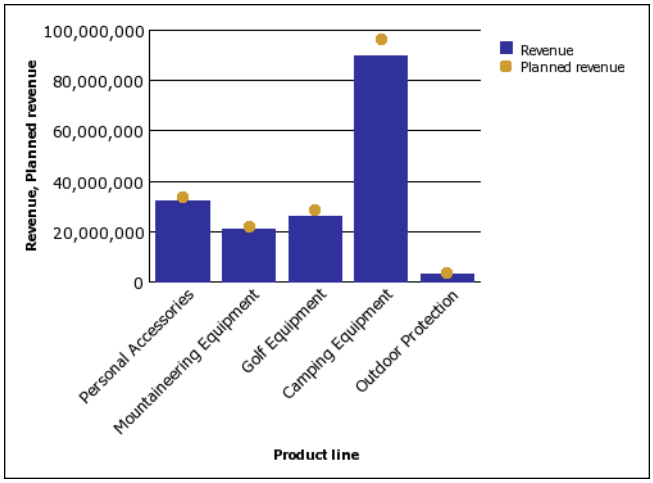
You can convert one chart type into another. We will start with a progressive column chart to show revenue contribution by product line.

Steps to Convert a Chart

1. Create a new chart.
2. In the **Chart group** pane, click **Progressive**.
3. In the **Chart type** pane, click **Progressive Column**.

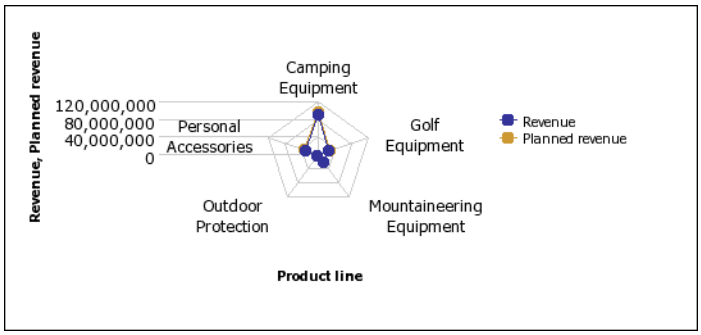
4. Click **OK**.
5. Drag **Revenue** (in **Orders**) to the **Measure** (**y-axis**) drop zone.
6. Drag **Product line** (in **Products**) to the **Category** (**x-axis**) drop zone.
7. Run the report to see your chart.
8. Return to Report Studio.
9. Right-click the chart, and from the right-click menu, click **Convert Chart**.
10. Click **OK** to open the default column chart.
11. In the **Chart Conversion Loss** dialog box, click **OK**.

    Up to now, we have used only a single measure on the chart. Now we will use two measures to compare revenue versus planned revenue for each product line.
12. Drag **Revenue** from the **Measure** (**y-axis**) drop zone to the **Series** drop zone.
13. In the **Properties** pane, ensure that the **Chart Type** property is set to **Bar**.
14. Drag **Planned Revenue** (in **Orders**) to the **Series** drop zone, under **Revenue**.
15. Select **Planned Revenue** and, in the **Properties** pane, set the **Chart Type** property to **Line**.
16. Change the **Line** property from **Yes** to **No**.
17. Run the report to see your chart.



18. Return to Report Studio.

    You can also show **Planned revenue** as a radar chart.
19. Right-click the chart, and from the right-click menu, click **Convert Chart**.
20. In the **Convert Chart** dialog box, click the **Radar, Polar** chart group and the **Radar with Markers** chart type. Click **OK**.
21. In the **Chart Conversion Loss** dialog box, click **OK**.
22. Run the report to see your chart.



It is difficult to distinguish **Revenue** from **Planned revenue** in this chart type. Another chart you can use to compare revenue and planned revenue is a metrics range chart.

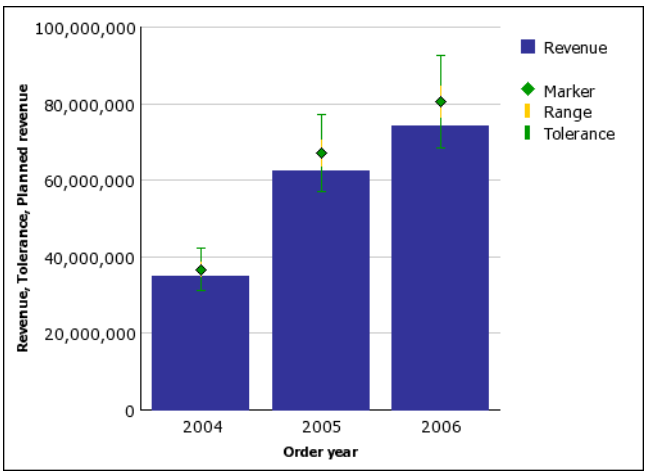
Steps to Create a Metrics Range Chart

1. In Cognos Connection, go to the GO Sales and Retailers package.
2. Click the Report Studio link.

   Report Studio starts.
3. From the **File** menu, click **New**.
4. Click **Chart** and click **OK**.
5. In the **Chart group** pane, click **Metrics Range**.
6. In the **Chart type** pane, click **Column Chart with Range Indicators**.
7. Click **OK**.
8. Drag the following items to the chart:
   - **Revenue** (in **Orders**) to the **Actual** (**y-axis**) drop zone.
   - **Planned revenue** (in **Orders**) to the **Target** (**marker**) drop zone.
   - **Order year** (in **Orders**) to the **Category** (**y-axis**) drop zone
9. From the **Insertable Objects** pane, on the toolbox tab, drag a query calculation to the **Tolerance** (**marker**) drop zone.
10. In the **Create Calculation** dialog box, type

    Tolerance
11. Click **OK**.
12. In the **Expression Definition** box, type the expression:

    [Planned revenue] * 0.1
13. Click **OK**.
14. Click run on the toolbar, and view the report.



Need More Help?

- Add Data to a Report
- Charts
- Working with Queries
- Run a Report

Try It Yourself - Create Map Reports

Use maps to represent data that can be displayed spatially. Cognos 8 includes maps that you can link to data in your data source. You can edit maps and add additional maps by using Map Manager. For more information see the Map Manager *Installation and User Guide*.

Show the Distribution of Revenue by Country

In this exercise, you use a map of the world to show revenue by country.

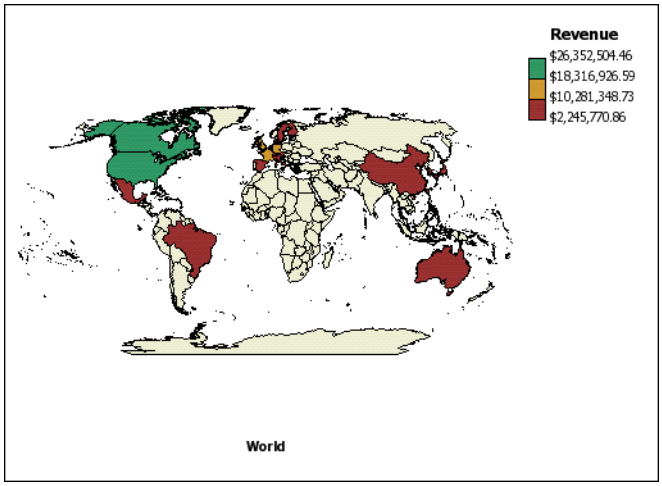
Steps to Begin a Map

1. Open Report Studio with the GO Sales and Retailers package.
2. Create a new **Map** report.
3. Select the map of the world.
4. In the **Region Layers** box, click **Countries + Territories**.
5. In the **Point Layers** box, click **None**.
6. Click **OK**.

   Tip: You can return to the **Choose Map** dialog box at any time by double-clicking the map object.

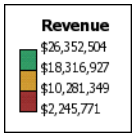
Steps to Define the Region Layer

1. In the **Insertable Objects** pane, expand **Orders**.
2. Drag **Revenue** to the **Color** drop zone.
3. In the **Insertable Objects** pane, expand **Countries**.
4. Drag **Country** to the **Location** drop zone.
5. Run the report.



Steps to Format the Numbers in the Legend

1. In the region layer, click the **Revenue** measure in the color drop zone.
2. In the **Data** section of the **Properties** pane, click **Data Format**, and then click the ellipsis points (...).
3. In the **Format Type** box, select **Currency**.
4. For the **Currency** property, select **$ (USD) - United States of America, dollar**.
5. For the **No. of Decimal Places** property, select **0** and click **OK**.
6. Run the report.



Steps to Add Another Color to the Map

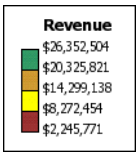
1. In the Report, click **Region Layer**.

2. In the **Color & Background** section of the **Properties** pane, click **Palette**, and then click the ellipses points (...).
3. Click the new button, and click **Color**.

   A new color is added to the list of colors.
4. With the new color selected, click **Color** in the right pane of the dialog box, and select a color.

   Four colors are now defined.
5. Change the percentage boundaries for the colors to **25**, **50**, and **75**.
6. Click **OK**.
7. Run the Report.

   The colors on the map now indicate four levels of revenue.



Show Revenue and Margins for Cities in the United States

In this exercise you use a map of the United States to show revenue and margins for outlets by using points for cities.

Steps to Begin the Map

1. Open Report Studio with the GO Sales and Retailers package.
2. Create a new **Map** report.
3. In the **Choose Map** dialog box, in the **Maps** pane, expand **Americas** and click **United States (Continental 48)**.
4. In the **Region Layers** box, click **States**.
5. In the **Point Layers** box, click **Major Cities**.
6. Click **OK**.

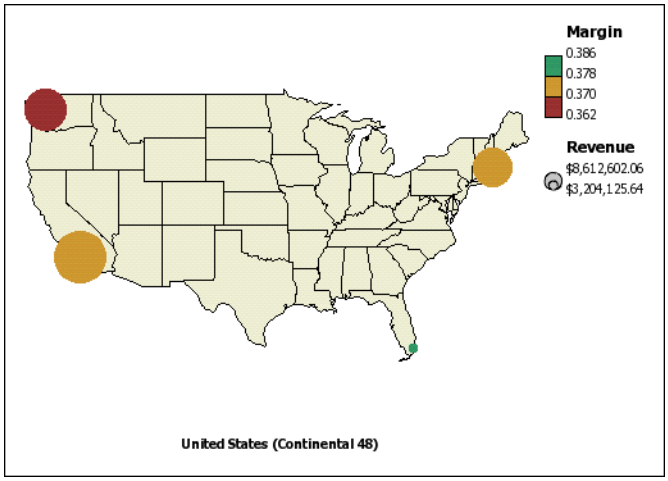
   Tip: You can return to the **Choose Map** dialog box at any time by double-clicking the map body.

Steps to Set the Point Layer

1. In the **Insertable Objects** pane, expand **Orders**.
2. Drag **Margin** to the **Color** drop zone in the **Point Layer**.
3. Drag **Revenue** to the **Size** drop zone in the **Point Layer**.
4. In the **Insertable Objects** pane, expand **Sales branch address**.
5. Drag **City** to the **Location** drop zone in the **Point Layer**.
6. Click the map object.
7. In the **Properties** pane, click **Ignore Data with no Features** and change the property to **Yes**.

   This specifies that the report can run even if there is not a match in the map file for every data value retrieved from the data source.
8. Run the report.

   The points are located at cities with sales branches. The size of the point shows the level of revenue and the color of the point shows the margin for each sales location.



Steps to Change the Size of the Points

1. In the point layer, click **Revenue** in the **Size** drop zone.
2. In the **General** section of the **Properties** pane, click **Minimum Size**, and select **2pt**.
3. Click **Maximum Size**, and select **10pt**.
4. Run the report.
   The points on the map are now smaller.

Need More Help?

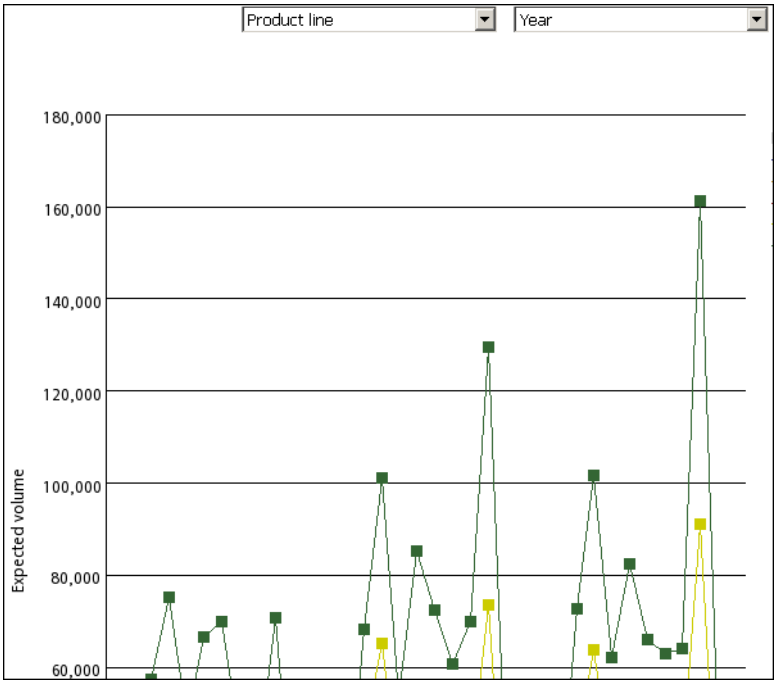
- Parts of a Map Report
- Ignore Data with no Features
- Match Data Values to Names in the Map File
- Add Titles, Subtitles, and Footers
- Add Legend Titles
- Add a Note to a Map
- Customize a Map
- Drill Through to Another Report From a Map
- Add and Edit Maps

Try It Yourself - Add Prompts

Use prompts to filter data by using the criteria entered when the report is run.

In this topic, you learn how to create cascading prompts for an existing report. Your users have the choice of filtering data in the line chart or viewing all the data.

It should take 10 to 15 minutes to complete this topic, and your prompt page will look like this.



Steps

1. Open the sample report named **Product Line by Year**.
2. Add a value prompt to the report that uses **Product line code** for the package item. Use **Product line** as the values to display. This is an optional filter.

   Tip: Use the toolbox tab in the **Insertable Objects** pane.
3. Add a value prompt to the report that uses **Year** for the package item and the values to display. This is an optional filter.
4. Ensure the **Required** property for both prompts is set to **No**.
5. Change the title text placeholder to this text:

   Product Line by Year
6. Run the report to view what it will look like for your users.

Need More Help?

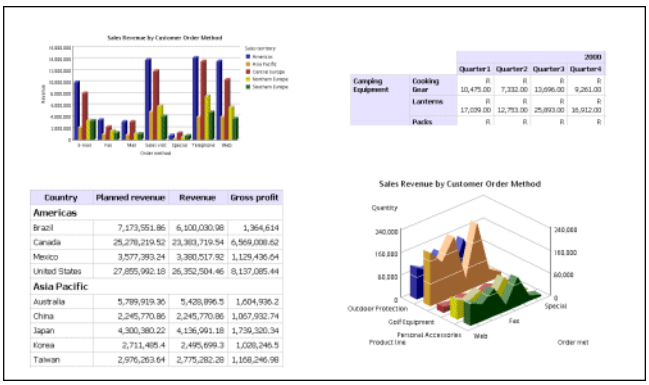
- Adding a Prompt
- Run a Report

Try It Yourself - Create a Multiple-Page Report

Add pages to a new or existing report that contain different content to create a multiple-page report.

In this topic, you learn how to create a multiple-page report that shows several sales performance reports.

It should take 15 to 20 minutes to complete this topic, and your report pages will look like this.



Steps

1. Open the sample report named **Global Sales (1)**. This sample contains several reports.
2. Add a page.
3. Add a 3-D column chart to the new page as a new query.
4. Add these data items to the chart:
   - **Revenue** (in **Orders**) as the axis measure
   - **Order method** (in **Orders**) as the category
   - **Sales territory** (in **Countries**) as the series

   Tip: Use the source tab in the **Insertable Objects** pane.
5. Add a list report to the page as a new query.
6. Add these data items to the list report:
   - **Sales territory** (in **Countries**)
   - **Country** (in **Countries**)
   - **Planned revenue** (in **Orders**)
   - **Revenue** (in **Orders**)
   - **Gross profit** (in **Orders**)
7. Group the **Sales territory** column to make the report easier to read. Then group the **Country** column.
8. Make the **Sales territory** column appear as a list header in the report.
9. Bold **Sales territory** in the body of the report.
10. Remove the **Sales territory** column title without removing **Sales territory** from the body of the report.
11. Change the blue text placeholder to this text:

    Performance by Sales Territory
12. Run the report to view what it will look like for your users.

Need More Help?

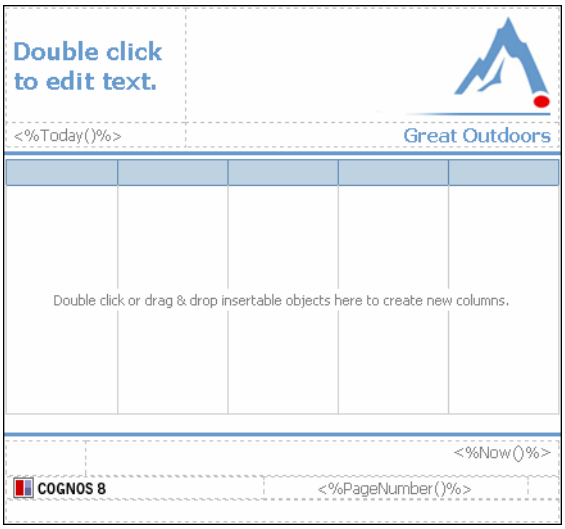
- Add a Page
- Add Data to a Report
- Charts
- Crosstab Reports
- List Reports
- Run a Report

Try It Yourself - Create a Template

Use templates to create a standard look for corporate reports. Any report can be used as a template. You simply create and format a report and then use it as your starting point for all other reports, leaving the original report unchanged.

In this topic, you learn how to create a template containing a header and footer.

It should take 15 to 20 minutes to complete this topic, and your template will look like this.



Steps

1. Create a new blank report, but do not add data to it.
2. Add a page header.
3. Add a border to the bottom of the header. Make the border 2-1/4 pts and apply the Web-safe color #6699CC.
4. Add a table that has two columns and two rows.

   Tip: Use the toolbox tab in the **Insertable Objects** pane.
5. Add a text item to the top-left cell of the table, and format it as Tahoma, 14 pt, bold, and apply the Web-safe color #6699CC.
6. Add the date to the bottom-left cell of the table, and format it as Tahoma, 9 pt, gray.
7. Add this sample image to the top-right cell and align the image to the right:

   ..\samples\images\logo.jpg
8. Add this text item to the bottom-right cell:

   Great Outdoors
9. Format the **Great Outdoors** text as Tahoma, 11 pt, bold. Apply the Web-safe color #6699CC. Align the text to the right.
10. Add a page footer.
11. Add the same border to the top of the footer as you added to the header.
12. Add a table that has three columns and three rows.
13. Merge the second and third cells in the first row.
14. Merge all cells in the third row.
15. Add the time to the top-right cell of the table, and align it to the right.
16. Add the page number to the middle-center cell of the table, and align it to the center.
17. Format the time and page number as Tahoma, 9 pt, gray.
18. Add this sample image to the middle-left cell:

    ..\samples\images\banner_cogsquare.jpg
19. Add a list report to the report body, and center the list report.

20. Convert the report to a template.
21. Save the template, and use it as the starting point for a report.

Need More Help?

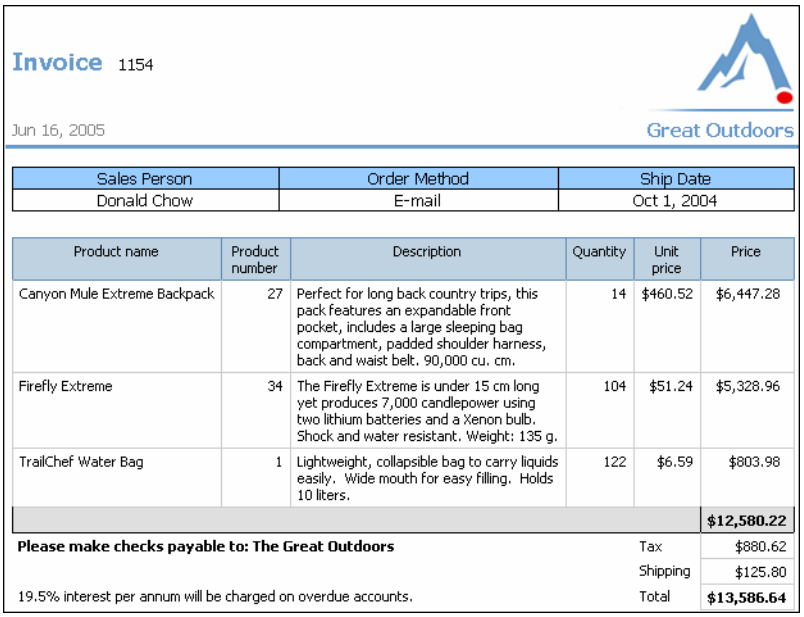
- Add Borders
- Add Color
- Add a Header or Footer
- Add Text
- Align an Object
- Choose a Report Template
- Create Your Own Report Templates
- Format Data
- Insert an Image
- Use Tables to Control Where Objects Appear

Try It Yourself - Create an Invoice

Combine a list report, tables, text, and an image to create an invoice.

In this topic, you learn how to create an invoice that shows each customer's order.

It should take 30 to 40 minutes to complete this topic, and your invoice will look like this.



Invoice 1154

Jun 16, 2005

Great Outdoors

Sales Person	Order Method	Ship Date
Donald Chow	E-mail	Oct 1, 2004

Product name	Product number	Description	Quantity	Unit price	Price
Canyon Mule Extreme Backpack	27	Perfect for long back country trips, this pack features an expandable front pocket, includes a large sleeping bag compartment, padded shoulder harness, back and waist belt. 90,000 cu. cm.	14	$460.52	$6,447.28
Firefly Extreme	34	The Firefly Extreme is under 15 cm long yet produces 7,000 candlepower using two lithium batteries and a Xenon bulb. Shock and water resistant. Weight: 135 g.	104	$51.24	$5,328.96
TrailChef Water Bag	1	Lightweight, collapsible bag to carry liquids easily. Wide mouth for easy filling. Holds 10 liters.	122	$6.59	$803.98
					$12,580.22

Please make checks payable to: The Great Outdoors	Tax	$880.62
	Shipping	$125.80
19.5% interest per annum will be charged on overdue accounts.	Total	**$13,586.64**

Get Started

1. Create a new blank report that uses the sample package named **GO Sales and Retailers**.
2. Change the font for the entire page to Arial.

Define the Page Header

1. Add a page header to the report.
2. Add a table with two vertical bands (columns) and two horizontal bands (rows).
3. Add this text to the top left cell and format it (we used the Tahoma font, 14 point, and a Web-safe blue):

   Invoice
4. Add the date to the bottom left cell.

5. Add this sample image to the top right cell:

   ..\samples\images\logo.jpg
6. Add this text to the bottom right cell and format it (we used the Tahoma font, 11 point, and a Web-safe blue):

   Great Outdoors
7. Add a border to the header.
8. Set the bottom padding to 10 pixels.

Associate the Page with a New Query

1. In the **Query Explorer** view, create a new query and add **Order number** (in **Orders**) as a dimension.
2. In the **Report Pages** view, create a new page set and add the new query to the Data properties of the page set. This means that you can add a data item anywhere in the report.
3. Add the **Order number** to the header, next to Invoice.
4. Group **Order number**.
5. Change the data format for the **Order number** to not have a comma.

Set Up the Page Body

1. Add a block to the page body.
2. Change the top padding for the block to 10 pixels.
3. Add a table with two rows and three columns to the block. The table must be the maximum width and have a border.
4. Add a block after the table.
5. Change the top padding for the bottom block to 20 pixels.
6. Change the background color of the first row of the table to purple.
7. Center the text in the table.
8. Add a list below the bottom block that you added.

Add Data to the Page Body

1. Add text items to the first row of the table and use this text for each one:
   - **Sales Name**
   - **Order Method**
   - **Ship Date**
2. Add these data items to the second row of the table:
   - **Staff name** (in **Sales reps**)
   - **Order method** (in **Orders**)
   - **Order date** (in **Orders**)

   Tip: Use the source tab in the **Insertable Objects** pane.
3. Change the **Data Format** property for the **Ship Date** column to be a date only, not date and time.
4. Add these data items to the list report:
   - **Product name** (in **Products**)
   - **Product number** (in **Products**)
   - **Description** (in **Products**)
   - **Quantity** (in **Orders**)
   - **Unit price** (in **Orders**)
5. Add this query calculation as a column and name it **Price**:

   [gosales_goretailers].[Order details].[Quantity]*[gosales_goretailers].[Order details].[Unit price]
6. Change the **Aggregate Function** property for the **Quantity** column to **None**.
7. Change the **Aggregate Function** property for the **Unit price** column to **None**.

Add Information at the Bottom of Each Invoice

1. Add an overall footer to the list.
2. Add a subtotal to the **Price** column.
3. Insert a row and add a query calculation named **Tax** to the new row:
   ([gosales_goretailers].[Orders].[Quantity]*[gosales_goretailers].[Orders].[Unit price])*0.07
4. Add this text item next to **Tax** and bold the text:
   Please make checks payable to: The Great Outdoors
5. Merge the cells next to the **Tax** label so that the text item has more room.
6. Insert a row and add a query calculation named **Shipping** to the new row:
   ([gosales_goretailers].[Orders].[Quantity]*[gosales_goretailers].[Orders].[Unit price])*0.01
7. Remove the borders from the cells to the left of the **Shipping** label.
8. Insert a row and add a query calculation named **Total** to the new row:
   ([gosales_goretailers].[Orders].[Quantity]*[gosales_goretailers].[Orders].[Unit price])*1.08
9. Add this text item next to the **Total** and then bold and center the text:
   19.5% interest per annum will be charged on overdue accounts.
10. Merge the cells next to the **Total** label so that the text item has more room.
11. Run the invoice to view what it will look like for your users.

Need More Help?

- Add Data to a Report
- Add a Header or Footer
- Add a Summary
- Add Text
- Choose a Report Template
- Create a Calculation
- Group Data
- Use Tables to Control Where Objects Appear
- Working with Queries
- Run a Report

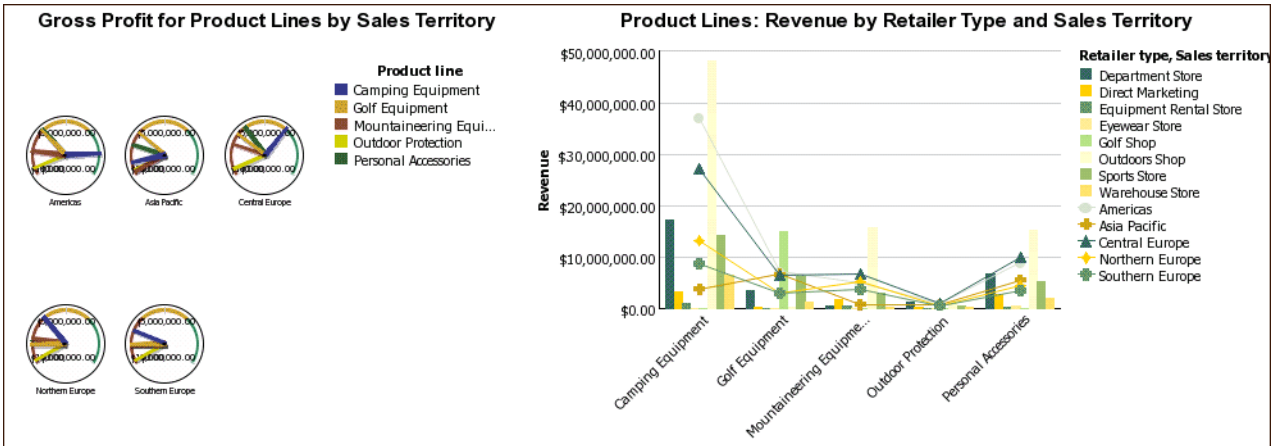
Try It Yourself - Create a Dashboard Report

Create a dashboard report to show a high-level graphical view of company performance indicators. By using dashboards, users can

- drill up and drill down to see higher and lower levels of detail
- navigate to other targets, including Report Studio reports, Query Studio reports, and Analysis Studio analyses
- include multiple charts derived from different data sources in a single report

In this topic, you will learn how to create a dashboard 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 will look like this.



Gross margin	2004	2005	2006	Americas	Asia Pacific	Central Europe	Northern Europe	Southern Europe
Camping Equipment	30.63%	29.52%	28.18%	29.03%	30.24%	28.90%	29.67%	29.78%
Golf Equipment	47.89%	46.18%	45.58%	46.13%	47.09%	45.86%	47.15%	45.88%
Mountaineering Equipment	0.00%	38.23%	36.59%	37.75%	39.78%	36.45%	37.75%	37.14%
Outdoor Protection	56.18%	56.06%	55.50%	56.32%	57.39%	52.91%	57.43%	55.91%
Personal Accessories	36.72%	35.53%	34.13%	35.59%	36.27%	34.55%	35.04%	34.86%

Steps to Create the Report and Add Data

1. Create a blank report that uses the sample package named **Great Outdoors Company**.
2. Add a gauge chart, column chart, and crosstab.
3. Drag the following items to the gauge chart:
   - **Gross profit** (in **Measures**) to the **Measure** drop zone
   - **Product line** level (in **Products**) to the **Needles (series)** drop zone
   - **Sales territory** level (in **Sales Territory**) to the **Gauges** drop zone
4. Drag the following items to the column chart:
   - **Revenue** (in **Measures**) to the **Measure (y-axis)** drop zone
   - **Product line** level (in **Products**) to the **Category (x-axis)** drop zone
   - **Retailer type** level (in **Retailer**) and **Sales territory** (in **Sales Territory**) to the **Series** drop zone
5. Drag the following items to the crosstab:
   - **Gross margin** (in **Measures**) to the **Measures** drop zone
   - **Product line** level (in **Products**) to the **Rows** drop zone
   - **Year** level (in **Years**) and **Sales territory** level (in **Sales Territory**) to the **Columns** drop zone

Steps to Format the Gauge Chart

1. Click the gauge chart.
2. Set the **Size & Overflow** property width to **400 px**.
3. Set the **Title** property to **Show**.
4. Double-click the chart title and type the following text:
   Gross Profit for Product Lines by Sales Territory
5. Set the **Font** property for the chart title to **Arial, 12pt, Bold**.
6. Expand **Axis titles**, click the horizontal axis title, and set the **Visible** property to **No**.
7. Click the numerical axis and, in the **Maximum Value** property, type **4000000**.

Steps to Format the Column Chart

1. Click the **Sales territory** series and set the **Chart Type** property to **Line**.
   Note: The column chart becomes a combination chart because you render the sales territory data as lines.
2. Click the chart, click the **Palette** property, and select **Contemporary** from the **Palette** list.

3. Set the **Size & Overflow** property width to **600 px**.
4. Set the **Title** property to **Show**.
5. Double-click the chart title and type the following text:
   Product Lines: Revenue by Retailer Type and Sales Territory
6. Set the **Font** property for the chart title to **Arial, 12pt, Bold**.
7. Expand **Axis titles**, click the horizontal axis title, and set the **Visible** property to **No**.

Steps to Specify the Drill Behavior

1. From the **Data** menu, click **Drill Behavior**.
2. Select the **Allow drill-up and drill-down** check box.
3. Run the dashboard to view what it will look like for your users.
   Users can drill up or drill down on values in the report to view related information.

Need More Help?

- Choose a Report Template
- Add Data to a Report
- Working with Dimensional Data
- Specify the Properties of a Gauge Chart
- Specify the Height and Width
- Specify the Font
- Example - Customize the Axis Titles
- Create a Drill-up/Drill-down Report

Try It Yourself - 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 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.

Revenue	2006/Mar	YTD Revenue for 2006/Mar	Percent of YTD Revenue
Camping Equipment	$4,075,510.86	$8,162,650.72	49.93%
Golf Equipment	$1,077,338.74	$1,796,219.58	59.98%
Mountaineering Equipment	$1,135,978.28	$1,848,903.76	61.44%
Outdoor Protection	$70,134.60	$121,930.90	57.52%
Personal Accessories	$1,146,324.42	$2,278,711.36	50.31%

Steps to Create the Report

1. Create a crosstab report that uses the sample package named **Great Outdoors Company**.
2. Add the following data items to the report:
   - **Product line** level (in **Products**) to the **Rows** drop zone
   - **Revenue** (in **Measures**) to the **Measures** drop zone

   Tip: Use the source tab in the **Insertable Objects** pane.
3. Create this query calculation named **Selected Month** in the **Columns** drop zone:
   great_outdoors_company].[Years].[Years].[Month]->?Month?
4. Create this query calculation named **Year to Date Set** next to the **Selected Month** column:
   periodsToDate([great_outdoors_company].[Years].[Years].[Year],[Selected Month])

Tip: For information about the **periodsToDate** function, click the function in the functions tab. It is located in the **M-Q** folder in the **Common Functions** folder.

5. Click the **Year to Date Set** column and click the cut button.
6. Create this query calculation named **YTD Revenue** next to the **Selected Month** column:

   total(currentMeasure within set[Year to Date Set])
7. Create this query calculation named **Percent of YTD Revenue** next to the **YTD Revenue** column:

   [Selected Month]/[YTD Revenue]

Steps to Format the Report

1. Click the lock/unlock button to unlock the report.
2. 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 for
3. Set the font style of the text to italic.
4. In the **YTD Revenue** column, click **<#YTD Revenue#>** and change the **Source Type** property to **Report Expression**.
5. 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.
6. Click the lock/unlock button to lock the report.
7. Right-click the **Percent of YTD Revenue** column title and click **Fact Cells for "Percent of YTD Revenue"**.
8. In the **Properties** pane, double-click the **Data Format** property.
9. Under **Format type**, click **Percent**.
10. Set the **No. of Decimals** property to **2**.
11. Run the report to view what it will look like for your users.

    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 generated for the selected month.

Need More Help?

- Choose a Report Template
- Add Data to a Report
- Working with Dimensional Data
- Using the Expression Editor
- Format Crosstabs
- Specify the Font
- Format Data

Try It Yourself - Create a Report with Drill-through Access to Itself

Create a report that drills 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.

Order Details for All Retailers

Retailer name	Revenue
Act'N'Up Fitness	$1,265,504.30
ActiForme	$242,494.60
Advanced Climbing Ltd	$33,521.60
Alles für Draußen	$897,925.30
Allo Allo	$728,601.00
Altitudes extrêmes	$1,337,309.80
American Home	$1,198,096.60
Anapurna	$976,845.70
Arjan Aitta	$3,762,524.74
Arnold's Clubs	$349,390.32
Artículos de Campismo El Aguila, S.A. de C.V.	$2,000,396.56
Ausrüstungshaus Globetrotter	$1,255,481.70
Beach Beds Pty Ltd.	$1,519,213.22
Beck's Sports Store	$1,422,316.62
Bellini	$855,837.34
Beter Buitenleven	$4,928,985.00
Blue Mountains Golfing Company	$300,157.04
Brambilla	$499,167.70
Browns Opticals	$40,264.04
Buena Vista Eyeware	$52,481.42

Select a retailer in the left list to view order details below

Order number	Order date	Product name	Quantity	Revenue

Steps to Create the Report and Add Data

1. Create a new blank report that uses the sample package named **GO Sales and Retailers**.
2. Add a table with two columns and one row.
3. Add a list object to each column in the table.
4. Add these data items to the first list:
   - **Retailer name** (in **Retailers**)
   - **Revenue** (in **Orders**)

   Tip: Use the source tab in the **Insertable Objects** pane.
5. Set the sort order for the **Retailer name** column to **Sort Ascending**.
6. Add **Retailer name** in **Retailers** to the second list.
7. Add these data items from the **Orders** folder to the second list:
   - **Order number**
   - **Order date**
   - **Product name**
   - **Quantity**
   - **Revenue**

   Tip: To simultaneously add all the data items to the list, press Ctrl+click and select the items before dragging them to the list.
8. In the second list, click the **Retailer name** column and click the cut button.
9. In the second list, create the following parameterized filter:

   [Retailer name]=?Selected retailer?
10. Save the report.

Steps to Set up Drill-through Access

1. In the left list, right-click a **Retailer name** column (and not the column title) and click **Drill throughs**.
2. Create a new drill-through definition.
3. Specify the report as the target report.
4. Under **Parameters**, click the edit button.
5. Set the **Selected retailer** parameter to pass data item values using the **Retailer name** data item.

Steps to Add a Prompt

1. 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.
2. In the **Prompt Wizard**, specify that the prompt is to use the existing parameter named **Selected retailer**, and click **Finish**.
3. Click the value prompt.
4. In the **Properties** pane, double-click the **Default Selections** property and add a simple selection named **NoRetailer**.
5. Set the following properties:
   - **Required** to **No**
   - **Hide Adornments** to **Yes**
   - **Visible** to **No**

Steps to Add a Dynamic Report Title

1. Double-click the report title and type the following text, adding a blank space at the end:

   Order Details for
2. In the toolbox tab, add the following layout calculation to the right of the report title:

   if(ParamDisplayValue('Selected retailer')='NoRetailer') then 'All Retailers' else (ParamDisplayValue('Selected retailer'))

   When the report is run, the report title changes to reflect the retailer selected by the user. If no retailer is selected, 'All Retailers' appears.
3. Click the layout calculation and, in the **Properties** pane, set the class to **Report title text**.

Steps to Add Conditional Formatting

1. 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.
2. Create this boolean variable named **HighLevel**:

   ParamDisplayValue('Selected retailer')<>'NoRetailer'
3. Pause the pointer over the page explorer button and click **Page1**.
4. In the toolbox tab, add a block above the second list.
5. Insert a text item in the block with the following text:

   Select a retailer in the left list to view order details below
6. Set the font style for the text to bold.
7. Click the text item and, in the **Properties** pane, set the **Style Variable** property to the **HighLevel** variable.
8. Pause the pointer over the condition explorer and click the **Yes** value for the **HighLevel** variable.
9. Set the **Visible** property for the text item to **No**.
10. Triple-click the explorer bar to turn off conditional formatting.
11. Run the report to view what it will look like for your users.

    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.

Need More Help?

- Use Tables to Control Where Objects Appear
- Add 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
- Create a Calculation

- Using the Expression Editor
- Add a Variable
- Hide and Show Objects

Appendix A: Troubleshooting

This chapter describes some common problems you may encounter. For more troubleshooting problems, see the *Troubleshooting Guide*.

A Web Server Error When Trying to Browse Images

When you want to insert an image into a report and you try to browse to the location (URL) where the image is located, an error message like the one below appears:

Web server error:

501 Not Implemented

To browse images on a Web server, you must enable Web-based Distributed Authoring and Versioning (WebDAV) on your Web server.

Note: iPlanet Web server does not support WebDAV.

Steps to Enable WebDav

1. If you are using an Apache Web server, configure WebDAV for your server.

   For more information, see the Apache documentation.
2. If you are using Microsoft Internet Information Services (IIS), enable the Read and Directory Browsing properties for the URL you want to access:
   - Open the **Control Panel** and double-click **Internet Services Manager**.
   - Right-click the directory that points to the folder containing the image you want to insert and click **Properties**. If no directory exists for the folder, create one and point it to the folder.
   - In the **Virtual Directory** or **Directory** tab, select the **Read** and **Directory Browsing** check boxes and click **OK**.
3. To confirm that WebDAV is configured correctly, use Internet Explorer as a WebDAV client:
   - In Internet Explorer, from the **File** menu, click **Open**.
   - In the **Open** box, type the URL containing the images you want to browse.
   - Select the **Open as a Web Folder** check box.
   - Click **OK**.

   If the folder opens successfully, WebDAV is configured correctly.

Do You Want to Allow this Webpage to Have Access to Your Clipboard Message Appears in Report Studio

When using Report Studio in Internet Explorer 7, you may receive the following message:

"Do you want to allow this webpage to have access to your Clipboard?"

You can answer Yes to this message. To prevent this message from appearing again, enable the Allow Programmatic clipboard access security setting in Internet Explorer 7.

Values Not Recognized in Multilingual Query Items

A report may be created in one language, and then run in a different language. When you filter for specific values on multilingual query items, the filter does not recognize the values because they exist only in the original language.

When creating filters in multilingual reports, try to use non-multilingual query items, such as numerical identifiers, for the search criteria, because they are applicable in all languages. For example, use the product code rather than a multilingual product name.

Changes in the Model Are Not Reflected in the Report If Model Versioning Is Used

If you make changes to the model in Framework Manager and republish the package, you may not see the changes when you run the report. This is because the version of the model has changed.

When you republish a model, you are adding a new version of the model to the package. When you run a report, the report uses the model that it was initially created with or last edited with.

To correct the problem, do one of the following:

- Edit the report and then save it.

  The report is set to the latest version of the model.
- Before you republish the model in Framework Manager, in the **Publish** wizard, select the option **Delete all previous model versions**.

  All reports linked to the model are moved to the latest version of the model.
- Before you republish the model in Framework Manager, in the **Publish** wizard, set the model version limit to 1.

  Model versioning is disabled.

Problems When Printing a PDF Manual

You print a document in PDF format, but the print job is incomplete. For example, the job stops printing when it reaches a particular graphic.This is an Adobe Acrobat Reader issue that can occur when printing some PDFs using some versions of Acrobat Reader and some printer models. The same PDF may print correctly under one or both of the following conditions:

- using a different version of Acrobat Reader
- using a different printer

If you print from an Acrobat 4.0 or later product, you can try the following solution.

Steps to Force a Page to Print as an Image

1. In Acrobat Reader, from the **File** menu, click **Print**.
2. In the **Printer** section, select the **Print as image** check box.

   Because files print more slowly using this option, make sure you specify only the nonprinting page.
3. In the **Print Range** section, click **Pages from** and **Pages to**, type only the page number of the nonprinting page, and then click **OK**.

   You can print the rest of the PDF by resending the job starting on the next page.
4. Clear the **Print as image** check box.
5. In the **Print Range** section, click **Pages from** and **Pages to**, type the page range for the remaining pages, and then click **OK**.

   Although you can use the **Print as image** option to print the file, this setting does not resolve the original printing problem. For more information, see the Adobe Web site.

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 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.

Browser Does Not Display Report

A report with an exceptionally large number of columns may exceed the browsers capability to render HTML. The report does not display and no error message appears.

To correct the problem, try a different report structure or layout. For example, swap rows and columns and run the report again. Another option is to run the report in PDF format with horizontal pagination.

Unexpected Alignment of Items in a Report

If the same data item is used more than once in Report Studio, the alignment may not be the same for all instances when you run the report. For example, the data is right-aligned in one column, and left-aligned in another column. The difference is due to the default settings of the objects associated with the data item.

To correct the problem, ask the data modeler to change the formatting of the query item in the model. For more information, see the Framework Manager User Guide.

The java.lang.OutOfMemory Error Message Appears in Cognos Connection or Report Studio

When working in Cognos Connection or Report Studio, this java error message may appear if not enough memory is allocated to the dispatcher.

Steps to Increase the Memory Allocated to the Dispatcher

1. Start Cognos Configuration.
2. Stop the Cognos 8 service.
3. In the **Explorer** window, under **Environment**, **Cognos 8 service**, click **Cognos 8**.
4. Click the **Value** box next to the **Dispatcher maximum memory in MB** property and type a new value.

   We recommend you change the value to 768.
5. From the **File** menu, click **Save**.
6. Restart the Cognos 8 service.

Report Studio Does Not Start

You may not be able to start Report Studio if you are using pop-up blocking software on your computer.

When you start Report Studio, it opens in a new browser window. In addition, a new browser window opens when you run a report and when an error is detected.

To correct the problem, disable any pop-up blocking software when working in Report Studio.

A Report is Not Rendered Properly

In Report Studio, if a report uses empty blocks, table rows, or table columns to space the report components, the report may not render properly.

The recommended way to add spaces between report components is to define the padding or margin properties for the objects. If you use empty blocks, table rows, or table columns, define a height or width for them.

Unable to Delete a Prompt

You delete a prompt from the design page of a report. When you run the report, the prompt still appears.

The prompt continues to appear because it is saved to the query filter, and not the design page. You must delete the prompt from the query filter.

Steps

1. From the **Data** menu, click **Filters**.
2. Select the prompt you wish to delete, then click **Delete**.
3. Click **OK**.
4. Run the report.

Appendix B: Samples

Cognos 8 includes sample reports. When installed, you can find them in the **Public Folders** tab in Cognos Connection.

Sample Reports in the GO Sales Package

The following reports are found in the GO Sales and Retailers package.

Conditional Display

The report uses a prompt page to query a chart and a list report.

Custom Legend

The report shows that the legend can be customized in a similar way to the titles.

Orders Report

This report shows the default style applied to List Column Title and the List Column Body in a single step.

Product Report

The report displays combination charts with drill through options.

Retailer Report (Multiple Prompt Values)

List report that accepts multiple prompt values.

Returns by Order Method - Prompted Chart

This report uses a bar chart and a list to show the reasons why products are returned. It uses a prompt to filter on the return description. It is illustrated in the *Getting Started* tutorial.

Revenue by Sales Territory

The report shows a chart and a crosstab display with conditional formatting that drills through to the Orders Report report.

Top 5 Sales Staff

List report that embeds a bar chart that shows the top five sales representatives by sales targets and revenue.

Top Revenue (Conditional)

List report that conditionally hides the Product Line group footers for the revenue lower than the specified value.

Sample Reports in the GO Sales and Retailers Package

The following reports are found in the GO Sales and Retailers package.

Actual Sales Against Target Sales

Report showing a simple list with conditional formatting that drills through to the Sales Representative Contact List report.

Actual Sales Against Target Sales - Burst

Report that is set up for bursting to email addresses. The email addresses point to sales representatives in the Sales reps table in the GOSales database.

Banded Report

Banded report that shows Product name, Quantity, and Revenue with sales opportunities for each Product line category.

Basket Analysis with Total Contribution (Top 10 Rank)

Pairing analysis report that uses prompts to compare two products. Orders are totaled and ranked, and the top ten orders are shown.

Business Details Drill Through

Report that shows product details in charts. This report is a drill-through target report for the GO Business View report. You can also run it as a product-prompted report.

Consumer Trends

Complex report that shows a list chart, bar chart and product images to illustrate revenue by product type.

Cost of Goods

Crosstab report illustrating conditional formatting.

Custom Grouping

Report showing the ability to group products with conditional statements. For example, show all products that start with the letter S.

Customer Invoice

Report illustrating The Great Outdoors Company invoice statement.

Global Sales

Book style report that uses multiple pages and charts to show global sales results.

Global Sales (1)

A cut down version of the Global Sales report. It is discussed in the section Try It Yourself - Create a Multiple-Page Report (p. 192).

GO Business View

Two-page business report showing a combination of metrics across the Great Outdoors company. Multiple charting types are used.

GO Media

Report that uses a media file in combination with multiple charting types.

Mailing Labels

Template style report illustrating a multilingual mailing list for retailers of the Great Outdoors company.

Margin Cost and Volume Report

Report showing volume and production cost using various chart types. The report drills through on Product line.

Multi-Grain Fact

Report that reads two fact tables, each at different levels of granularity, and combines them. Bookmarks are used to permit easy navigation within the report.

Multiple Charts

Report showing similar information on one page using multiple chart types. Visual representation of metrics.

Order Analysis

Pairing analysis report that prompts the user for two products, and shows the orders that they both appear in.

Percent Contribution by Country

Report showing the percentage of products sold by country as well as a percentage comparison across other countries. Bookmarks are used for easy navigation throughout the report.

Product Comparison Charts

Report using multiple charts to reveal trends and relationships that are not evident in tabular reports. For more information, see Try It Yourself - Create Charts (p. 179).

Product Line by Year

Nested line chart. For more information, see Try It Yourself - Add Prompts (p. 191).

Product Line by Year-Prompt

Report illustrating the use of adding a prompt to an existing report. For more information, see Try It Yourself - Add Prompts (p. 191).

Product Revenue

Group list report that uses a list header. For more information, see Try It Yourself - Create a List Report (p. 177).

Product Revenue - Lifetime/Q2

Pairing analysis report showing lifetime revenue for each product with sales from Q2 in 2002.

Product Summary

Report that uses a prompt page to query a list.

Products Ranked by Revenue

Pairing analysis report that prompts the user to select a product, order year, and revenue. The results are ranked by revenue.

Quantity by Retailer

Grouped list report that drills through to the Retailer Contact report. Part of the Report Studio Tour.

Regional Orders

Report showing order quantity by country, using bookmarks for navigation. The report is the drill-through target report for the Margin Cost and Volume report.

Report with Totals

List report that uses prompts to show product sales by date and product value lifetime contribution.

Retailer Contact

List report that uses a prompt page. It is also used as a drill-through report.

Retailer Contact (Multiple Prompt Values)

List report that accepts multiple prompt values. It is referenced by the Event Studio *User Guide* and the Event Studio Quick Tour.

Returns by Order Method

Report showing the percentage lost by return orders.

Revenue by Date and Amount

Pairing analysis report showing revenues in a specified month whose life to date revenues exceed a specified amount.

Sales Representative Contact List

Report illustrating multiple prompt styles on the same report page. It is also used as a drill-through report.

Sales Reps Performance

Report using a text prompt to query the performance of all sales reps by lowest percentage of sales.

Union Crosstab

Report using tabular table joins to create a complex crosstab.

Waterfall Chart

Waterfall and pareto charts used in combination with a crosstab to show various metrics. Hover with your mouse over Product line to see Product line images.

Sample Reports in the GO Data Warehouse Package

The following reports are found in the GO Data Warehouse package.

Employee Profile

Human resources report showing data for each employee.

Sales Target by Region

Report showing sales target by region, using a prompt to provide values for the report.

Tool Tips

Report that shows tool tips and baselines in chart.

Sample Reports in the Great Outdoors Company Package

The following reports are found in the Great Outdoors Company package.

Margins and Revenue Map for United States

Map report that shows revenue and margins for each sales branch in the United States.

Product Cost by Years

Report that shows product details using a 3-D combination chart and a crosstab display. It is a drill-through target for the Product Report sample in the Analysis Studio Samples folder.

Profit and Revenue Combination Chart

Combination bar chart and line-with-markers chart that shows revenue and gross profit. It also shows the totals for product lines by year.

Revenue by Product Line

Report that uses conditional formatting to highlight revenue. You can also drill through to an Analysis Studio report.

Revenue by Sales Branch

List report that shows revenue by country, sales branch and sales staff.

Revenue by Year over Year Growth

Stacked column chart that shows year-over-year growth. The list report shows revenue growth by product line and year.

Revenue Chart by Sales Territory and Year

Report that contains a chart that compares the revenue contribution of different sales territories to the series total by year. The data is also included in a list report.

Revenue for Specified Order Method and Sales Territory

Report that shows revenue filtered by year, product line, order method and sales territory.

Sales Territory Map

Report that uses a map-style chart to show revenue by sales territory. You can drill through to different reports in Report Studio.

Slicers Filters

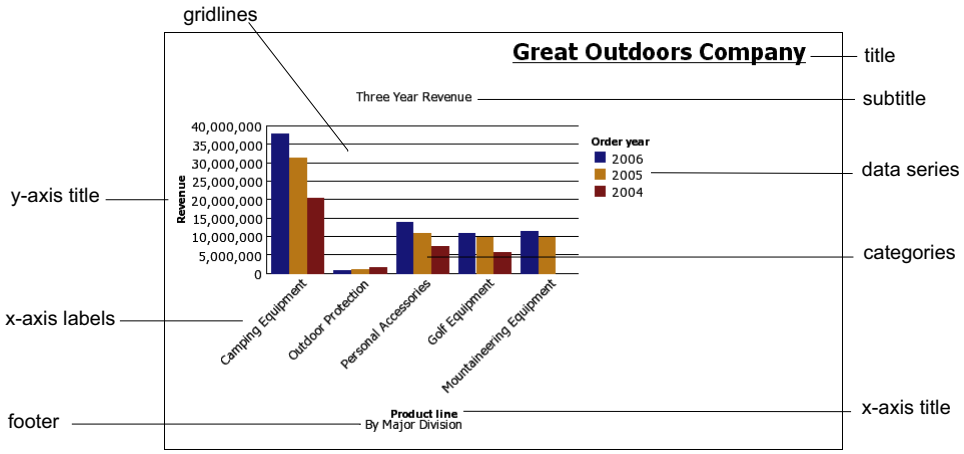
Report that shows dimensional filters applied to the cells of the crosstab but not the rows or columns.

Appendix C: Chart Types

Charts are a graphical way of presenting data. For example, you can create a chart to visualize how actual sales compare to projected sales, or to discover whether sales are falling or rising over quarterly periods.

Chart Elements

This column chart shows the most common chart elements. You can add extra elements such as baselines and notes using Report Studio.



Axes

Axes are lines that provide a frame of reference for measurement or comparison.

The y-axis refers to measures of quantitative data, such as sales figures or quantities. Charts may have more than one y-axis.

The 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.

Gridlines

Axes are lines that provide a frame of reference for measurement or comparison. Major gridlines extend from the tick marks on an axis and run behind the data markers.

Data Series

A data series is a group of related data points plotted in a chart. Each series has a unique color or pattern and is described in the legend.

In the example chart, the data series are order years 2004, 2005, and 2006.

Legend

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

Categories

Categories are groups of related data from the data series, 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 Great Outdoors Company, in clustered columns.

Columns, Lines, and Areas

Charts use columns, lines, and areas as visual representations of data points. Other examples of visual representations include horizontal bars, points, and bubbles.

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.

Purpose	Chart type or configuration
Show contributions of parts to a whole	pie stacked configuration 100% stacked configuration
Show trends in time or contrast values across different categories	line area bar column
Compare groups of related information against actual values	standard configuration radar 3D
Compare different kinds of quantitative information	column-line

Pie Charts

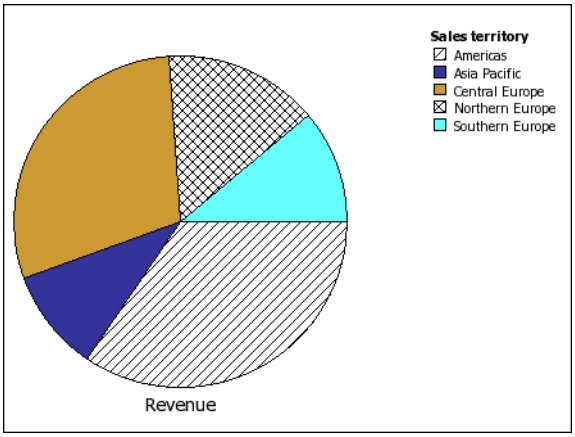
Pie charts are useful for highlighting proportions.

Pie charts use segments of a circle to show the relationship of parts to the whole. To highlight actual values, we recommend that you use another chart type, such as a stacked chart.

Pie charts plot a single data series. To avoid multiple pies when plotting multiple data series, we recommend that you use a 100% stacked chart.

The maximum number of pies that can be shown is 16.

This pie chart shows that the largest proportion of revenue comes from the Americas, followed closely by the Central Europe region.



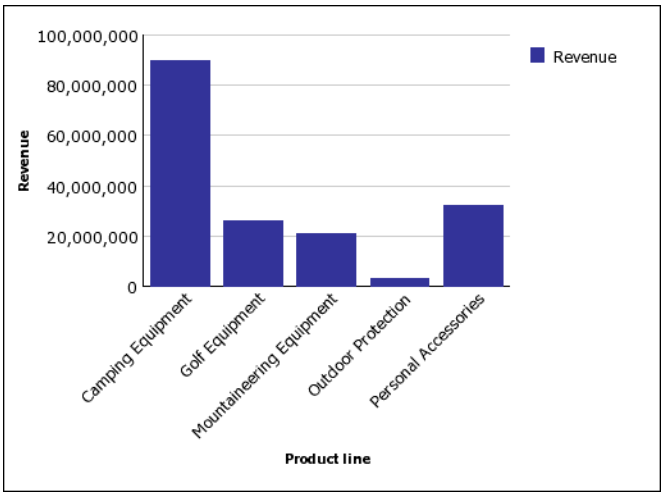
Pie charts can plot data using standard, 100%, and 3D configurations.

Column Charts

Column charts are useful to compare discrete data or to show trends over time.

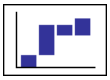
Column charts use vertical data markers to compare individual values.

This column chart uses actual values to show the revenue for each product line.



Column charts can plot data using standard, stacked, 100% stacked, and 3D configurations.

Progressive Column Charts

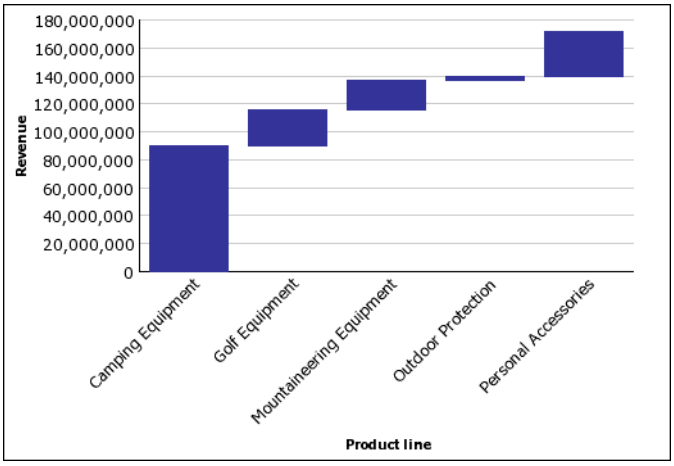


Progressive column charts are like stacked charts with each segment of a single stack displaced vertically from the next segment.

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

Progressive column charts are also known as waterfall charts.

This progressive column chart analyzes the contribution of each product line to revenue.



Progressive column charts can plot data using standard and 3D configurations. You can also create progressive charts using horizontal bars.

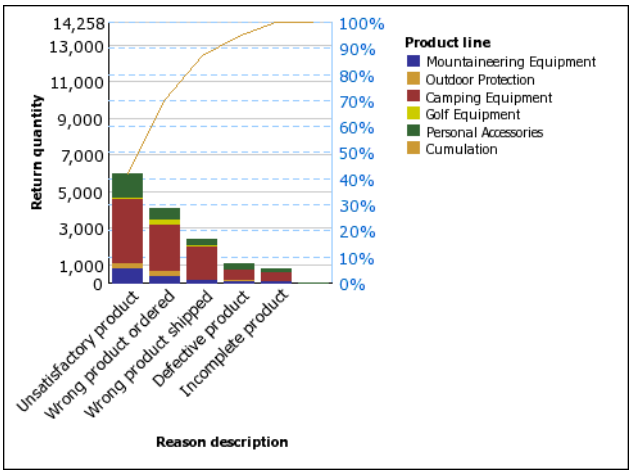
Pareto Charts

Pareto charts are useful for prioritizing and focusing process changes.

Pareto charts rank categories from the most frequent to the least frequent. It is more effective to act on the most frequent causes of events than to solve an easy yet infrequent issue.

You can create before and after comparisons of Pareto charts to show the impact of corrective actions.

This Pareto chart shows that the most frequent reason for product returns is unsatisfactory product.



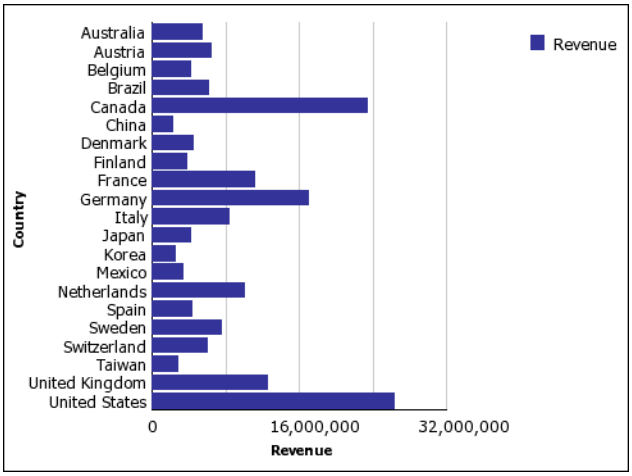
You can also create Pareto charts using horizontal bars.

Bar Charts

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

Bar charts use horizontal data markers to compare individual values.

This bar chart shows actual revenue for every country.



Bar charts can plot data using standard, stacked, and 100% stacked configurations.

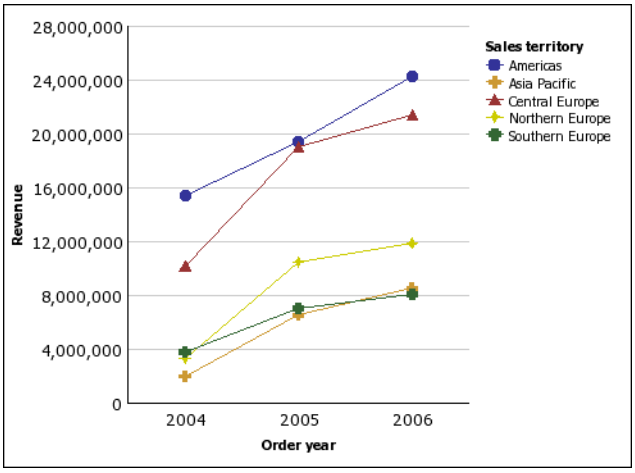
Line Charts

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

Line charts plot data at regular points connected by lines.

We do not recommend that you use stacked line charts because they are difficult to distinguish from unstacked line charts using multiple data series.

This line chart shows a rising revenue trend in every territory.



Line charts can plot data using standard, stacked, 100% stacked, and 3D configurations.

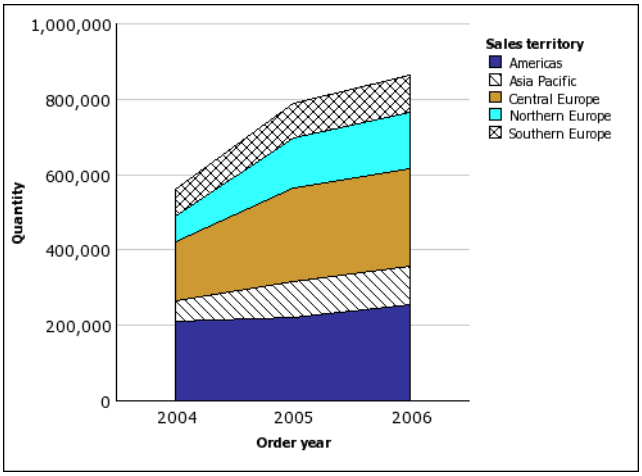
Area Charts

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 that have the areas below the lines filled with colors or patterns.

We do not recommend that you use standard area charts in a chart that has multiple data series because it is possible for areas with lower values to be covered by others.

This stacked area chart shows the quantity of products sold over a two-year period in multiple territories.

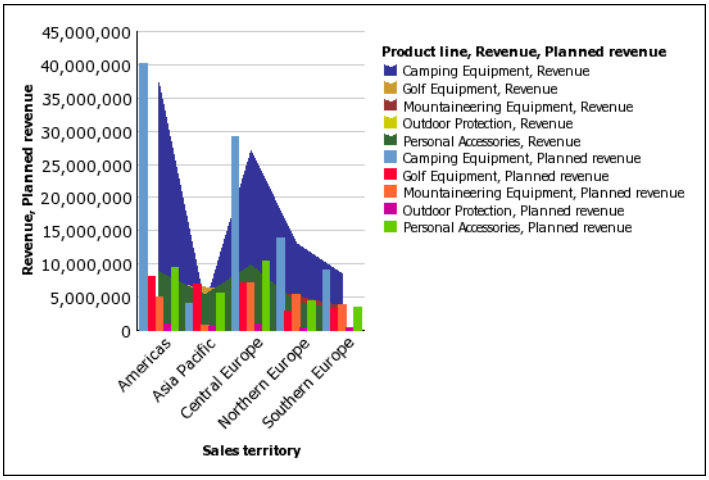


Area charts can plot data using standard, stacked, 100% stacked, and 3D configurations.

Combination Charts

Combination charts are useful for plotting multiple data series by using combinations of columns, areas, and lines.

This combination chart shows planned revenue as a column chart and actual revenue as an area chart.



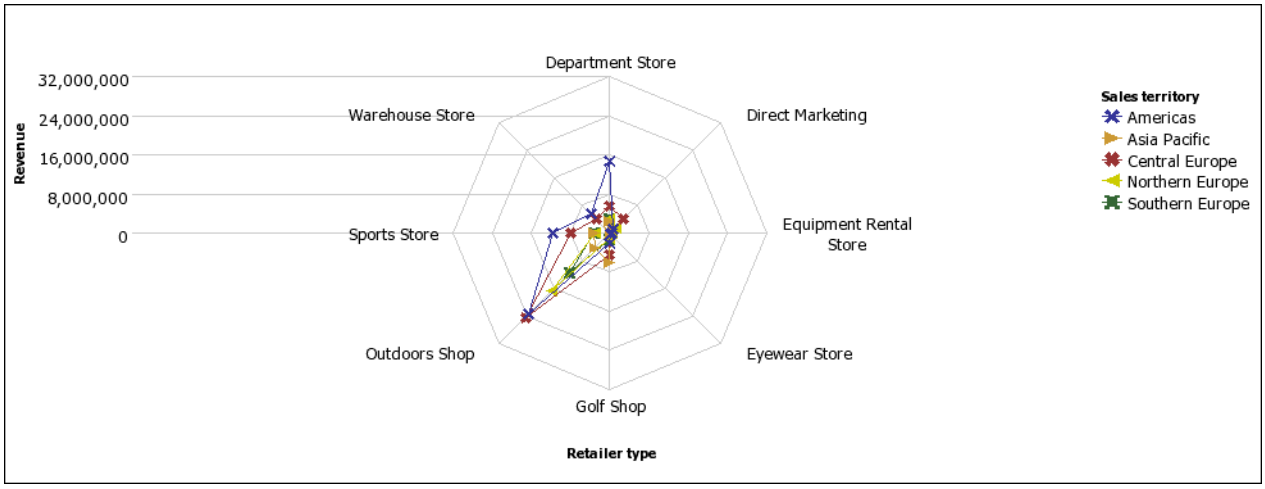
Combination charts can plot data using standard, stacked, 100% stacked, and 3D configurations.

Radar Charts

Radar charts are useful as a comparative tool and for charts with few data series.

Radar charts integrate multiple axes into a single radial figure. Data is plotted on each axis and joined to adjacent axes by connecting lines.

This radar chart shows the revenue from multiple retailer types in multiple territories.



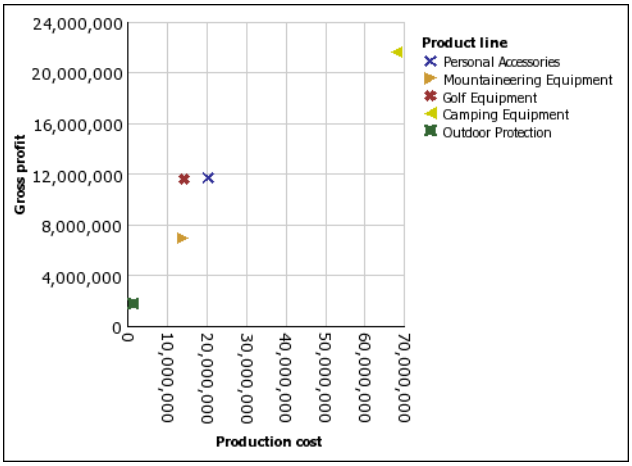
Radar charts can plot data using standard and stacked configurations.

Scatter Charts

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

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

This scatter chart shows the correlation between production cost and gross profit for each product line.

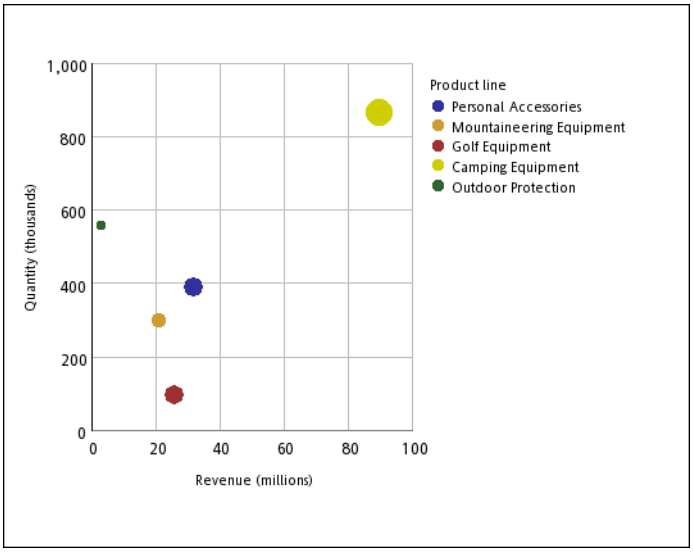


Bubble Charts

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

Bubble charts are useful for visually representing financial data.

This bubble chart plots quantity and revenue by product line. The size of the bubble represents gross profit.



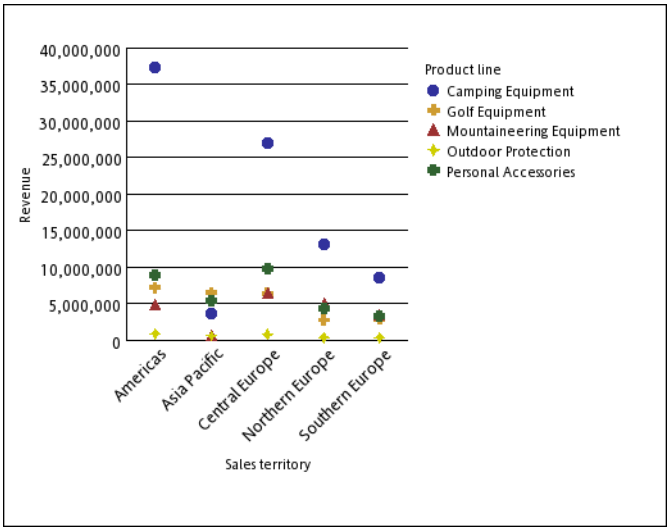
Point Charts

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 is similar to a line chart without the lines. Only the data points are shown.

This point chart shows the revenue for each product line.



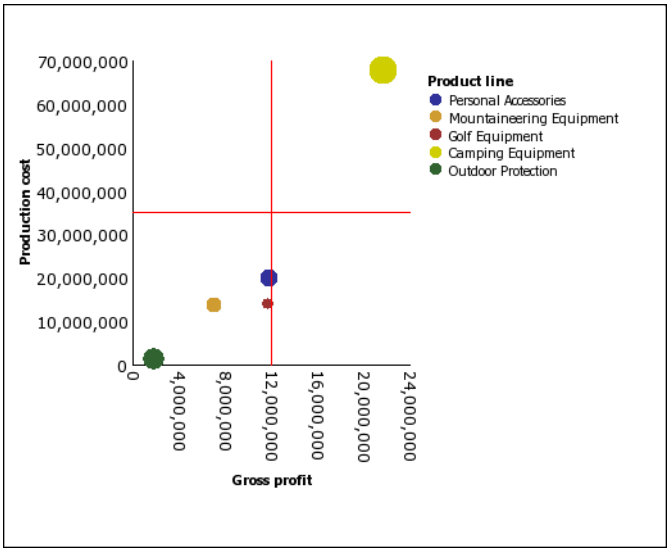
Quadrant Charts

Quadrant charts are useful for plotting data that contains three measures, using an x-axis, a y-axis, and a bubble size that represents the value of the third measure.

Quadrant charts are like bubble charts divided into four equal sections.

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

This quadrant chart shows the relationship between production cost and gross profit. The size of the bubble represents quantity.

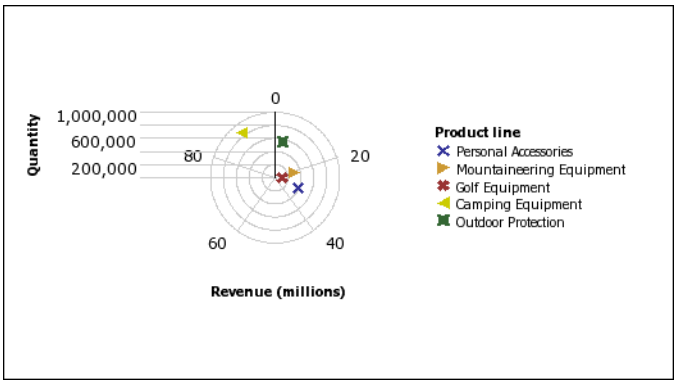


Polar Charts

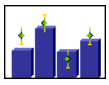
Polar charts are useful for showing scientific data.

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

This polar chart shows the revenue and quantity for each product line. The distance along the radial axis represents revenue while the angle around the polar axis represents quantity.



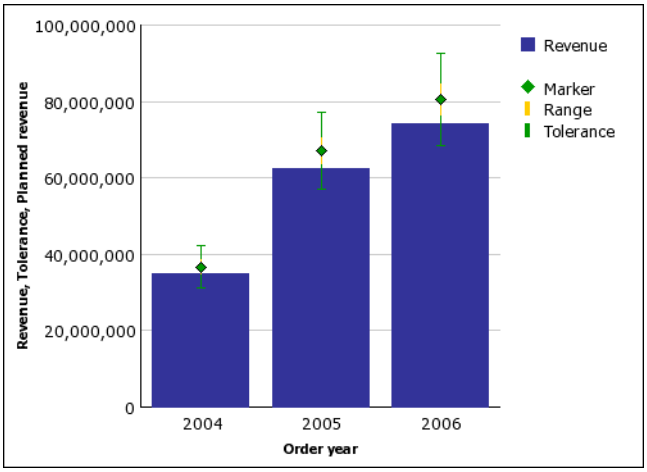
Metrics Range Charts



Metric range charts are useful for showing a target range and a tolerance range.

A metric range chart adds a target and range marker to a column, line, or area chart.

This metric range chart shows actual revenue versus planned revenue.



Gauge Charts

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.

Gauge charts use needles to show information as a reading on a dial. The value for each needle is easily read against the colored data range.

This gauge chart shows the revenue and planned revenue for each sales territory.

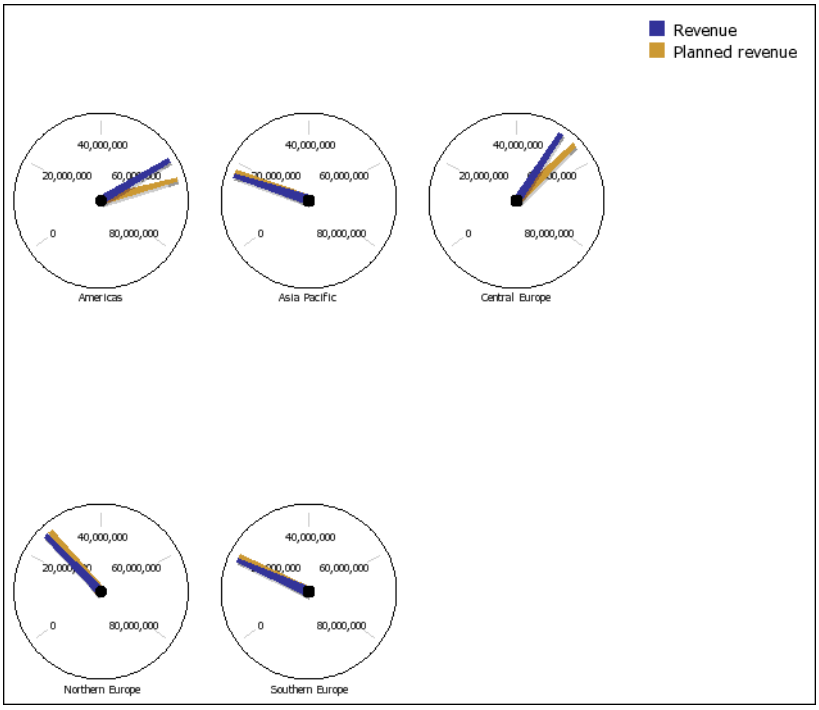


Chart Configurations

Chart configurations specify the grouping type of the columns, bars, lines, and areas in a chart. Some examples are standard, stacked, and 100% stacked charts.

Standard Charts

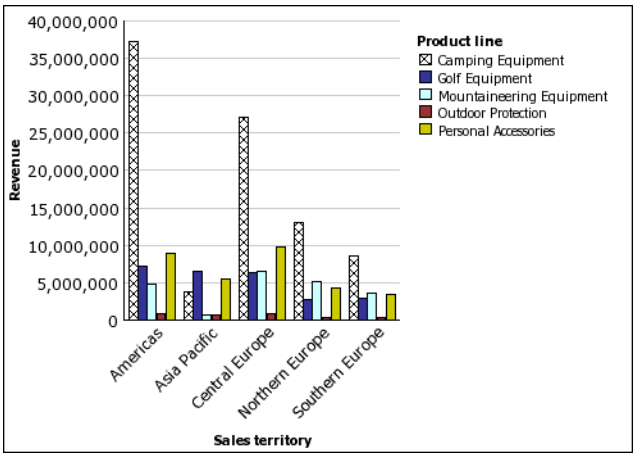
Standard - or absolute - charts are useful for comparing specific values and for representing discrete data, such as different regions or individual employees. For example, a standard column chart that plots regional sales emphasizes the actual value that each region achieves in sales.

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 area and radar charts that have multiple data series, areas with lower values may be covered by others.

This clustered column chart shows the revenue values for each product line within each territory.



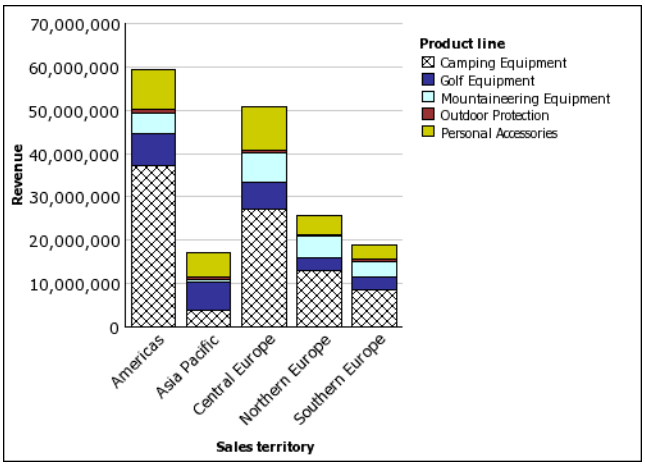
Stacked Charts

Stacked charts are useful for comparing proportional contribution 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.

We recommend that you do not use the stacked configuration in line charts that have multiple data series because it is difficult to distinguish between standard and stacked configurations.

This stacked column chart shows the high proportion that camping equipment contributed to the actual revenue in most markets.



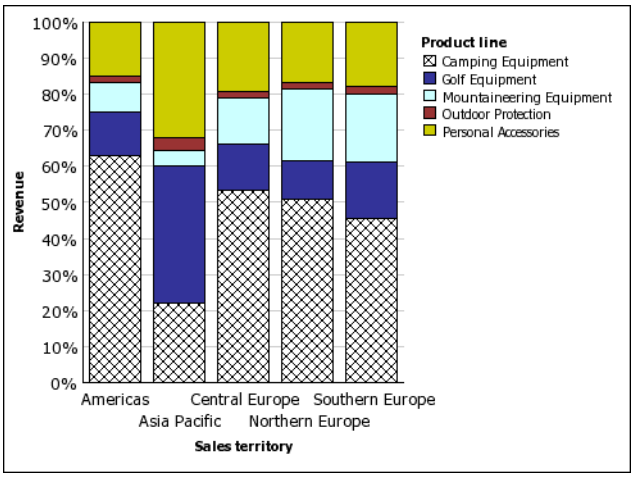
100% Stacked Charts

100% stacked charts are useful for comparing proportional contribution across all categories. They plot the relative contribution of each data series to the total, expressed as a percentage. For example, a 100% 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 per cent.

100% stacked charts highlight proportions. When actual values are important, we recommend that you use another chart configuration.

This 100% stacked chart shows the percentage of sales for each product line in each region.



3-D Charts

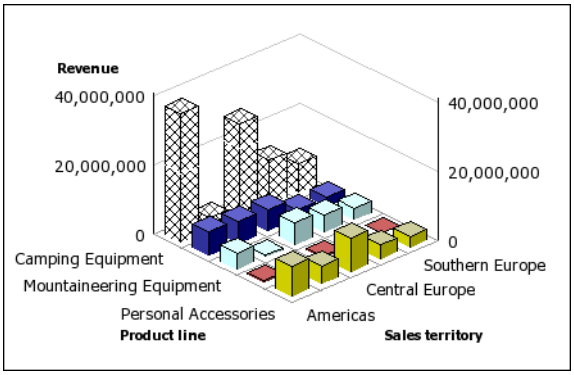
3-D charts are useful for creating charts with high visual content, such as graphics for use in presentations.

3-D column, bar, line, and area charts plot data by using three axes.

3-D pie charts have a three-dimensional visual effect.

We recommend that you do not use 3-D charts where there is a need to show exact values, such as for control or monitoring purposes. The distortion in 3-D charts can make them difficult to read accurately.

This 3-D chart shows actual revenue for each product line in each territory. Note the skipping of the labels on the x and y axes.



Appendix D: Summary Functions

This appendix describes the summary functions that are available in Report Studio. Some functions, such as Custom, are available only when you click the aggregate button on the toolbar. Additional summary functions are available in the Expression Editor (p. 233).

Aggregate

In lists, sets the summary function to **Summarize**. In crosstabs, sets the summary function to **Automatic**.

Automatic

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 a measure
- 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 Aggregation 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.

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.

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.

In crosstabs, this function overrides any solve orders that are specified (p. 114).

Count

Counts all existing values.

If the underlying data source is OLAP, Count behaves as follows if it is specified in the **Rollup Aggregation Function** property for a data item.

Object	Behavior
Level	Count distinct is used. A warning appears when you validate the report.
Member set	Count distinct is used. A warning appears when you validate the report.
Attribute	Not supported. An error is returned when you run the report.
Measure	Supported.

If the underlying data source is Dimensionally Modeled Relational (DMR), Count behaves as follows if it is specified in the **Aggregate Function** or **Rollup Aggregation Function** property for a data item.

Object	Behavior
Level	Count distinct is used. A warning appears when you validate the report.
Member set	Count distinct is used. A warning appears when you validate the report.
Attribute	Supported.
Measure	Supported.

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 Aggregation Function** property for a data item.

Object	Behavior
Level	Supported.
Member set	Supported.
Attribute	Not supported. An error is returned when you run the report.
Measure	Not supported. An error is returned when you run the report.

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 Aggregation 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** (p. 109) is set to **Yes**.

Not Applicable

This function is available only for the **Aggregate Function** property, and it specifies that detail level aggregation does not apply. Instead, aggregates are computed based on the value of the **Rollup Aggregate Function** property.

Use this function only for reports upgraded from Cognos ReportNet(R) 1.x that contain cube data items. In Cognos ReportNet 1.x, summary functions applied to cube data items are equivalent to summary level aggregation in Cognos 8. As a result, only the summary level aggregation is applied.

Standard Deviation

Returns the standard deviation of the selected data item.

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** (p. 109) 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 simply 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 Framework Manager and 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.

Data source	Framework Manager	Report Studio
None specified, or `none`	**Unsupported**	**None**
`average` (`avg`)	**Average**	**Average**
`count`	**Count**	**Count**
`count distinct`	**Count Distinct**	**Count distinct**
`maximum` (`max`)	**Maximum**	**Maximum**
`minimum` (`min`)	**Minimum**	**Minimum**
`median`	**Median**	**Median**
`standard deviation` (`stddev`)	**Standard Deviation**	**Standard Deviation**
`sum`	**Sum**	**Total**
`variance` (`var`)	**Variance**	**Variance**
`count non zero`	**Count Non-Zero**	**Automatic**
`average non zero`	**unknown**	**Automatic**
`external`	**unknown**	**Automatic**
`any`	**unknown**	**Automatic**
`calculated`	**unknown**	**Automatic**
`unknown`	**unknown**	**Automatic**
`first_period`	**unknown**	**Automatic**
`last_period`	**unknown**	**Automatic**

The following summary functions exist only in either Framework Manager or Report Studio. There is no corresponding summary function in data sources.

Framework Manager	Report Studio
Automatic	**Automatic**
Calculated	**Calculated**
No corresponding summary function	**Summarize**
No corresponding summary function	**Not Applicable**

Changes in the Behavior of Count Between Cognos ReportNet(R) 1.1 and Cognos 8

In Cognos 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 Cognos 8, 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 (p. 88) or by counting the literal value 1.

Appendix E: 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.

The expression editor shows the expression components that are supported by the data source in which the metadata is stored. For example, if you import metadata from an Oracle data source, the expression editor shows only the elements that are supported in Oracle.

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)

  This function is not available for any data sources in the package
- limited availability (!!)

  The function is not available for some data sources in the package
- poor performance (!)

  The function is available for all data sources in the package but may have poor performance in some data sources
- unconstrained (no symbol)

  The function is available for all data sources

SAP BW Support

SAP BW does not support all operators or summaries. This can be confusing if you have imported SAP BW metadata and non-SAP BW metadata into the same model.

SAP BW does not support the following operators:

- like
- lookup

SAP BW does not support the following member summaries:

- date-time
- interval
- interval month
- interval day
- interval day to hour
- interval day to minute
- interval day to second
- interval hour
- interval hour to minute
- interval hour to second
- interval minute
- interval minute to second
- interval second
- interval year

- interval year to month
- moving
- running
- time with time zone
- timestamp with time zone

Cell values are date, number, or time. Attribute values are strings.

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 calculations and 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 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 Cognos 8 application server, which slows down the performance of the report.

Browse the Data 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 China. The actual value in the database for China is The People's Republic of China, and this is what you must insert in the filter expression to obtain the correct results.

Steps

1. If you are using a dimensional data source, do the following:
   - In the source, data items, or query tab, right-click the data item you want to browse and click **Search**.
   - Search for the members you want (p. 34).
   - On the search tab, drag the members you want to the **Expression Definition** box and click **OK**.
2. If you are using a relational data source, do the following:
   - In the source, data items, or query tab, right-click the data item you want to browse.
   - If you want to insert a single value, click **Select Value.**
   - If you want to insert multiple values, click **Select Multiple Values**.
   - In the **Values** box, click the value you want to insert in the expression.

     Tip: Use the **Words** box to search for specific values. 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**.
   - If you clicked multiple values, click the right arrow button to add them to the **Selected values** box.
   - Click **Insert**.

     Tip: To control where the values appear in the **Expression Definition** box, click the copy button instead. The values are copied to the clipboard, and you can then paste them where you want.

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'.

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, Cognos 8 uses MUNs to uniquely identify members. MUNs are very complex. For example, the MUN for the member 2004 might appear as follows:

```
[great_outdoors_company].[Years].[Years].[Year]->:[PC].[@MEMBER].[20040101-200
41231]
```

When you are building 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 **Insertable Objects** pane, on the source tab, right-click the member and click **Properties**.

Report Studio has an option that automatically generates an alias for MUNs (p. 28). The MUN alias is a data item that has the MUN as its definition. For the previous example, you would see 2004 as the data item, and the MUN would appear in the data item's **Expression** property. Enable this option to produce a MUN alias whenever you add a member to a report or expression.

Note: 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.

Functions Not Available When Creating a Report or Layout Expression

When you create a report expression or a calculation based on a layout expression, 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 Report Studio.

To see the complete list of functions available in the expression editor, except for report functions, create a detail or group calculation. All functions are available when you create a detail or group calculation because these calculations are performed in the database and not in Report Studio.

Concatenating Strings

When Cognos 8 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 Cognos 8 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.

Dimensional Coercion Rules

To simplify building expressions, Cognos 8 provides coercion rules that automatically convert dimensional types. With coercion rules, you can build simpler expressions, making them easier to understand. In addition to the implicit rules that Cognos 8 provides, you can specify these rules explicitly by using various functions.

Cognos 8 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 (p. 237).
- 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. 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.

```
[Sales].[Products].[].[Product Line] = [Sales].[Products].[].[Product
Line]->[Outdoor Equipment]
```

The left operand is coerced to the following member set:

```
members( [Sales].[Products].[].[Product Line])
```

The following expressions are invalid:

- `[Sales].[Products].[].[Product Line] = NULL`
- `[Sales].[Products].[].[Product Line] + 1`
- `[Sales].[Products].[].[Product Line] = 'Canada'`
- `[Sales].[Products].[].[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:

```
(tuple([Sales].[Products].[].[Product Line]->[Outdoor Equipment]))
```

- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] = NULL`
- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] + 1`
- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] > 2000`
- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] + '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.

Source type	Target type	Result
Measure dimension	Hierarchy	A measure dimension can be used without any conversion as a hierarchy.
Measure dimension	Member set	The member set contains the members of the measure dimension. This coercion is done in two steps. First, measure dimensions are coerced to hierarchies, and then hierarchies are coerced to member sets.
Hierarchy	Member set	A set containing the members of the hierarchy.
Level	Member set	A set containing the members of the level. The equivalent expression is `members(level)`.
Measure	Member	A measure can be used without any conversion as a member.
Measure	Tuple	This coercion is done in two steps. First, measures are coerced to members, and then members are coerced to tuples.
Measure	Member set	This coercion is done in two steps. First, measures are coerced to members, and then members are coerced to member sets.

Source type	Target type	Result
Member	Tuple	The tuple consists of the member. The equivalent expression is `tuple(member)`
Member	Member set	The set consists of the member. This coercion is done in two steps. First, members are coerced to tuples, and then tuples are coerced to member sets.

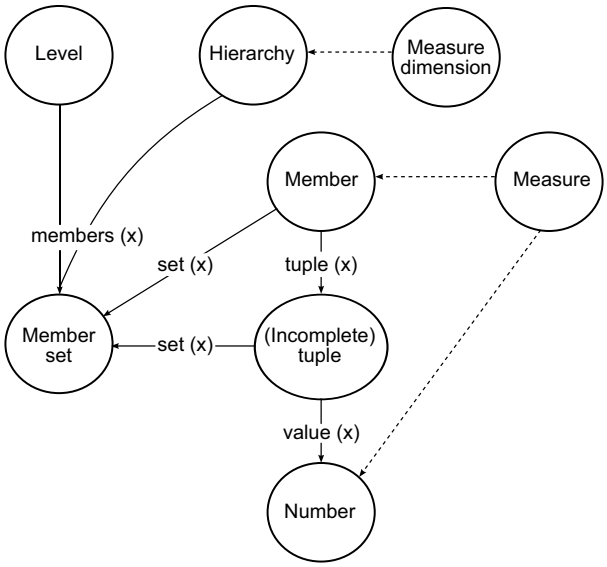
Coercion Rules for Numeric and String Values

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.

Source type	Target type	Result
Measure	Number, Date/Time, or String	The result is the value of the measure at the current cube intersection.
Member	Number	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.
Tuple	Number	The value of the cell at the cube intersection specified by the tuple. The equivalent expression is `value(operand or parameter)`

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.



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.

(

Inserts an open parenthesis in your expression.

Syntax

```
(expression)
```

)

Inserts a closed parenthesis in your expression.

Syntax

```
(expression)
```

*

Multiplies two numeric values.

Syntax

```
value1 * value2
```

,

Separates expression components.

Syntax

```
expression(param1, param2)
```

/

Divides two numeric values.

Syntax

```
value1 / value2
```

||

Concatenates strings.

Syntax

```
string1 || string2
```

+

Adds two values.

Syntax

```
value1 + value2
```

-

Subtracts two numeric values or negates a numeric value.

Syntax

```
value1 - value2
or
- value
```

<

Compares values against a defined value and retrieves the values that are less than the defined value.

Syntax

```
value1 < value2
```

<=

Compares values against a defined value and retrieves the values that are less than or equal to the defined value.

Syntax

```
value1 <= value2
```

<>

Compares values against a defined value and retrieves the values that are not equal to the defined value.

Syntax

```
value1 <> value2
```

=

Compares values against a defined value and retrieves the values that are equal to the defined value.

Syntax

```
value1 = value2
```

>

Compares values against a defined value and retrieves the values that are greater than the defined value.

Syntax

```
value1 > value2
```

->

Used as a separator in a literal member expression.

Syntax

```
[namespace].[dimension].[hierarchy].[level]->[L1]
```

>=

Compares values against a defined value and retrieves the values that are greater than or equal to the defined value.

Syntax

```
value1 >= value2
```

and

Returns true if the conditions on both sides of the expression are true.

Syntax

```
arg1 AND arg2
```

auto

Use 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

```
name BETWEEN value1 and value2
```

case

Use with When, Then, Else, and End.

Syntax

```
CASE expression { WHEN expression THEN expression } [ ELSE expression ] END
```

contains

Determines if a string contains another string.

Syntax

```
string1 CONTAINS string2
```

currentMeasure

Keyword that can be used as the first argument of member summary functions.

Syntax

```
aggregate_function( currentMeasure within set expression )
```

default

Use with 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

Use with If or Case constructs.

Syntax

```
IF (condition) THEN .... ELSE (expression) , or CASE .... ELSE expression END
```

end

Use with Case When construct.

Syntax

```
CASE .... END
```

ends with

Determines if a string ends with a given string.

Syntax

```
string1 ENDS WITH string2
```

for

Use with summary expressions to define the scope of the aggregation in the query.

Syntax

```
aggregate_function ( expression FOR expression { , expression } )
```

for all

Use with summary expressions to define the scope to be all the specified grouping columns in the query. See also FOR clause.

Syntax

```
aggregate_function ( expression FOR ALL expression { , expression } )
```

for any

Use with summary expressions to define the scope to be adjusted based on a subset of the grouping columns in the query. Equivalent to FOR clause.

Syntax

```
aggregate_function ( expression FOR ANY expression { , expression } )
```

for report

Use with summary expressions to define the scope to be the whole query. See also FOR clause.

Syntax

```
aggregate_function ( expression FOR REPORT )
```

if

Use with Then and Else.

Syntax

```
IF (condition is true) THEN (action) ELSE (alternate action)
```

in

Determines if a value exists in a given list of values.

Syntax

```
exp1 IN (exp_list)
```

in_range

Determines if an item exists in a given list of constant values or ranges.

Syntax

```
expression IN_RANGE { constant : constant [ , constant : constant ] }
```

is missing

Determines if a value is undefined in the data.

Syntax

```
value IS MISSING
```

is null

Determines if a value is undefined in the data.

Syntax

```
value IS NULL
```

is not missing

Determines if a value is defined in the data.

Syntax

```
value IS NOT MISSING
```

is not null

Determines if a value is defined in the data.

Syntax

```
value IS NOT NULL
```

like

Determines if a string matches the pattern of another string.

Syntax

```
string1 LIKE string2
```

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 the condition is false, otherwise returns false.

Syntax

```
NOT arg
```

or

Returns true if either of the two conditions on both sides of the expression is true.

Syntax

```
arg1 OR arg2
```

prefilter

Performs a summary calculation before applying the summary filter.

Syntax

```
summary ([expression] PREFILTER)
```

rows

Counts the number of rows output by the query. Use with Count().

Syntax

```
count(ROWS)
```

starts with

Determines if a string starts with a given string.

Syntax

```
string1 STARTS WITH string2
```

then

Use with If or Case constructs.

Syntax

```
IF (condition) THEN ...., or CASE expression WHEN expression THEN .... END
```

when

Use with Case construct.

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.

aggregate

Returns a calculated value using the appropriate aggregation function, based on the aggregation type of the expression.

Syntax

```
aggregate ( expr [ auto ] )
aggregate ( expr for [ all | any ] expr { , expr } )
aggregate ( expr for report )
```

average

Returns the average value of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
average ( [ distinct ] expr [ auto ] )
average ( [ distinct ] expr for [ all | any ] expr { , expr } )
average ( [ distinct ] expr for report )
```

Example

```
average ( Sales )
```

Result: The average of all Sales values.

count

Returns the number of selected data items excluding NULL values. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
count ( [ distinct ] expr [ auto ] )
count ( [ distinct ] expr for [ all | any ] expr { , expr } )
count ( [ distinct ] expr for report )
```

Example

```
count ( Sales )
```

Result: The total number of entries under Sales.

maximum

Returns the maximum value of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
maximum ( [ distinct ] expr [ auto ] )
maximum ( [ distinct ] expr for [ all | any ] expr { , expr } )
maximum ( [ distinct ] expr for report )
```

Example

```
maximum ( Sales )
```

Result: The maximum value of all Sales values.

median

Returns the median value of selected data items.

Syntax

```
median ( expr [ auto ] )
median ( expr for [ all | any ] expr { , expr } )
median ( expr for report )
```

minimum

Returns the minimum value of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
minimum ( [ distinct ] expr [ auto ] )
minimum ( [ distinct ] expr for [ all | any ] expr { , expr } )
minimum ( [ distinct ] expr for report )
```

Example

```
minimum ( Sales )
```

Result: The minimum value 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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
moving-average ( numeric_expr , numeric_expr  [ at exp {, expr } ]  [
<for-option> ] [ prefilter ] )
moving-average ( [ distinct ] numeric_expr , numeric_expr [ <for-option> ] [
prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
moving-average ( Qty, 3 )
```

Result: For each row, this displays the quantity and a moving average of the current row and the preceding two rows.

```
Qty           Moving-Average (Qty, 3)
------        -----------------------------------
200           NULL
700           NULL
400           433.3333
200           433.3333
200           266.6667
500           300.0000
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
moving-total ( numeric_expr , numeric_expr  [ at exp {, expr } ]  [
<for-option> ] [ prefilter ] )
moving-total ( [ distinct ] numeric_expr , numeric_expr [ <for-option> ] [
prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
moving-total ( Quantity, 3 )
```

Result: For each row, this displays the quantity and a moving total of the current row and the preceding two rows.

```
Qty                 Moving-Total (Qty, 3)
------              ------------------------
200                 NULL
700                 NULL
400                 1300
200                 1300
200                 800
500                 900
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
percentage ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [ prefilter ]
)
percentage ( [ distinct ] numeric_expr [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
percentage ( sales 98 )
```

Result: Shows the percentage of the total sales for 1998 that is attributed to each sales representative.

```
Sales Rep           Sales 98            Percentage
------------        --------------      ----------------
Bill Gibbons        60646               7.11%
Bjorn Flertjan      62523               7.35%
Chris Cornel        22396               2.63%
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
percentile ( numeric_expr  [ at exp {, expr } ] [ <for-option> ] [ prefilter ]
)
percentile ( [ distinct ] numeric_expr [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
percentile ( Sales 98 )
```

Result: For each row, displays the percentage of rows that are equal to or less than the quantity value of that row.

```
Qty                 Percentile (Qty)
------              ----------------
800                 1
700                 0.875
600                 0.75
500                 0.625
400                 0.5
400                 0.5
200                 0.25
200                 0.25
```

quantile

Returns the rank of a value for 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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
quantile ( numeric_expr, numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
quantile ( [ distinct ] numeric_expr, numeric_expr  [ <for-option> ] [
prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
quantile ( Qty, 4 )
```

Result: The quantity, the rank of the quantity value, and the quantity values broken down into 4 quantile groups (quartiles).

```
Qty                 Rank (Qty)          Quantile (Qty, 4)
------              ----------------    ----------------
800                 1                   1
700                 2                   1
600                 3                   2
500                 4                   2
400                 5                   3
400                 5                   3
200                 7                   4
200                 7                   4
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
quartile ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [ prefilter ]
)
quartile ( [ distinct ] numeric_expr [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
quartile ( Qty )
```

Result: This displays the quantity and the quartile of the quantity value represented as integers from 1 (highest) to 4 (lowest).

```
Qty                 Quartile (Qty)
------              --------------
450                 1
400                 1
350                 2
300                 2
250                 3
200                 3
150                 4
100                 4
```

rank

Returns the rank value of selected data items. 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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product. Null values are ranked last.

Syntax

```
rank ( expr [sort_order] {, expr [sort_order] }  [ at exp {, expr } ]  [
<for-option> ] [ prefilter ] )
rank ( [ distinct ] expr [sort_order] {, expr [sort_order] } [ <for-option>]  [
prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
rank ( Sales 98 )
```

Result: For each row, this displays the rank value of sales for 1998 that is attributed to each sales representative, and skips some numbers when there is a tie between rows.

```
Sales Rep           Sales 98            Rank
----------------    -----------         ------------
Bill Gibbons        60000               1
Bjorn Flertjan      50000               2
Chris Cornel        50000               2
John Smith          48000               4
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-average ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
running-average ( [ distinct ] numeric_expr [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-average ( Qty )
```

Result: For each row, this displays the quantity and a running average of the current and the previous rows.

```
Name          Qty           Avg           Running-Average for name
-------       ------        ------        ---------------------------------
Smith         7             5             7
Smith         3             5             5
Smith         6             5             5.33
Smith         4             5             5
Wong          3             4             3
Wong          5             4             4
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-count ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
running-count ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-count ( Qty )
```

Result: For each row, this displays the quantity and a running count of the position of current row.

```
Name          Qty           Count         Running-Count for name
-------       ------        ---------     ---------------------------
Smith         7             4             1
Smith         3             4             2
Smith         6             4             3
Smith         4             4             4
Wong          3             3             1
Wong          5             3             2
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-difference ( numeric_expr [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
running-difference ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ]
)
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-difference ( Qty )
```

Result: For each row, this displays the quantity and a running difference between the value for the current row and the preceding row.

```
Name          Qty           Running-Difference for name
-------       ------        -------------------------------------
Smith         7             NULL
Smith         3             -4
Smith         6             3
Smith         4             -2
Wong          3             -1
Wong          5             2
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-maximum (  numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
running-maximum ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-maximum ( Qty )
```

Result: For each row, this displays the quantity and a running maximum of the current and previous rows.

```
Name          Qty           Max           Running-Maximum (Qty) for name
-------       ------        ------        ----------------------------
Smith         2             7             2
Smith         3             7             3
Smith         6             7             6
Smith         7             7             7
Wong          3             5             3
Wong          5             5             5
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-minimum ( numeric_expr  [ at exp {, expr } ]  [ <for-option> ] [
prefilter ] )
running-minimum ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-minimum ( Qty )
```

Result: For each row, this displays the quantity and a running minimum of the current and previous rows.

```
Name          Qty           Min           Running-Minimum (Qty) for name
-------       -----         ------        ------------------------------
Smith         7             2             7
Smith         3             2             3
Smith         6             2             3
Smith         2             2             2
Wong          4             3             4
Wong          5             3             4
```

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 only be used in the context of relational datasources. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
running-total ( numeric_expr [ at exp {, expr } ]   [ <for-option> ] [
prefilter ] )
running-total ( [ distinct ] numeric_expr  [ <for-option> ] [ prefilter ] )
<for-option> ::= for expr {, expr } | for report | auto
```

Example

```
running-total ( Qty )
```

Result: For each row, this displays the quantity and a running total of the current and previous rows.

```
Name          Qty           Total         Running-Total (Qty) for name
-------       ------        --------      ------------------------------
Smith         2             18            2
Smith         3             18            5
Smith         6             18            11
Smith         7             18            18
Wong          3             12            3
Wong          5             12            8
```

standard-deviation

Returns the standard deviation of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
standard-deviation ( [ distinct ] expr [ auto ] )
standard-deviation ( [ distinct ] expr for [ all | any ] expr { , expr } )
standard-deviation ( [ distinct ] expr for report )
```

Example

```
standard-deviation ( ProductCost )
```

Result: 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. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
standard-deviation-pop ( [ distinct ] expr [ auto ] )
standard-deviation-pop ( [ distinct ] expr for [ all | any ] expr { , expr } )
standard-deviation-pop ( [ distinct ] expr for report )
```

Example

```
standard-deviation-pop ( ProductCost )
```

Result: A value of the square root of the population variance.

total

Returns the total value of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
total ( [ distinct ] expr [ auto ] )
total ( [ distinct ] expr for [ all | any ] expr { , expr } )
total ( [ distinct ] expr for report )
```

Example

```
total ( Sales )
```

Result: The total value of all Sales values.

variance

Returns the variance of selected data items. The keyword distinct is available for backward compatibility of expressions used in previous versions of the product.

Syntax

```
variance ( [ distinct ] expr [ auto ] )
variance ( [ distinct ] expr for [ all | any ] expr { , expr } )
variance ( [ distinct ] expr for report )
```

Example

```
variance ( Product Cost )
```

Result: 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 ( [ distinct ] expr [ auto ] )
variance-pop ( [ distinct ] expr for [ all | any ] expr { , expr } )
variance-pop ( [ distinct ] expr for report )
```

Example

```
variance-pop ( Qty)
```

Result: For each row, this displays the population variance of a set of numbers after discarding the nulls in this set.

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 | expr > within set set_expr { , set_expr } )
aggregate ( < currentMeasure | expr > within < detail | aggregate > expr )
```

average

Returns the average value of selected data items.

Syntax

```
average ( < currentMeasure | expr > within set set_expr { , set_expr } )
average ( < currentMeasure | expr > within < detail | aggregate > expr )
```

count

Returns the number of selected data items excluding NULL values.

Syntax

```
count ( < currentMeasure | expr > within set set_expr { , set_expr } )
count ( < currentMeasure | expr > within < detail | aggregate > expr )
```

maximum

Returns the maximum value of selected data items.

Syntax

```
maximum ( < currentMeasure | expr > within set set_expr { , set_expr } )
maximum ( < currentMeasure | expr > within < detail | aggregate > expr )
```

median

Returns the median value of selected data items.

Syntax

```
median ( < currentMeasure | expr > within set set_expr { , set_expr } )
median ( < currentMeasure | expr > within < detail | aggregate > expr )
```

minimum

Returns the minimum value of selected data items.

Syntax

```
minimum ( < currentMeasure | expr > within set set_expr { , set_expr } )
minimum ( < currentMeasure | expr > within < detail | aggregate > expr )
```

percentage

Returns the percent of the total value for selected data items.

Syntax

```
percentage ( numeric_expr  [ tuple member_expr {, member_expr } ] within set
set_expr  {, set_expr } )
```

Example

```
percentage ( [gosales].[sales measures].[quantity] tuple
[gosales].[Staff].[].[department]->[West] within set
children([gosales].[Staff].[].[Staff] )
```

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.

Syntax

```
percentile ( numeric_expr  [ tuple member_expr {, member_expr } ] within set
set_expr  {, set_expr } )
```

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_expr, numeric_expr  [ tuple member_expr {, member_expr } ]
within set set_expr  {, set_expr } )
```

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_expr [ tuple member_expr {, member_expr } ] within set
set_expr  {, set_expr } )
```

rank

Returns the rank value of 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_expr [ ASC | DESC ] [ tuple member_expr {, member_expr } ]
within set set_expr  {, set_expr } )
```

Example

```
rank  ( [gosales].[sales measures].[quantity] tuple
[gosales].[Staff].[].[department]->[West] within set
children([gosales].[Staff].[].[Staff] )
```

standard-deviation

Returns the standard deviation of selected data items.

Syntax

```
standard-deviation ( < currentMeasure | expr > within set set_expr { , set_expr
} )
standard-deviation ( < currentMeasure | expr > within < detail | aggregate >
expr )
```

total

Returns the total value of selected data items.

Syntax

```
total ( < currentMeasure | expr > within set set_expr { , set_expr } )
total ( < currentMeasure | expr > within < detail | aggregate > expr )
```

variance

Returns the variance of selected data items.

Syntax

```
variance ( < currentMeasure | expr > within set set_expr { , set_expr } )
variance ( < currentMeasure | expr > within < detail | aggregate > expr )
```

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.

interval year

Inserts a zero year interval.

interval month

Inserts a zero month interval.

interval year to month

Inserts a zero year to month interval.

interval day

Inserts a zero day interval.

interval hour

Inserts a zero hour interval.

interval minute

Inserts a zero minute interval.

interval second

Inserts a zero second interval.

interval day to hour

Inserts a zero day to hour interval.

interval day to minute

Inserts a zero day to minute interval.

interval day to second

Inserts a zero day to second interval.

interval hour to minute

Inserts a zero hour to minute interval.

interval hour to second

Inserts a zero hour to second interval.

interval minute to second

Inserts a zero minute to second interval.

null

Inserts a null value if the expression conditions are not met.

number

Inserts the number 0, which you can replace with a new numeric value.

string

Inserts an empty string.

time

Inserts the current system time.

Constructs

if then else

Syntax

```
IF ([Country] = 'Canada')
    THEN ([List Price] * 0.60)
    ELSE ([List Price])
```

in_range

Specify one or more constants or ranges. A range can be open ended.

Example

```
[gosales].[CONVERSIONRATE].[COUNTRYCODE] IN_RANGE { :30 , 40, 50, 999: }
```

search case

Syntax

```
CASE
    WHEN [Country] = 'Canada' THEN ([List Price] * 0.60)
    WHEN [CountryCode] > 100 THEN [List Price] * 0.80
    ELSE [List Price]
END
```

simple case

Syntax

```
CASE [Country]
    WHEN 'Canada' THEN ([List Price] * 0.60)
    WHEN 'Australia' THEN [List Price] * 0.80
    ELSE [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 dependent on the first argument resulting from adding integer_exp days to date_exp.

Syntax

```
_add_days ( date_exp, integer_exp )
```

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

```
_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 dependent on the first argument resulting from adding integer_exp months to date_exp.

Syntax

```
_add_months ( date_exp, integer_exp )
```

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 dependent on the first argument resulting from adding integer_exp years to date_exp.

Syntax

```
_add_years ( date_exp, integer_exp )
```

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_exp from today's date. This 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_exp )
```

Example

```
Today's date=2003-02-05 _age ( 1990-04-30 )
```

Result: 120906 that is 12 years, 9 months and 6 days

_day_of_week

Returns the day of week (between 1 and 7), where 1 is the first day of the week as indicated by the second parameter(between 1 and 7, 1 being Monday and 7 being Sunday). Note that in ISO 8601 standard, a week begins with Monday being day 1. In North America where Sunday is the first day of the week being day 7.

Syntax

```
_day_of_week ( date_exp, integer )
```

Example

```
_day_of_week ( 2003-01-01, 1 )
```

Result: 3

_day_of_year

Returns the ordinal for the day of the year in date_ exp (1 to 366). Also known as Julian day.

Syntax

```
_day_of_year ( date_exp )
```

Example

```
_day_of_year ( 2003-03-01 )
```

Result: 61

_days_between

Returns a positive or negative number representing the number of days between the two date expressions. If date_exp1 < date_exp2 then the result will be a negative number.

Syntax

```
_days_between ( date_exp1, date_exp2 )
```

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 the date expression date_exp.

Syntax

```
_days_to_end_of_month ( date_exp )
```

Example

```
_days_to_end_of_month ( 2002-04-20 14:30:22.123 )
```

Result: 10

_first_of_month

Returns a date or datetime dependent on the argument obtained from converting date_exp to a date with the same year and month but the day set to 1.

Syntax

```
_first_of_month ( date_exp )
```

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 dependent on the argument that is the last day of the month represented by date_exp.

Syntax

```
_last_of_month ( date_exp )
```

Example 1

```
_last_of_month ( 2002-01-14 )
```

Result: 2002-01-31

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_exp1 (the year), integer_exp2 (the month) and integer_exp3 (the day).

Syntax

```
_make_timestamp ( integer_exp1, integer_exp2, integer_exp3 )
```

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_exp1 to date_exp2. If date_exp1 < date_exp2, then a negative number is returned.

Syntax

```
_months_between ( date_exp1, date_exp2 )
```

Example

```
_months_between ( 2002-01-30, 2002-04-03 )
```

Result: 2

_week_of_year

Returns the number of the week of the year of the date_exp according to ISO 8601, in which 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.

Syntax

```
_week_of_year ( date_exp )
```

Example

```
_week_of_year ( 2003-01-01 )
```

Result: 1

_years_between

Returns a positive or negative integer number representing the number of years from date_exp1 to date_exp2. If date_exp1 < date_exp2 then a negative value is returned.

Syntax

```
_years_between ( date_exp1, date_exp2 )
```

Example

```
_years_between ( 2003-01-30, 2001-04-03 )
```

Result: 1

_ymdint_between

Returns a number representing the difference between the date expressions date_exp1 and date_exp2. This 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_exp1, date_exp2 )
```

Example

```
_ymdint_between ( 1990-04-30 , 2003-02-05 )
```

Result: 120906 that is 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_exp_max + numeric_exp_overflow. If numeric_exp_max + numeric_exp_overflow is exceeded, then only the max number of members are returned.

Syntax

```
_firstFromSet ( set_exp, numeric_exp_max , numeric_exp_overflow )
```

_remainderSet

The member expression will be included in the returned set when the size of the set_exp set is greater than numeric_exp.

Syntax

```
_remainderSet (member_exp,  set_exp , numeric_exp )
```

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, which is executed at run time.

+

Concatenates two strings.

Syntax

```
value1 + value2
```

Example

```
# '{' + $runLocale + '}'#
```

Result: {en-us}

array

Constructs an array out of the list of parameters.

Syntax

```
array ( string_exp | array_exp { , string_exp | array_exp } )
```

Example

```
#csv ( 'x1' , 'x2' , array ( 'a1' , 'a2' ) )#
```

Result: 'x1' , 'x2' , 'a1' , 'a2'

csv

Constructs a comma separated values string from the elements of the array. 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_exp [ , separator_string [ , quote_string ] ] )
```

Example

```
#csv ( array ( 'a1' , 'a2' ) )#
```

Result: 'a1' , 'a2'

dq

Surround the passed string with double quotes.

Syntax

```
dq ( string_exp )
```

Example

```
#dq ( 'zero' )#
```

Result: "zero"

grep

Searches for elements of an array that match the pattern specified in the first argument. It returns an array with the elements that pass the pattern.

Syntax

```
grep ( pattern_string , array_exp )
```

Example

```
#csv ( grep ( 's' , array ( 'as', 'an', 'arts' ) ) )#
```

Result: 'as', 'arts'

join

Joins the elements of an array using the separator string.

Syntax

```
join ( separator_string , array_exp )
```

Example

```
# sq ( join ( ' | | ' , array ( 'as', 'an', 'arts' ) ) )#
```

Result: 'as | | an | | arts'

prompt

Prompt the user for a single value. Only the prompt_name argument is required. The datatype defaults to string when not specified. The prompt is optional when defaultText is specified. The text, when specified, will precede the value. A queryItem can be specified to take advantage of the Prompt Info properties of the query item. The trailing_text, when specified, will be appended to the value.

Syntax

```
prompt ( prompt_name , datatype , defaultText , text , queryItem ,
trailing_text )
```

Example

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE  > #prompt('Starting
CountryCode',
                   'integer',
                   '10'
)#
```

Result: select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE > 10

promptmany

Prompt the user for one or more values. Only the prompt_name argument is required. The datatype defaults to string when not specified. The prompt is optional when defaultText is specified. The text, when specified, will precede the value. A queryItem can be specified to take advantage of the Prompt Info properties of the query item. The trailing_text, when specified, will be appended to the value.

Syntax

```
promptmany ( prompt_name , datatype , defaultText , text , queryItem ,
trailing_text )
```

Example

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY IN ( #promptmany (
'CountryName' ) # )
```

Result: select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE IN ('Canada' , 'The Netherlands' , 'Russia')

sb

Surround the passed string with square brackets.

Syntax

```
sb ( string_exp )
```

Example

```
#sb ( 'abc' )#
```

Result: [abc]

sq

Surround the passed string with single quotes.

Syntax

```
sq ( string_exp )
```

Example

```
#sq ( 'zero' )#
```

Result: 'zero'

sort

Sorts the elements of the array in alphabetical order. Duplicates are retained.

Syntax

```
sort ( array_exp )
```

Example

```
#csv ( sort ( array ( 's3', 'a', 'x' ) ) )#
```

Result: 'a', 's3', 'x'

split

Splits a string or the string elements of the array into separate elements.

Syntax

```
split ( pattern_string, string_exp | array_exp )
```

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'

substitute

Search for a pattern in a string or in the string elements of an array and substitute the found text with other text.

Syntax

```
substitute ( pattern_string, replacement_string, string_exp | array_exp )
```

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'

unique

Removes duplicate entries from the array. The order of the elements is retained.

Syntax

```
unique ( array_exp )
```

Example

```
Example:
#csv ( unique ( array ( 's3', 'a', 's3', 'x' ) ) )#
```

Result: 's3', 'a', 'x'

urlencode

URL encodes the passed argument. Useful when specifying XML connection strings.

Syntax

```
field_one=urlencode(prompt('userValue'))
```

Example

```
urlencode(prompt('some_val'))
```

Result: `%27testValue%27`

CSVIdentityName

Use the identity information of the current authenticated user to lookup 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 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

```
CSVIdentityNameList ( [ separator_string ] )
```

Example

```
#CSVIdentityNameList ( )#
```

Result: 'Everyone' , 'Report Administrators' , 'Query User'

CAMPassport

Returns the passport.

Syntax

```
CAMPassport ( )
```

Example

```
#CAMPassport ( )#
```

Result: 111:98812d62-4fd4-037b-4354-26414cf7ebef:3677162321

CAMIDList

Returns the pieces of the user's identity (account name, group names, role names) as a list of values separated by commas.

Syntax

```
CAMIDList ( [ separator_string ] )
```

Example

```
#CAMIDList ( )#
```

Result: CAMID("::Everyone"), CAMID(":Authors"), CAMID(":Query Users"), CAMID(":Consumers"), CAMID(":Metrics Authors")

CAMIDListForType

Returns an array of the user's identities based on the identity type (account, group, or role). It can be used with the macro functions csv or join.

Syntax

```
CAMIDListForType ( identity type )
```

Example

```
[qs].[userRole] IN ( #csv ( CAMIDListForType ( 'role' ) ) # )
```

Result: [qs].[userRole] IN ('Administrator', 'developer')

Common Functions

nullif

Returns NULL if exp1 equals exp2, otherwise returns exp1.

Syntax

```
nullif( exp1, exp2 )
```

_format

Associates a format with the expression. The keyword can be PERCENTAGE_0, PERCENTAGE_1 or PERCENTAGE_2.

Syntax

```
_format ( expr , keyword  )
```

Example

```
_format( [Unit Sale Price] / [Unit Price] , PERCENTAGE_2 )
```

Result: 0.75123 displayed as 75.12%

_round

Returns the numeric expression rounded to the integer_exp places right of the decimal point. Note: integer_exp MUST be a non-negative integer.

Syntax

```
_round ( numeric_exp, integer_exp )
```

Example

```
_round ( 1220.42369, 2 )
```

Result: 1220.42

abs

Returns the absolute value of numeric_exp. The sign of negative values is changed to positive.

Syntax

```
abs ( numeric_exp )
```

Example 1

```
abs ( 15 )
```

Result: 15

Example 2

```
abs ( -15 )
```

Result: 15

ancestor

Returns the ancestor of the specified member at either the specified (named) level or the specified number of levels above the member. Note: The result is not guaranteed to be consistent when there is more than one such ancestor.

Syntax

```
ancestor ( member, level | integer )
```

ancestors

Returns all the ancestors of a member at a specified level, or distance above the member. (Most data sources support only one ancestor at a specified level, but some support more than one. Hence the result is a member set.)

Syntax

```
ancestors ( member , level | index )
```

bottomCount

This function sorts a set according to the value of "numeric_exp" evaluated at each of the members of "set_exp", and returns the bottom "index_exp" members.

Syntax

```
bottomCount ( set_exp , index_exp , numeric_exp )
```

bottomPercent

This function is similar to bottomSum, but the threshold is "numeric_exp1" percent of the total.

Syntax

```
bottomPercent ( set_exp , numeric_exp1 , numeric_exp2 )
```

bottomSum

This function sorts on "numeric_exp2", evaluated at the corresponding member of "set_exp", and picks up the bottommost elements whose cumulative total is at least numeric_exp1.

Syntax

```
bottomSum ( set_exp , numeric_exp1 , numeric_exp2 )
```

caption

Returns the caption values of the specified argument.

Syntax

```
caption ( level | member | set_exp )
```

cast

Converts an 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.

Syntax

```
cast ( expression, datatype_specification )
```

Example 1

```
cast ( '123' , integer )
```

Result: 123

Example 2

```
cast ( 12345 , VARCHAR ( 10 ) )
```

Result: a string containing 12345

Notes

- You can specify the following datatypes: CHARACTER, VARCHAR, CHAR, NUMERIC, DECIMAL, INTEGER, SMALLINT, REAL, FLOAT, DATE, TIME, TIMESTAMP, TIME WITH TIME ZONE, TIMESTAMP WITH TIME ZONE, and INTERVAL.
- 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.
- When you type cast to an INTERVAL type, you must specify one of the following interval qualifiers: 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. 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.

ceil

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceil ( numeric_exp )
```

ceiling

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceiling ( numeric_exp )
```

Example 1

```
ceiling ( 4.22 )
```

Result: 5

Example 2

```
ceiling ( -1.23 )
```

Result: -1

char_length

Returns the number of characters in string_exp.

Syntax

```
char_length ( string_exp )
```

Example

```
char_length ( 'Canada' )
```

Result: 6

character_length

Returns the number of characters in string_exp.

Syntax

```
character_length ( string_exp )
```

Example

```
character_length ( 'Canada' )
```

Result: 6

children

Returns the set of children of a specified member.

Syntax

```
children ( member )
```

closingPeriod

Returns the last sibling among the descendants of a member at a specified level. Typically used with a time dimension.

Syntax

```
closingPeriod ( level [, member ] )
```

coalesce

Returns the first non-null argument (or null if all arguments are null). The Coalesce function takes two or more arguments.

Syntax

```
coalesce ( exp_list )
```

completeTuple

Similar to "tuple", 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. The value of this cell can be obtained with the "value" function.

Syntax

```
completeTuple ( member { , member } )
```

cousin

Returns the child member of member2 with the same relative position as the member1 is under its parent.

Syntax

```
cousin ( member1 , member2 )
```

current_date

Returns a date value representing the current date of the computer that the database software runs on.

Syntax

```
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.

Syntax

```
current_time
```

Example

```
current_time
```

Result: 16:33:11+05:00

current_timestamp

Returns a datetime with time zone value, representing the current timestamp of the computer that runs the database software.

Syntax

```
current_timestamp
```

Example

```
current_timestamp
```

Result: 2003-03-03 16:40:15.535000+05:00

currentMember

Returns the current member of the hierarchy during an iteration. If the specified hierarchy is not present in the context in which the expression is being evaluated, its default member is assumed.

Syntax

```
currentMember ( hierarchy )
```

defaultMember

Returns the default member of a hierarchy.

Syntax

```
defaultMember ( hierarchy )
```

descendants

Returns the set of descendants of a set of members at a specified level(qualified name) or distance(integer 0..n) from the root. Duplicates will be removed from the set. Multiple options may be specified (separated by a space) to determine which members are to be returned.

Syntax

```
descendants ( set_expr , level | distance [ , { self | before |
beforewithmember | after } ] )
```

Example

```
descendants([national].[Line].[Line].[Line1]->:[PC].[Line
(Root)].[Dishwashers], 2, SELF AFTER)
```

Result: `Result: AcmeWash MR AcmeWash AcmeWash HE`

Notes

- 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.

emptySet

Returns an empty member set for the specified hierarchy.

Syntax

```
emptySet ( hierarchy )
```

except

Returns the members of "set_exp1" that are not also in "set_exp2". Duplicates are retained only if the optional keyword ALL is supplied as the third argument.

Syntax

```
except ( set_exp1 , set_exp2 [,ALL] )
```

exp

Returns e raised to the power of numeric_exp. The constant e is the base of the natural logarithm. See also log.

Syntax

```
exp ( numeric_exp )
```

Example

```
exp ( 2 )
```

Result: 7.389056

extract

Returns an integer representing the value of datepart (year, month, day, hour, minute, second) in datetime_exp.

Syntax

```
extract ( datepart , datetime_exp )
```

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

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_exp" is true.

Syntax

```
filter ( set_exp , boolean_exp )
```

firstChild

Returns the first child of a member.

Syntax

```
firstChild ( member)
```

firstSibling

Returns the first child of the parent of a member.

Syntax

```
firstSibling ( member )
```

floor

Returns the largest integer less than or equal to numeric_exp.

Syntax

```
floor ( numeric_exp )
```

Example 1

```
floor ( 3.22 )
```

Result: 3

Example 2

```
floor ( -1.23 )
```

Result: -2

generate

This function evaluates "set_exp2" for each member of "set_exp1" and joins the resulting sets by union. If ALL is specified, duplicates in the result are retained.

Syntax

```
generate ( set_exp1 , set_exp2 [ , ALL ] )
```

head

Returns the first "index_exp" elements of "set_exp". The default for "index_exp" is 1.

Syntax

```
head ( set_exp [ , index_exp ] )
```

hierarchize

This function orders the members of a set in a hierarchy. Members in a level are sorted in their natural order, which is the default ordering of the members along a dimension when no other sort conditions are specified.

Syntax

```
hierarchize ( set_exp )
```

hierarchy

Returns the hierarchy that contains the specified level, member or member set.

Syntax

```
hierarchy ( level | member | set_exp )
```

intersect

Returns the intersection of two input sets. The result retains duplicates only when the optional keyword ALL is supplied as the third argument.

Syntax

```
intersect ( set_exp1 , set_exp2 [ , ALL ] )
```

item

Returns a member from a specified location within a set. The index into the set is zero based

Syntax

```
item ( set_exp , index )
```

lag

Returns the sibling member that is a specified number of positions prior to a specified member.

Syntax

```
lag ( member , index_exp )
```

lastChild

Returns the last child of a specified member.

Syntax

```
lastChild ( member )
```

lastPeriods

Returns the set of members from the same level that ends with the specified member. The number of members returned is the absolute value of "integer_exp". If "integer_exp" is negative, members following and including the specified member are returned. Typically used with a time dimension.

Syntax

```
lastPeriods ( integer_exp , member )
```

lastSibling

Returns the last child of the parent of a specified member.

Syntax

```
lastSibling ( member )
```

lead

Returns the sibling member that is a specified number of positions following a specified member.

Syntax

```
lead ( member , index_exp )
```

level

Returns the level of a member.

Syntax

```
level ( member )
```

levels

Returns the level in the hierarchy whose distance from the root is specified by "index".

Syntax

```
levels ( hierarchy , index )
```

ln

Returns the natural logarithm of the numeric_exp.

Syntax

```
ln ( numeric_exp )
```

Example

```
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 16:40:15.535000

lower

Returns string_exp with all uppercase characters shifted to lowercase.

Syntax

```
lower ( string_exp )
```

Example

```
lower ( 'ABCDEF' )
```

Result: 'abcdef'

member

Defines a member based on the specified expression in the specified hierarchy. "string1" is used to identify the member created by this function it must be unique in the query, and must be different from any other member in the same hierarchy. "string2" is used as the caption of the member; if it is absent, the caption is empty. If the hierarchy is omitted, the measure dimension is assumed. 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, otherwise the results are not predictable. The only exception to this is where the calculation involves only members of the same hierarchy as the siblings. In that case the calculation is assumed to belong to that hierarchy.

Syntax

```
member ( value_exp [ , string1 [ , string2 [ , hierarchy ] ] ] )
```

members

Returns the set of members in a 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 )
```

mod

Returns the remainder (modulus) of integer_exp1 divided by integer_exp2. The integer_exp2 must not be zero or an exception condition is raised.

Syntax

```
mod ( integer_exp1, integer_exp2 )
```

Example

```
mod ( 20 , 3 )
```

Result: 2

nestedSet

Returns the set of members of set_expr2 evaluated in the context of the current member of set_expr1.

Syntax

```
nestedSet ( set_expr1 , set_expr2 )
```

nextMember

Returns the next member in the level to which the specified member exists.

Syntax

```
nextMember ( member )
```

octet_length

Returns the number of bytes in string_exp.

Syntax

```
octet_length ( string_exp )
```

Example 1

```
octet_length ( 'ABCDEF' )
```

Result: 6

Example 2

```
octet_length ( '' )
```

Result: 0

openingPeriod

Returns the first sibling member among the descendants of a member at a specified level. Typically used with a time dimension.

Syntax

```
openingPeriod ( level [ , member ] )
```

order

Arranges members of a specified set, as determined from the set of values created by evaluating "value_exp" for each value of the set, and modified by the third parameter. There are two varieties of order: hierarchized (ASC or DESC) and non-hierarchized (BASC or BDESC, where B stands for "break hierarchy"). The hierarchized ordering first arranges members according to their position in the hierarchy. Then it orders the children of each member according to "value_exp". The non-hierarchized ordering arranges members in the set without regard to the hierarchy. In the absence of an explicit specification, ASC is the default.

Syntax

```
order ( set_exp , value_exp [ , ASC | DESC | BASC | BDESC ] )
```

ordinal

Returns the zero-based ordinal value (distance from the root level) of the specified level.

Syntax

```
ordinal ( level )
```

parallelPeriod

Returns a member from a different period in the same relative position as a specified 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" (call it "ancestor"); then it takes the sibling of "ancestor" that is offset (follows) by "int exp" positions, and returns the descendants of that sibling in the same relative position as the specified member as under "ancestor".

Syntax

```
parallelPeriod ( level , int_exp  , member )
```

parent

Returns the member that is the parent of the specified member.

Syntax

```
parent ( member )
```

periodsToDate

Returns a set of sibling members from the same level as a given member, as constrained by a specified 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.

Syntax

```
periodsToDate ( level , member )
```

position

Returns integer value representing the position of the first string_exp in the second string_exp or 0 when the first string_exp is not found.

Syntax

```
position ( string_exp , string_exp )
```

Example 1

```
position ( 'C' , 'ABCDEF' )
```

Result: 3

Example 2

```
position ( 'H' , 'ABCDEF' )
```

Result: 0

power

Returns numeric_exp1 raised to the power numeric_exp2. If numeric_exp1 is negative then numeric_exp2 must result in an integer value.

Syntax

```
power ( numeric_exp1, numeric_exp2 )
```

Example

```
power ( 3 , 2 )
```

Result: 9

prevMember

Returns the member that immediately precedes the specified member in the same level.

Syntax

```
prevMember ( member )
```

roleValue

Returns the value of the attribute that is associated with the role whose name is specified by "string" within the specified context. The second argument is optional only in a number of limited circumstances, where it can be derived from other 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 modelled relational data sources, assignment of roles is the modeller's responsibility.) Intrinsic roles that are defined for members of all data source types include: '_businessKey', '_memberCaption', '_memberDescription', '_memberUniqueName'.

Syntax

```
roleValue ( string [ , member | set_exp ] )
```

Example

```
roleValue ( '_memberCaption', [Sales].[Product].[Product].[Product
line]->[all].[1] )
```

Result: Camping Equipment

rootMembers

Returns the root members of a hierarchy.

Syntax

```
rootMembers ( hierarchy )
```

set

Returns the list of members defined in the expression. The members must belong to the same hierarchy.

Syntax

```
set ( member { , member } )
```

siblings

Returns the children of the parent of the specified member.

Syntax

```
siblings ( member )
```

sqrt

Returns the square root of numeric_exp. numeric_exp must be non-negative.

Syntax

```
sqrt ( numeric_exp )
```

Example

```
sqrt ( 9 )
```

Result: 3

subset

Returns a subset of members from a specified set starting "index_exp1" from the beginning. If the count "index_exp2" is specified, that many members (if available) are returned. Otherwise, all remaining members are returned.

Syntax

```
subset ( set_exp, index_exp1 [ , index_exp2 ] )
```

substring

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters or to the end of string_exp if integer_exp2 is omitted. The first character in string_exp is at position 1.

Syntax

```
substring ( string_exp , integer_exp1 [ , integer_exp2 ] )
```

Example

```
substring ( 'abdefg', 3, 2)
```

Result: 'de'

tail

Returns the last "index_exp" elements of "set exp". The default for "index_exp" is 1.

Syntax

```
tail ( set_exp [ , index_exp ] )
```

topCount

This function sorts a set according to the values of "numeric_exp" evaluated at each of the members of "set_exp", and returns the top "index_exp" members.

Syntax

```
topCount ( set_exp , index_exp , numeric_exp )
```

topPercent

This function is similar to topSum, but the threshold is "numeric_exp1" percent of the total.

Syntax

```
topPercent ( set_exp , numeric_exp1, numeric_exp2 )
```

topSum

This function sorts on "numeric_exp2", evaluated at the corresponding members of "set_exp", and picks up the topmost elements whose cumulative total is at least "numeric_exp1".

Syntax

```
topSum ( set_exp , numeric_exp1 , numeric_exp2 )
```

trim

Returns a string_exp trimmed of leading and\or trailing blanks or trimmed of a certain character specified in match_character_exp. BOTH is implicit when first argument is not stated and blank is implicit when second argument is not stated.

Syntax

```
trim ( [ [ TRAILING | LEADING | BOTH ] [ match_character_exp ] , ] string_exp
)
```

Example 1

```
trim ( TRAILING 'A' , 'ABCDEFA' )
```

Result: 'ABCDEF'

Example 2

```
trim ( BOTH ' ABCDEF ' )
```

Result: 'ABCDEF'

tuple

Identifies a cell location (intersection) based on the specified members, each of which must be from a different dimension. Implicitly includes the current member from all dimensions 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 } )
```

union

This function returns the union of 2 sets "set_exp1" and "set_exp2". The result retains duplicates only when the optional keyword ALL is supplied as the third argument.

Syntax

```
union ( set_exp1 , set_exp2 [ , ALL ] )
```

unique

Removes all duplicates from the specified set. The remaining members retain their original order.

Syntax

```
unique ( set_expr )
```

upper

Returns string_exp with all lowercase characters shifted to uppercase.

Syntax

```
upper ( string_exp )
```

Example

```
upper ( 'abcdef' )
```

Result: 'ABCDEF'

value

Returns the value of the cell identified by a tuple. Note that the default member of the Measures dimension is the Default Measure

Syntax

```
value ( tuple )
```

DB2

ascii

Returns the ASCII code value of the leftmost character of the argument as an integer.

Syntax

```
ascii ( string_exp )
```

ceiling

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceiling ( numeric_exp )
```

char

Returns a string representation of a date/time value or a decimal number.

Syntax

```
char ( exp )
```

chr

Returns the character that has the ASCII code value specified by integer_exp. integer_exp should be between 0 and 255.

Syntax

```
chr ( integer_exp )
```

concat

Returns a string that is the result of concatenating string_exp1 with string_exp2.

Syntax

```
concat ( string_exp1, string_exp2 )
```

date

Returns a date from a single input value. exp can be a string or integer representation of a date.

Syntax

```
date ( exp )
```

day

Returns the day of the month (1-31) from date_exp. date_exp can be a date value or a string representation of a date.

Syntax

```
day ( date_exp )
```

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_exp. date_exp can be a date value or a string representation of a date.

Syntax

```
dayname ( date_exp )
```

dayofweek

Returns the day of the week in date_exp as an integer in the range 1 to 7, where 1 represents Sunday. date_exp can be a date value or a string representation of a date.

Syntax

```
dayofweek ( date_exp )
```

dayofweek_iso

Returns the day of the week in date_exp as an integer in the range 1 to 7, where 1 represents Monday. date_exp can be a date value or a string representation of a date.

Syntax

```
dayofweek_iso ( date_exp )
```

dayofyear

Returns the day of the year in date_exp as an integer in the range 1 to 366. date_exp can be a date value or a string representation of a date.

Syntax

```
dayofyear ( date_exp )
```

days

Returns an integer representation of a date. exp can be a date value or a string representation of a date.

Syntax

```
days ( exp )
```

dec

Returns decimal representation of string_exp1 with precision numeric_exp1, scale numeric_exp2 and decimal character string_exp2. String_exp1 must be formatted as a SQL Integer or Decimal constant.

Syntax

```
dec ( string_exp1 [ , numeric_exp1 [ , numeric_exp2 [ , string_exp2 ] ] ] )
```

decimal

Returns decimal representation of string_exp1 with precision numeric_exp1, scale numeric_exp2 and decimal character string_exp2. String_exp1 must be formatted as a SQL Integer or Decimal constant.

Syntax

```
decimal ( string_exp1 [ , numeric_exp1 [ , numeric_exp2 [ , string_exp2 ] ] ]
)
```

difference

Returns an integer value representing the difference between the values returned by the data source_specific soundex function for string_exp1 and string_exp2. 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_exp1, string_exp2 )
```

digits

Returns the character string representation of a non-floating point number.

Syntax

```
digits ( numeric_exp )
```

double

Returns the floating-point representation of an expression. 'exp' can be either a numeric or string expression.

Syntax

```
double ( exp )
```

event_mon_state

Returns the operational state of a particular state monitor.

Syntax

```
event_mon_state ( string_exp )
```

float

Returns the floating-point representation of a number.

Syntax

```
float ( numeric_exp )
```

hex

Returns the hexadecimal representation of a value.

Syntax

```
hex ( exp )
```

hour

Returns the hour (an integer from 0, which is midnight, to 23, which is 11:00 pm) from time_exp. time_exp can be a time value or a string representation of a time.

Syntax

```
hour ( time_exp )
```

insert

Returns a string where length (integer_exp2) characters have been deleted from string_exp1 beginning at start (integer_exp1) and where string_exp2 has been inserted into string_exp1 at start. The first character in a string is at position 1.

Syntax

```
insert ( string_exp1, integer_exp1, integer_exp2, string_exp2 )
```

integer

Returns the integer representation of an expression. exp can be a numeric value or a string representation of a number.

Syntax

```
integer ( exp )
```

int

Returns the integer representation of an expression. exp can be a numeric value or a string representation of a number.

Syntax

```
int ( exp )
```

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 exp. exp can be a date value or a string representation of a date.

Syntax

```
julian_day ( exp )
```

lcase

Returns string_exp with all uppercase characters shifted to lowercase.

Syntax

```
lcase ( string_exp )
```

left

Returns the leftmost integer_exp characters of string_exp.

Syntax

```
left ( string_exp, integer_exp )
```

length

Returns the length of the operand in bytes (except for double byte string types which return the length in characters).

Syntax

```
length ( exp )
```

locate

Returns the starting position of the first occurrence of string_exp1 within string_exp2. The search starts at position start (integer_exp) of string_exp2. The first character in a string is at position 1. If string_exp1 is not found then zero is returned.

Syntax

```
locate ( string_exp1, string_exp2 [ , integer_exp ] )
```

long_varchar

Returns a long string.

Syntax

```
long_varchar ( string_exp )
```

ltrim

Returns string_exp with leading spaces removed.

Syntax

```
ltrim ( string_exp )
```

microsecond

Returns the microsecond (time-unit) part of a value. exp can be a timestamp or a string representation of a timestamp.

Syntax

```
microsecond ( exp )
```

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. exp can be a time value, a timestamp or a string representation of a time.

Syntax

```
midnight_seconds ( exp )
```

minute

Returns the minute (an integer from 0-59) from time_exp. time_exp can be a time value, a timestamp or a string representation of a time.

Syntax

```
minute ( time_exp )
```

month

Returns the month (an integer from 1-12) from date_exp.

Syntax

```
month ( date_exp )
```

monthname

Returns a character string containing the data source_specific name of the month (for example, January through December or Jan. through Dec. for a data source that uses English, or Januar through Dezember for a data source that uses German) for the month portion of date_exp.

Syntax

```
monthname ( date_exp )
```

quarter

Returns the quarter in date_exp as a number in the range 1 to 4, where 1 represents January 1 through March 31.

Syntax

```
quarter ( date_exp )
```

radians

Returns the number of radians converted from numeric_exp degrees.

Syntax

```
radians ( numeric_exp )
```

repeat

Returns a string consisting of string_exp repeated integer_exp times.

Syntax

```
repeat ( string_exp, integer_exp )
```

replace

Replaces all occurrences of string_exp2 in string_exp1 with string_exp3.

Syntax

```
replace ( string_exp1, string_exp2, string_exp3 )
```

right

Returns the rightmost integer_exp characters of string_exp.

Syntax

```
right ( string_exp, integer_exp )
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point, e.g., round-near (125, -1) rounds to 130.

Syntax

```
round ( numeric_exp, integer_exp )
```

rtrim

Returns string_exp with trailing spaces removed.

Syntax

```
rtrim ( string_exp )
```

second

Returns the second (an integer from 0-59) from time_exp.

Syntax

```
second ( time_exp )
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign ( numeric_exp )
```

smallint

Returns the small integer representation of a number.

Syntax

```
smallint ( exp )
```

soundex

Returns a 4 character string code obtained by systematically abbreviating words and names in string_exp according to phonetics. Can be used to determine if two strings sound the same, e.g., does sound-of ('SMITH') = sound-of ('SMYTH').

Syntax

```
soundex ( string_exp )
```

space

Returns a string consisting of integer_exp spaces.

Syntax

```
space ( integer_exp )
```

substr

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters. The first character in string_exp is at position 1.

Syntax

```
substr ( string_exp, integer_exp1 [ , integer_exp2 ] )
```

table_name

Returns an unqualified name of a table or view based on the object name in string_exp1 and the schema name given in string_exp2. It is used to resolve aliases.

Syntax

```
table_name ( string_exp1 [ , string_exp2 ] )
```

table_schema

Returns the schema name portion of the two part table or view name based on the object name in string_exp1 and the schema name in string_exp2. It is used to resolve aliases.

Syntax

```
table_schema ( string_exp1 [ , string_exp2 ] )
```

time

Returns a time from a value.

Syntax

```
time ( exp )
```

timestamp

Returns a timestamp from a value or a pair of values. exp1 must represent a date value, and exp2 must represent a time value.

Syntax

```
timestamp ( exp1 [ , exp2 ] )
```

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.nnnnnn). If the exp 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 ( exp )
```

timestampdiff

Returns an estimated number of intervals of type exp1 based on the difference between two timestamps. Exp2 is the result of subtracting two timestamp types and converting the result to CHAR. Valid values of exp1 are: 1 Fractions of a second; 2 Seconds; 4 Minutes; 8 Hours; 16 Days; 32 Weeks; 64 Months; 128 Quarters; 256 Years

Syntax

```
timestampdiff ( exp1, exp2 )
```

to_char

Returns the string representation of a timestamp with the format of string_exp.

Syntax

```
to_char ( timestamp_exp , string_exp )
```

translate

Returns string_exp1 in which characters from string_exp3 are translated to the equivalent characters in string_exp2. string_exp4 is a single character that is used to pad string_exp2 if it is shorter than string_exp3. If only string_exp1 is present, then this function translates it to uppercase characters.

Syntax

```
translate ( string_exp1 [ , string_exp2, string_exp3 [ , string_exp4 ] ] )
```

trunc

Returns numeric_exp1 truncated to numeric_exp2 places RIGHT of the decimal point. If numeric_exp2 is negative, numeric_exp1 is truncated to the absolute value of numeric_exp2 places to the LEFT of the decimal point.

Syntax

```
trunc ( numeric_exp1, numeric_exp2 )
```

truncate

Returns numeric_exp1 truncated to numeric_exp2 places RIGHT of the decimal point. If numeric_exp2 is negative, numeric_exp1 is truncated to the absolute value of numeric_exp2 places to the LEFT of the decimal point.

Syntax

```
truncate ( numeric_exp1, numeric_exp2 )
```

ucase

Returns string_exp with all lowercase characters shifted to uppercase.

Syntax

```
ucase ( string_exp )
```

value

Returns the first non-null argument (or null if all arguments are null). The Value function takes two or more arguments.

Syntax

```
value ( exp_list )
```

varchar

Returns a VARCHAR representation of exp, with length numeric_exp.

Syntax

```
varchar ( exp [ , numeric_exp ] )
```

week

Returns the week of the year in date_exp as an integer value in the range 1 to 53.

Syntax

```
week ( date_exp )
```

year

Returns the year from date_exp.

Syntax

```
year ( date_exp )
```

DB2 Cast

cast_char

Returns the first numeric_exp characters of the value of exp cast as a string. The whole string is returned when the second argument is not specified.

Syntax

```
cast_char ( exp [ , numeric_exp ] )
```

cast_date

Returns the value of the expression cast as a date.

Syntax

```
cast_date ( exp )
```

cast_decimal

Returns the value of exp cast as a decimal with the precision of numeric_exp1 and scale of numeric_exp2.

Syntax

```
cast_decimal ( exp [ , numeric_exp1, numeric_exp2 ] )
```

cast_double_precision

Returns the value of the expression cast as a double.

Syntax

```
cast_double_precision ( exp )
```

cast_float

Returns the value of the expression cast as a float.

Syntax

```
cast_float ( exp )
```

cast_integer

Returns the value of the expression cast as a integer.

Syntax

```
cast_integer ( exp )
```

cast_longvarchar

Returns the value of the expression cast as a longvarchar.

Syntax

```
cast_longvarchar ( string_exp )
```

cast_smallint

Returns the value of the expression cast as a smallint.

Syntax

```
cast_smallint ( exp )
```

cast_time

Returns the value of the expression cast as a time value.

Syntax

```
cast_time ( string_exp )
```

cast_timestamp

Returns the value of the expression cast as a datetime.

Syntax

```
cast_timestamp ( exp )
```

cast_varchar

Returns the value of the expression cast as a varchar with length.

Syntax

```
cast_varchar ( exp, integer_exp )
```

DB2 Math

log

Returns the natural logarithm of numeric_exp.

Syntax

```
log ( numeric_exp )
```

log10

Returns the base ten logarithm of numeric_exp.

Syntax

```
log10 ( numeric_exp )
```

rand

Generates a random number using integer_exp as a seed value.

Syntax

```
rand ( integer_exp )
```

DB2 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 )
```

atanh

Returns the hyperbolic arctangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
atanh (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

```
cos ( numeric_exp )
```

cosh

Returns the hyperbolic cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cosh ( numeric_exp )
```

cot

Returns the cotangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cot ( numeric_exp )
```

degrees

Returns numeric_exp radians converted to degrees.

Syntax

```
degrees ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

sinh

Returns the hyperbolic sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sinh ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

tanh

Returns the hyperbolic tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tanh ( numeric_exp )
```

Informix

cardinality

Returns the number of elements in a collection column (SET, MULTISET, LIST).

Syntax

```
cardinality ( string_exp )
```

char_length

Returns the number of logical characters (which can be distinct from the number of bytes in some East Asian locales) in string_exp.

Syntax

```
char_length ( string_exp )
```

concat

Returns a string that is the result of concatenating string_exp1 to string_exp2.

Syntax

```
concat ( string_exp1, string_exp2 )
```

date

Returns the date value of either string_exp or date_exp or int_exp

Syntax

```
date ( string_exp | date_exp | int_exp )
```

day

Returns an integer that represents the day of the month.

Syntax

```
day ( date_exp )
```

extend

The extend function 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 (that is, 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 field (that is, less significant) 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 the missing HOUR to FRACTION fields are filled in with 0.

Syntax

```
extend ( date_exp , ' { ' YEAR TO SECOND ' } ' )
```

Example

```
EXTEND (some_date_column , {YEAR TO SECOND} )
```

hex

Returns the hexadecimal encoding of an integer integer_exp.

Syntax

```
hex ( integer_exp )
```

initcap

Returns string_exp, with the first letter of each word in uppercase, all other letters in lowercase. With this function, a word begins after any character other than a letter. Thus, in addition to a blank space, symbols such as commas, periods, colons, and so on, introduce a new word..

Syntax

```
initcap ( string_exp )
```

length

Returns the number of bytes in string_exp, which is not including any trailing blank spaces. For BYTE or TEXT string_exp, LENGTH returns the full number of bytes, including any trailing blank spaces.

Syntax

```
length ( string_exp )
```

lpad

Returns a copy of string_exp1 that is left-padded (string_exp2) to the total number of characters specified by integer_exp. The sequence of string_exp2 occurs as many times as necessary to make the return string the length specified by integer_exp.

Syntax

```
lpad ( string_exp1, integer_exp, string_exp2 )
```

mdy

Returns a type DATE value with three expressions that evaluate to integers that represent the month(integer_exp1), day(integer_exp2), and year(integer_exp3).

Syntax

```
mdy ( integer_exp1, integer_exp2, integer_exp3 )
```

month

Returns an integer corresponding to the month portion of date_exp.

Syntax

```
month ( date_exp )
```

nvl

Returns the value of exp1 if exp1 is not NULL. If exp1 is NULL, then the value of exp2 is returned.

Syntax

```
nvl ( exp1, exp2 )
```

octet_length

Returns the number of bytes in string_exp, including any trailing spaces.

Syntax

```
octet_length ( string_exp )
```

replace

Returns a copy of string_exp1 in which every occurrence of string_exp2 is replaced by string_exp3. If you omit the string_exp3 option, every occurrence of string_exp2 is omitted from the return string.

Syntax

```
replace ( string_exp1, string_exp2 [ , string_exp3 ] )
```

round

Returns the rounded value of an numeric_exp. If you omit the integer_exp, the value is rounded to zero digits or to the units place. The digit range of 32 (+ and -) refers to the entire decimal value.

Syntax

```
round ( numeric_exp [ , integer_exp ] )
```

rpad

Returns a copy of string_exp1 that is right-padded (string_exp2) to the total number of characters specified by integer_exp. The sequence of string_exp2 occurs as many times as necessary to make the return string the length specified by integer_exp.

Syntax

```
rpad ( string_exp1, integer_exp, string_exp2 )
```

substr

Returns the substring of string_exp that starts at position integer_exp1. The first character in string_exp is at position 1. integer_exp2 can be used to select fewer characters, by default it selects character to the end of the string.

Syntax

```
substr ( string_exp, integer_exp1 [ , integer_exp2 ] )
```

to_char

Returns the character string of date_exp with the specified string_exp formatting. You can use this function only with built-in data types.

Syntax

```
to_char ( date_exp, string_exp )
```

to_date

Returns the string_exp1 as a date according to the date format you specify in the string_exp2 parameter. If string_exp1 is NULL, then a NULL value is returned.

Syntax

```
to_date ( string_exp1, string_exp2 )
```

trunc

Returns the truncated value of a numeric_exp. If you omit integer_exp, the value 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_exp [ , integer_exp ] )
```

weekday

Returns an integer that represents the day of the week; zero (0) represents Sunday, one (1) represents Monday, and so on.

Syntax

```
weekday ( date_exp )
```

year

Returns a four-digit integer that represents the year.

Syntax

```
year ( date_exp )
```

Informix Math

log10

Returns the log of a numeric_exp to base 10.

Syntax

```
log10 ( numeric_exp )
```

logn

Returns the natural logarithm of a numeric_exp.

Syntax

```
logn ( numeric_exp )
```

root

Returns the root value of a numeric_exp. Requires at least one numeric argument (the radians argument). If only the numeric_exp1 is supplied, the value 2 is used as a default value for numeric_exp2; 0 cannot be used as the value of numeric_exp2.

Syntax

```
root ( numeric_exp1[ , numeric_exp2 ] )
```

Informix 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_exp1.

Syntax

```
atan2 ( numeric_exp1, numeric_exp2 )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

MS Access

ascii

Returns a number representing the ascii code value of the leftmost character of string_exp.

Syntax

```
ascii(string_exp)
```

ceiling

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceiling(numeric_exp)
```

chr

Returns the character that has the ASCII code value specified by integer_exp. integer_exp should be between 0 and 255.

Syntax

```
chr(integer_exp)
```

concat

Returns a string that is the result of concatenating string_exp1 to string_exp2.

Syntax

```
concat(string_exp1, string_exp2)
```

curdate

Returns a date value representing the current date of the computer that the database software runs on.

Syntax

```
curdate()
```

curtime

Returns a time value representing the current time of the computer that the database software runs on.

Syntax

```
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 a data source that uses English, or Sonntag through Samstag for a data source that uses German) for the day portion of date_exp.

Syntax

```
dayname(date_exp)
```

dayofmonth

Returns the day of the month (1-31) from date_exp. Returns the days field (a signed integer) from interval_exp.

Syntax

```
dayofmonth(date_exp|interval_exp)
```

dayofweek

Returns the day of the week in date_exp as an integer in the range 1 to 7, where 1 represents Monday.

Syntax

```
dayofweek(date_exp)
```

dayofyear

Returns the day of the year in date_exp as an integer in the range 1 to 366.

Syntax

```
dayofyear(date_exp)
```

hour

Returns the hour (an integer from 0, which is midnight, to 23, which is 11:00 pm) from time_exp.

Syntax

```
hour(time_exp)
```

instr

Searches string_exp1 for the first occurrence of string_exp2. Returns an integer specifying the position of string_exp2. The optional argument integer_exp1 sets the starting position for the search. If omitted, the search begins at the first character position of string_exp1. The optional argument integer_exp2 specifies the type of string comparison. integer_exp1 is required if integer_exp2 is specified.

Syntax

```
instr ( [ integer_exp1 , ] string_exp1, string_exp2 [ , integer_exp2 ] )
```

lcase

Returns string_exp with all uppercase characters shifted to lowercase.

Syntax

```
lcase(string_exp)
```

left

Returns the leftmost integer_exp characters of string_exp.

Syntax

```
left(string_exp, integer_exp)
```

length

Returns the number of characters in string_exp, excluding trailing blanks and the string termination character.

Syntax

```
length(string_exp)
```

locate

Returns the starting position of the first occurrence of string_exp1 within string_exp2. The search starts at position start (integer_exp) of string_exp2. The first character in a string is at position 1. If string_exp1 is not found then zero is returned.

Syntax

```
locate(string_exp1, string_exp2 [ , integer_exp ] )
```

ltrim

Returns string_exp with leading spaces removed.

Syntax

```
ltrim(string_exp)
```

minute

Returns the minute (an integer from 0-59) from time_exp.

Syntax

```
minute(time_exp)
```

month

Returns the month (an integer from 1-12) from date_exp.

Syntax

```
month(date_exp)
```

monthname

Returns a character string containing the data source_specific name of the month (for example, January through December or Jan. through Dec. for a data source that uses English, or Januar through Dezember for a data source that uses German) for the month portion of date_exp.

Syntax

```
monthname(date_exp)
```

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_exp1 in string_exp2. The first character in a string is at position 1.

Syntax

```
position(string_exp1, string_exp2)
```

quarter

Returns the quarter in date_exp as a number in the range 1 to 4, where 1 represents January 1 through March 31.

Syntax

```
quarter(date_exp)
```

right

Returns the rightmost integer_exp characters of string_exp.

Syntax

```
right(string_exp, integer_exp)
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point.

Syntax

```
round(numeric_exp, integer_exp)
```

rtrim

Returns string_exp with trailing spaces removed.

Syntax

```
rtrim(string_exp)
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign(numeric_exp)
```

space

Returns a string consisting of integer_exp spaces.

Syntax

```
space(integer_exp)
```

substr

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters. The first character in string_exp is at position 1.

Syntax

```
substr(string_exp, integer_exp1, integer_exp2)
```

substring

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters. The first character in string_exp is at position 1.

Syntax

```
substring(string_exp, integer_exp1, integer_exp2)
```

truncate

Returns string_exp with trailing spaces removed.

Syntax

```
truncate(string_exp)
```

ucase

Returns string_exp with all lowercase characters shifted to uppercase.

Syntax

```
ucase(string_exp)
```

week

Returns the week of the year in date_exp as an integer value in the range 1 to 53.

Syntax

```
week(date_exp)
```

year

Returns the year from date_exp.

Syntax

```
year(date_exp)
```

MS Access Cast

cast_decimal

Returns the value of the expression cast as a decimal.

Syntax

```
cast_decimal(exp)
```

cast_float

Returns the value of the expression cast as a float.

Syntax

```
cast_float(exp)
```

cast_integer

Returns the value of the expression cast as a integer.

Syntax

```
cast_integer(exp)
```

cast_numeric

Returns the value of string_exp cast as a numeric value.

Syntax

```
cast_numeric(string_exp)
```

cast_real

Returns the value of the expression cast as a real.

Syntax

```
cast_real(exp)
```

cast_smallint

Returns the value of the expression cast as a smallint.

Syntax

```
cast_smallint(exp)
```

cast_varchar

Returns the value of the expression cast as a varchar.

Syntax

```
cast_varchar(exp)
```

MS Access Math

log

Returns the natural logarithm of numeric_exp.

Syntax

```
log(numeric_exp)
```

rand

Generates a random number using integer_exp as a seed value.

Syntax

```
rand(integer_exp)
```

MS Access Trigonometry

atan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
atan(numeric_exp)
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos(numeric_exp)
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin(numeric_exp)
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan(numeric_exp)
```

Oracle

add_months

Returns the datetime resulting from adding integer_exp months to date_exp.

Syntax

```
add_months ( date_exp, integer_exp )
```

ascii

Returns a number representing the ascii code value of the leftmost character of string_exp, e.g. ascii('A') is 65.

Syntax

```
ascii ( string_exp )
```

ceil

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceil ( numeric_exp )
```

char_length

Returns the number of characters in string_exp.

Syntax

```
char_length ( string_exp )
```

chr

Returns the character that has the ASCII code value specified by integer_exp. integer_exp should be between 0 and 255.

Syntax

```
chr ( integer_exp )
```

concat

Returns a string that is the result of concatenating string_exp1 to string_exp2.

Syntax

```
concat ( string_exp1, string_exp2 )
```

decode

DECODE compares expr to each search value one by one. If expr is equal to a search, then returns the corresponding result. If no match is found, then returns default. If default is omitted, then returns null.

Syntax

```
decode ( expr , search , result [, search , result]... [, default] )
```

dump

Returns internal representation of 'expr' with the format of numeric_exp1 starting from position numeric_exp2 for numeric_exp3.

Syntax

```
dump ( expr [ , numeric_exp1 [ , numeric_exp2 [ , numeric_exp3 ] ] ] )
```

greatest

Returns the greatest value in a list of expressions.

Syntax

```
greatest ( exp_list )
```

initcap

Returns string_exp, 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_exp )
```

instr

Searches string_exp1 from the integer_exp1 position for the (integer_exp2)th occurance of string_exp2. If integer_exp1 is negative then the search is backwards from the end of string_exp1. Returns an integer indicating the position of string_exp2.

Syntax

```
instr ( string_exp1, string_exp2 [ , integer_exp1 [ , integer_exp2 ] ] )
```

instrb

Searches string_exp1 from the integer_exp1 position for the (integer_exp2)th occurance of string_exp2. If integer_exp1 is negative then the search is backwards from the end of string_exp1. The result returned indicates the position (byte number) where search was found.

Syntax

```
instrb ( string_exp1, string_exp2 [ , integer_exp1 [ , integer_exp2 ] ] )
```

least

Returns the least value in a list of expressions.

Syntax

```
least ( exp_list )
```

length

Returns the number of characters in string_exp.

Syntax

```
length ( string_exp )
```

lengthb

Returns the number of bytes in string_exp.

Syntax

```
lengthb ( string_exp )
```

lpad

Returns string_exp1 padded to length integer_exp with occurrences of string_exp2. If string_exp1 is longer than integer_exp then returns the appropriate portion of string_exp1.

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 ] )
```

months_between

Returns the number of months from date_exp1 to date_exp2. If date_exp1 is later than date_exp2 then the result will be a positive number. The days and time portion of the difference are ignored, i.e. the months are not rounded, except if date_exp1 and date_exp2 are the last days of a month.

Syntax

```
months_between ( date_exp1, date_exp2 )
```

new_time

Returns the Datetime in timezone 'new_tz' for 'datetime' in 'old_tz' timezone. 'Old_tz' and 'new_tz' 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_exp, old_tz, new_tz )
```

next_day

Returns the datetime of the first weekday named by string_exp that is later than datetime_exp. The return value has the same hours, minutes, and seconds as datetime_exp.

Syntax

```
next_day ( datetime_exp, string_exp )
```

nls_initcap

Returns string_exp1 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. string_exp2 specifies the sorting sequence.

Syntax

```
nls_initcap ( string_exp1 [ , string_exp2 ] )
```

nls_lower

Returns string_exp1 with all letters in lowercase. string_exp2 specifies the sorting sequence.

Syntax

```
nls_lower ( string_exp1 [ , string_exp2 ] )
```

nls_upper

Returns string_exp1 with all letters in uppercase. string_exp2 specifies the sorting sequence.

Syntax

```
nls_upper ( string_exp1 [ , string_exp2 ] )
```

nvl

If exp is null (missing) returns constant. If exp is not null returns exp. Valid for numeric_exp, string_exp, date_exp, and time_exp.

Syntax

```
nvl ( exp, constant )
```

replace

Replaces all occurrences of string_exp2 in string_exp1 with string_exp3. If string_exp3 is not specified then it replaces all occurrences with null (ie: removes all occurances of string_exp2).

Syntax

```
replace ( string_exp1, string_exp2 [ , string_exp3 ] )
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point, e.g., round (125, -1) rounds to 130.

Syntax

```
round ( numeric_exp [ , integer_exp ] )
```

rpad

Returns string_exp1 right-padded to length integer_exp with occurrences of string_exp2. If string_exp1 is longer than integer_exp then returns the appropriate portion of string_exp1. 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('ABxXxyx', 'xy') returns 'ABxX'. If string_exp2 is not specified it removes th final space characters.

Syntax

```
rtrim ( string_exp1 [ , string_exp2 ] )
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign ( numeric_exp )
```

soundex

Returns a character string containing the phonetic representation of string_exp.

Syntax

```
soundex ( string_exp )
```

substr

Returns the substring of string_exp that starts at position integer_exp1. The first character in string_exp is at position 1. integer_exp2 can be used to select fewer characters, by default it selects character to the end of the string.

Syntax

```
substr ( string_exp, integer_exp1 [ , integer_exp2 ] )
```

substrb

Same as substr, except that the arguments are expressed in bytes (not characters).

Syntax

```
substrb ( string_exp, numeric_exp1 [ , numeric_exp2 ] )
```

{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 exp with the format of string_exp. exp can either be a date value or a numeric value.

Syntax

```
to_char ( exp [ , string_exp ] )
```

to_date

Converts string_exp1 to a datetime value as specified by the format string_exp2. string_exp3 specifies format elements such as language.

Syntax

```
to_date ( string_exp1 [ , string_exp2 [ , string_exp3 ] ] )
```

to_number

Converts string_exp1 to a numeric value as specified by the format string_exp2. string_exp3 specifies format elements such as currency information.

Syntax

```
to_number ( string_exp1, string_exp2, string_exp3 )
```

translate

Returns string_exp1, with all occurrences of each character in string_exp2 replaced by its corresponding character in string_exp3.

Syntax

```
translate ( string_exp1, string_exp2, string_exp3 )
```

trunc

Truncates the date_exp using the format specified by string_exp. For example, if string_exp is 'YEAR' then date_exp is truncated to the first day of the year.

Syntax

```
trunc ( date_exp, string_exp )
```

trunc

Truncates digits from numeric_exp1 using numeric_exp2 as the precision.

Syntax

```
trunc ( numeric_exp1, numeric_exp2 )
```

{user}

Returns the username of the current Oracle user.

Syntax

```
{ user }
```

vsize

Returns the number of bytes in the internal representation of 'exp'. 'exp' must be a string expression.

Syntax

```
vsize ( exp )
```

Oracle Math

log

Returns the logarithm of numeric_exp2 to the base numeric_exp1.

Syntax

```
log ( numeric_exp1, numeric_exp2 )
```

Oracle 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

```
cos ( numeric_exp )
```

cosh

Returns the hyperbolic cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cosh ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

sinh

Returns the hyperbolic sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sinh ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

tanh

Returns the hyperbolic tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tanh ( numeric_exp )
```

Red Brick

ceil

Returns the smallest integer greater than or equal to numeric_exp or string_exp. Note that string_exp must represent a valid numeric value.

Syntax

```
ceil ( numeric_exp | string_exp )
```

concat

This function concatenates character strings and returns the concatenated string of characters.

Syntax

```
concat ( string_exp1 , string_exp2 )
```

{current_user}

Returns the database username (authorization ID) of the current user.

Syntax

```
{ current_user }
```

date

This function creates a date value from a character string or a timestamp expression and returns a date data type. The expression can be either characters or timestamp.

Syntax

```
date ( expression )
```

dateadd

This function adds an interval to a datetime value and returns a result that is the same datetime data type as that of datetime_expression. The datepart refers to year, month, day, hour, minute, second. The interval must be an integer and datetime_exp can be date, time or timestamp.

Syntax

```
dateadd ( { datepart }, interval, datetime_exp )
```

datediff

This function finds the difference between two datetime expressions and returns an integer result in datepart units. The datepart refers to year, month, day, hour, minute, second. The datetime_exp can be date, time or timestamp.

Syntax

```
datediff ( { datepart }, datetime_exp, datetime_exp )
```

datename

This function extracts the specified datepart component and returns its value as a character string. The datepart refers to year, month, day, hour, minute, second. The datetime_exp can be date, time or timestamp.

Syntax

```
datename ( { datepart }, datetime_exp )
```

dec

This function converts a specified value to a decimal value and returns a 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

This function converts a specified value to a decimal value and returns a 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

This function compares and converts an expression to another value. If the expression matches target, it is replaced by the corresponding replacement; otherwise the expression is replaced by default or by NULL if no default is specified. The expressions can be any data type and all expressions must be the same data type.

Syntax

```
decode ( expression, target, replacement [,default] )
```

float

This function converts a specified value into a double-precision floating-point value.

Syntax

```
float ( numeric_exp )
```

ifnull

This function tests an expression for missing values and replaces each one with a specified value. If expression is NULL, this function returns substitute; otherwise it returns the value of the expression. The expressions can be any data type and all expressions must be the same data type.

Syntax

```
ifnull ( expression , substitute )
```

int

This function converts a specified numeric string into an integer value and returns an integer value. If the argument is null, this function returns NULL.

Syntax

```
int ( numeric_exp )
```

integer

This function converts a specified numeric string into an integer value and returns an integer value. If the argument is null, this function returns NULL.

Syntax

```
integer ( numeric_exp )
```

length

If the argument is not null, this function returns an integer result specifying the number of characters in the string; otherwise the result is NULL.

Syntax

```
length ( string_exp )
```

lengthb

If the argument is not null, this function returns an integer result specifying the number of bytes in the string. If the argument is null, the result is NULL.

Syntax

```
lengthb ( string_exp )
```

ltrim

If the argument is not null, this function removes leading blanks from the character string; otherwise the result is NULL.

Syntax

```
ltrim ( string_exp )
```

nullif

This function returns NULL if both expressions have the same value. If the expressions have different values, the value of the first expression is returned. The exp1 and exp2 can be any data type and must be the same data type.

Syntax

```
nullif ( exp1, exp2 )
```

positionb

If the first string_exp is located, this function returns an integer that is relative to the beginning byte position of the first string_exp in the second string_exp. If the first string_exp is not located, the result is 0. If the first string_exp is of zero length, the result is 1. If the first string_exp is null, an error message is returned. If the second string_exp is null, the result is 0.

Syntax

```
positionb ( string-exp, string_exp )
```

real

This function returns a real value. If the argument is null, this function returns NULL.

Syntax

```
real ( numeric_exp )
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point, e.g., round (125, -1) rounds to 130.

Syntax

```
round ( numeric_exp, integer_exp )
```

rtrim

If the argument is not null, this function removes trailing blanks from the character string; otherwise the result is NULL.

Syntax

```
rtrim ( string_exp )
```

sign

This function calculates the sign of the expression, and returns 1 for a positive value, –1 for a negative value, and 0 for zero.

Syntax

```
sign ( numeric_exp )
```

string

This function converts numeric or datetime values to character strings. The expression can be numeric or datetime.

Syntax

```
string ( expression [, length [, scale]] )
```

substr

If the first argument is not null, this function returns the substring that begins at position start and continues for length characters. If length is not specified, this function returns a substring from start to the end of string_exp.

Syntax

```
substr ( string_exp, start_integer, length_integer )
```

substrb

If the first argument is not null, this function returns the substring that begins at position start and continues for length bytes. If length is not specified, this function returns a substring from start to the end of string_exp.

Syntax

```
substrb ( string_exp, start_integer, length_integer )
```

time

This function creates a time value from a character string or a time-stamp data type expression.

Syntax

```
time ( expression )
```

timestamp

This function creates a time-stamp value from a character string.

Syntax

```
timestamp ( timestamp_exp )
```

timestamp

This function creates a time-stamp value from time and date values. If there are two arguments, the first must be a date expression and the second must be a time expression, separated by a comma (,). If either the date expression or the time expression is null, the resulting time-stamp expression is also null.

Syntax

```
timestamp ( date_exp, time_exp )
```

to_char

This function is a datetime scalar function that operates on a date, time, or timestamp data type and returns the character string specified by a given format.

Syntax

```
to_char ( source_date, format_str )
```

SQL Server

ascii

Returns a number representing the ascii code value of the leftmost character of string_exp, e.g. ascii('A') is 65.

Syntax

```
ascii(string_exp)
```

char

Returns the character that has the ASCII code value specified by integer_exp. integer_exp should be between 0 and 255. For example, char(65) has the value 'A'.

Syntax

```
char(integer_exp)
```

charindex

Searches string_exp2 for the first occurrence of string_exp1 and returns an integer. start_location is the character position to start searching for string_exp1 in string_exp2. If start_location is not given, is a negative number, or is zero, the search starts at the beginning of string_exp2

Syntax

```
charindex ( string_exp1, string_exp2 [ , start_location ]  )
```

{current_user}

Returns the name of the current user.

Syntax

```
{ current_user }
```

datalength

Returns the length in bytes of the string.

Syntax

```
datalength(string_exp)
```

dateadd

Returns the date resulting from adding integer_exp units indicated by datepart(e.g. day, month, year) to date_exp.

Syntax

```
dateadd({datepart}, integer_exp, date_exp)
```

datediff

Returns the number of units indicated by datepart(e.g. day, month, year) between date_exp1 and date_exp2.

Syntax

```
datediff({datepart}, date_exp1, date_exp2)
```

datename

Returns part of a datetime, smalldatetime, date or time value as an ASCII string. Note that the datepart argument 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_exp )
```

Example

```
datename ( {mm}, 2000-01-01 )
```

Result: January.

datepart

Returns part of a datetime, smalldatetime, date or time value (for example, the month) as an integer. Note that the datepart argument 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_exp )
```

Example

```
datepart ( {wk}, 2000-01-01 )
```

Result: 1 (first week of the year).

day

Returns the day portion of date_exp. Same as extract(day from date_exp).

Syntax

```
day(date_exp)
```

difference

Returns an integer value representing the difference between the values returned by the data source_specific soundex function for string_exp1 and string_exp2. 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_exp1, string_exp2)
```

getdate

Returns a datetime value representing the current date and time of the computer that the database software runs on.

Syntax

```
getdate()
```

left

Returns the leftmost integer_exp characters of string_exp.

Syntax

```
left(string_exp, integer_exp)
```

ltrim

Returns string_exp with leading spaces removed.

Syntax

```
ltrim(string_exp)
```

month

Returns the month portion of date_exp. Same as extract(month from date_exp).

Syntax

```
month(date_exp)
```

patindex

Returns an integer which represents the starting position if the first occurrence of string_exp1 is in the string_exp2; returns 0 if string-exp1 is not found. The % wildcard character must precede and follow pattern in string_exp1, except when searching for first or last characters

Syntax

```
patindex ( string_exp1, string_exp2 )
```

Example

```
patindex ( '%nos%', 'Cognos' )
```

Result: 4

replace

Replaces all occurrences of string_exp2 in string_exp1 with string_exp3.

Syntax

```
replace ( string_exp1 , string_exp2 , string_exp3 )
```

replicate

Returns a string consisting of string_exp repeated integer_exp times.

Syntax

```
replicate(string_exp, integer_exp)
```

reverse

Returns the reverse of the character expression.

Syntax

```
reverse ( string_exp )
```

right

Returns the rightmost integer_exp characters of string_exp.

Syntax

```
right(string_exp, integer_exp)
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point.

Syntax

```
round(numeric_exp,integer_exp)
```

rtrim

Returns string_exp with trailing spaces removed.

Syntax

```
rtrim(string_exp)
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign(numeric_exp)
```

soundex

Returns a four character string representing the sound of the words in string_exp.

Syntax

```
soundex(string_exp)
```

space

Returns a string consisting of integer_exp spaces.

Syntax

```
space(integer_exp)
```

str

Returns a string representation of numeric_exp. integer_exp1 is the length of the string returned. integer_exp2 is the number of decimal digits.

Syntax

```
str(numeric_exp [ , integer_exp1 [ , integer_exp2 ] ] )
```

stuff

Returns a string where length (integer_exp2) characters have been deleted from string_exp1 beginning at start (integer_exp1) and where string_exp2 has been inserted into string_exp1 at start. The first character in a string is at position 1.

Syntax

```
stuff(string_exp1, integer_exp1, integer_exp2, string_exp2)
```

year

Returns the year portion of date_exp. Same as extract(year from date_exp).

Syntax

```
year(date_exp)
```

SQL Server Cast

cast_char

Returns the value of the expression cast as a char.

Syntax

```
cast_char(exp)
```

cast_float

Returns the value of the expression cast as a float.

Syntax

```
cast_float(exp)
```

cast_integer

Returns the value of the expression cast as an integer.

Syntax

```
cast_integer(exp)
```

cast_real

Returns the value of the expression cast as a real.

Syntax

```
cast_real(exp)
```

cast_smallint

Returns the value of the expression cast as a small integer.

Syntax

```
cast_smallint(exp)
```

SQL Server Math

log

Returns the natural logarithm of numeric_exp.

Syntax

```
log(numeric_exp)
```

log10

Returns the base ten logarithm of numeric_exp.

Syntax

```
log10(numeric_exp)
```

pi

Returns the constant value of pi as a floating point value.

Syntax

```
pi()
```

rand

Generates a random number using integer_exp as a seed value.

Syntax

```
rand(integer_exp)
```

SQL Server 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)
```

atn2

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_exp1.

Syntax

```
atn2(numeric_exp1, numeric_exp2)
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos(numeric_exp)
```

cot

Returns the cotangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cot(numeric_exp)
```

degrees

Returns numeric_exp radians converted to degrees.

Syntax

```
degrees(numeric_exp)
```

radians

Returns the number of radians converted from numeric_exp degrees.

Syntax

```
radians(numeric_exp)
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin(numeric_exp)
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan(numeric_exp)
```

Teradata

account

This function returns the account string for the current user.

Syntax

```
{account}
```

add_months

Returns the date or the datetime resulting from adding integer_exp months to date_exp or datetime_exp.

Syntax

```
add_months ( date_exp | datetime_exp, integer_exp )
```

bytes

This function returns the number of bytes contained in the specified byte string.The byte_exp are restricted to BYTE or VARBYTE.

Syntax

```
bytes ( byte_exp )
```

case_n

This function evaluates a list of conditions and returns the position of the first condition that evaluates to TRUE, provided that no prior condition in the list evaluates to UNKNOWN. The keywords must be enclosed in curly brackets. The NO CASE is an optional condition that evaluates to TRUE if every conditional_expression in the list evaluates to FALSE. The NO CASE OR UNKNOWN condition evaluates to TRUE if every conditional_expression in the list evaluates to FALSE, or if a conditional_expression evaluates to UNKNOWN and all prior conditions in the list evaluate to FALSE. The UNKNOWN is an optional condition that evaluates to TRUE if a conditional_expression evaluates to UNKNOWN and all prior conditions in the list evaluate to FALSE.

Syntax

```
case_n ( condition_exp_list [, NO CASE | UNKNOWN | NO CASE OR UNKNOWN [,
UNKNOWN ]] )
```

char2hexint

This function returns the hexadecimal representation for a character string.

Syntax

```
char2hexint ( string_exp )
```

characters

This function returns an integer value representing the number of logical characters or bytes contained in the specified operand string.

Syntax

```
characters ( string_exp )
```

concat

Returns a string that is the result of concatenating string_exp1 to string_exp2.

Syntax

```
concat ( string_exp1, string_exp2 )
```

database

This function returns the name of the default database for the current user.

Syntax

```
{database}
```

date

This function returns the current date.

Syntax

```
{date}
```

format

This function returns the declared format for the named expression. The data type returned by a FORMAT phrase is a variable character string of up to 30 characters.

Syntax

```
format ( expression )
```

index

This function returns the position in string_exp1 where string_exp2 starts.

Syntax

```
index ( string_exp1, string_exp2 )
```

log

Computes the base 10 logarithm of an argument. The numeric_exp is a non-zero, positive numeric expression.

Syntax

```
log ( numeric_exp )
```

nullif

This function returns NULL if scalar_exp1 and scalar_exp2 are equal. Otherwise, it returns its first argument, scalar_exp1. The scalar_exp1 and scalar_exp2 can be any data type.

Syntax

```
nullif ( scalar_exp1, scalar_exp2 )
```

nullifzero

This function converts data from zero to null to avoid problems with division by zero.

Syntax

```
nullifzero ( numeric_exp )
```

profile

This function returns the current profile for the session or NULL if none.

Syntax

```
{profile}
```

random

This function returns a random integer number for each row of the results table. The lower_bound and upper_bound are integer constants. The limits for lower_bound, upper_bound range from -2147483648 to 2147483647, inclusive. The upper_bound must be greater than or equal to lower_bound.

Syntax

```
random ( lower_bound, upper_bound )
```

role

This function returns the current role for the session or NULL if none.

Syntax

```
{role}
```

session

This function returns the number of the session for the current user.

Syntax

```
{session}
```

soundex

This function returns a character string that represents the Soundex code for string_exp.

Syntax

```
soundex ( string_exp )
```

substr

Returns the substring of string_exp that starts at position integer_exp1. The first character in string_exp is at position 1. integer_exp2 can be used to select fewer characters, by default it selects character to the end of the string.

Syntax

```
substr ( string_exp, integer_exp1 [ , integer_exp2 ] )
```

time

This function returns the current time based on a 24-hour day.

Syntax

```
{time}
```

type

This function returns the data type defined for an expression.

Syntax

```
type ( expression )
```

user

This function returns the user name of the current user.

Syntax

```
{user}
```

vargraphic

This function returns a character string that represents the vargraphic code for string_exp.

Syntax

```
vargraphic ( string_exp )
```

zeroifnull

This function converts data from null to 0 to avoid cases where a null result creates an error. If the numeric_exp is not null, it returns the value of the numeric_exp, if numeric_exp is a character string, it is converted to a numeric value of FLOAT data type. If the numeric_exp is null or zero, it returns zero.

Syntax

```
zeroifnull ( numeric_exp )
```

Teradata Trigonometry

acos

Returns the arccosine of numeric_exp in radians. The arccosine is the angle whose cosine is numeric_exp. The values of numeric_exp must be between -1 and 1, inclusive.

Syntax

```
acos ( numeric_exp )
```

acosh

Returns the inverse hyperbolic cosine of an argument. The numeric_exp can be any real number equal to or greater than 1.

Syntax

```
acosh ( numeric_exp )
```

asin

Returns the arcsine of numeric_exp in radians. The arcsine is the angle whose sine is numeric_exp. The values of numeric_exp must be between -1 and 1, inclusive.

Syntax

```
asin ( numeric_exp )
```

asinh

Returns the inverse hyperbolic sine of an argument. The numeric_exp can be any real number.

Syntax

```
asinh ( 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. ATAN2(x,y) equals ATAN(y/x), except that x can be 0 in ATAN2(x,y).The returned angle is between - and π radians, excluding π.

Syntax

```
atan2 ( numeric_exp1, numeric_exp2 )
```

atanh

Returns the inverse hyperbolic tangent of an argument. The numeric_exp can be any real number between 1 and -1, excluding 1 and -1.

Syntax

```
atanh (numeric_exp )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

cosh

Returns the hyperbolic cosine of an argument. The numeric_exp can be any real number.

Syntax

```
cosh ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

sinh

Returns the hyperbolic sine of an argument. The numeric_exp can be any real number.

Syntax

```
sinh ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

tanh

Returns the hyperbolic tangent of an argument. The numeric_exp can be any real number.

Syntax

```
tanh ( numeric_exp )
```

SAP BW

SAP BW Trigonometry

arccos

Returns the arccosine of numeric_exp in radians. The arccosine is the angle whose cosine is numeric_exp.

Syntax

```
arccos ( numeric_exp )
```

arcsin

Returns the arcsine of numeric_exp in radians. The arcsine is the angle whose sine is numeric_exp.

Syntax

```
arcsin ( numeric_exp )
```

arctan

Returns the arctangent of numeric_exp in radians. The arctangent is the angle whose tangent is numeric_exp.

Syntax

```
arctan ( numeric_exp )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

coshyp

Returns the hyperbolic cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
coshyp ( numeric_exp )
```

sinhyp

Returns the hyperbolic sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sinhyp ( numeric_exp )
```

tanhyp

Returns the hyperbolic tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tanhyp ( numeric_exp )
```

SAP BW Math

log10

Returns the base ten logarithm of numeric_exp.

Syntax

```
log10 ( numeric_exp )
```

Sybase

ascii

Returns a number representing the ascii code value of the leftmost character of string_exp.

Syntax

```
ascii ( string_exp )
```

Example

```
ascii( 'A' )
```

Result: 65

char

Converts a single-byte integer value to a character value. char is usually used as the inverse of ascii. integer_exp must be between 0 and 255. Returns a char datatype. If the resulting value is the first byte of a multibyte character, the character may be undefined.

Syntax

```
char ( integer_exp )
```

charindex

Searches string_exp2 for the first occurrence of string_exp1and returns an integer, which represents its starting position. If string_exp1 is not found, it returns 0. If string_exp1 contains wildcard characters, charindex treats it them as literals

Syntax

```
charindex ( string_exp1, string_exp2 )
```

datalength

Returns the length in bytes of the string.

Syntax

```
datalength ( string_exp )
```

dateadd

Returns the date resulting from adding integer_exp units indicated by datepart(e.g. day, month, year) to date_exp. Note that the datepart argument must be enclosed in curly brackets.

Syntax

```
dateadd ( ' { ' datepart ' } ' , integer_exp, date_exp )
```

Example

```
dateadd ( {dd}, 16, 1997-06-16 )
```

Result: Jul 2, 1997

datediff

Returns the number of units indicated by datepart(e.g. day, month, year) between date_exp1 and date_exp2. Note that the datepart argument must be enclosed in curly brackets.

Syntax

```
datediff ( ' { ' datepart ' } ' , date_exp1, date_exp2 )
```

Example

```
datediff ( {yy}, 1984-01-01, 1997-01-01 )
```

Result: 13

datename

Returns part of a datetime, smalldatetime, date or time value as an ASCII string. Note that the datepart argument must be enclosed in curly brackets.

Syntax

```
datename ( ' { ' datepart ' } ' , date_exp )
```

Example

```
datename ( {mm}, 1999-05-01 )
```

Result: May

datepart

Returns part of a datetime, smalldatetime, date or time value (for example, the month) as an integer. Note that the datepart argument must be enclosed in curly brackets.

Syntax

```
datepart ( ' { ' datepart ' } ' , date_exp )
```

Example

```
datepart ( {mm}, 1999-05-01 )
```

Result: 5

day

Returns the day of the month (1-31) from date_exp.

Syntax

```
day ( date_exp )
```

difference

Returns an integer value representing the difference between the values returned by the data source_specific soundex function for string_exp1 and string_exp2. 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_exp1, string_exp2 )
```

getdate

Returns current system date and time.

Syntax

```
getdate ()
```

left

Returns the leftmost integer_exp characters of string_exp.

Syntax

```
left ( string_exp, integer_exp )
```

ltrim

Returns string_exp with leading spaces removed.

Syntax

```
ltrim ( string_exp )
```

month

Returns the month (an integer from 1-12) from date_exp.

Syntax

```
month ( date_exp )
```

patindex

Returns an integer which represents the starting position if the first occurrence of string_exp1 is in the string_exp2; returns 0 if string-exp1 is not found. By default, patindex returns the offset in characters. To return offset in bytes, that is, multibyte character string, specify using bytes. The % wildcard character must precede and follow pattern in string_exp1, except when searching for first or last characters

Syntax

```
patindex ( string_exp1, string_exp2 [ using {bytes | chars | characters} ] )
```

rand

Returns a random float value between 0 and 1, using the optional integer as a seed value.

Syntax

```
rand ( integer_exp )
```

replicate

Returns a string with the same datatype as string_exp, containing the same expression repeated integer_exp times or as many times as will fit into a 225-byte space, whichever is less.

Syntax

```
replicate ( string_exp, integer_exp )
```

reverse

Returns the reverse of the character or binary expression; if string_exp is "abcd", it returns "dcba".

Syntax

```
reverse ( string_exp )
```

right

Returns the rightmost integer_exp characters of string_exp.

Syntax

```
right ( string_exp, integer_exp )
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point.

Syntax

```
round ( numeric_exp, integer_exp )
```

rtrim

Returns string_exp with trailing spaces removed.

Syntax

```
rtrim ( string_exp )
```

soundex

Returns a four-character soundex code for character strings that are composed of a contigous sequence of valid single- or double byte Roman letter.

Syntax

```
soundex ( string_exp )
```

space

Returns a string with the indicated number of single-byte space.

Syntax

```
space ( integer_exp )
```

str

Returns a string representation of numeric_exp. integer_exp1 is the length of the string returned. integer_exp2 is the number of decimal digits. length and decimal are optional(default length is 10; default decimal is 0.)

Syntax

```
str ( numeric_exp [ , integer_exp1 [ , integer_exp2 ] ] )
```

stuff

Delete integer_exp2 characters from string_exp1 at integer_exp1, and then insert string_exp2 into string_exp1 at integer_exp. To delete characters without inserting other characters, string_exp2 should be NULL, not " ", which indicates a single space.

Syntax

```
stuff ( string_exp1, integer_exp1, integer_exp2, string_exp2 )
```

substring

Returns the substring of string_exp that starts at position integer_exp1. integer_exp2 specifies the number of characters in the substring.

Syntax

```
substring ( string_exp, integer_exp1, integer_exp2 )
```

to_unichar

Returns a unichar expression having the value of the integer_exp. If the integer_exp is in the range 0xD800..0xDFFF, the operation is aborted. If the integer_exp is in the range 0..0xFFFF, a single Unicode value is returned. If the integer_exp is in the range 0x10000..0x10FFFF, a surrogate pair is returned.

Syntax

```
to_unichar ( integer_exp )
```

uhighsurr

Returns 1 if the Unicode value at integer_exp is the high half of a surrogate pair (which should appear first in the pair). Otherwise, 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_exp, integer_exp )
```

ulowsurr

Returns 1 if the Unicode value at integer_exp is the low half of a surrogate pair (which should appear second in the pair). Otherwise, 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_exp, integer_exp )
```

uscalar

Returns the Unicode scalar value for the first Unicode character in string_exp. 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 uchar_expr containing an unmatched surrogate half and the operation aborted.

Syntax

```
uscalar ( string_exp )
```

year

Returns the year from date_exp.

Syntax

```
year ( date_exp )
```

Sybase Math

log

Returns the natural logarithm of numeric_exp.

Syntax

```
log ( numeric_exp )
```

log10

Returns the base ten logarithm of numeric_exp.

Syntax

```
log10 ( numeric_exp )
```

pi

Returns the constant value of pi as a floating point value.

Syntax

```
pi ()
```

sign

Returns an indicator of the sign of numeric_exp: +1 if numeric_exp is positive, 0 if zero or -1 if negative.

Syntax

```
sign ( numeric_exp )
```

Sybase 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 )
```

tan

Returns the tangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
tan ( numeric_exp )
```

atn2

Returns the angle (in radians) whose tangent is (numeric_exp1/numeric_exp2).

Syntax

```
atn2 ( numeric_exp1, numeric_exp2 )
```

cos

Returns the cosine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cos ( numeric_exp )
```

cot

Returns the cotangent of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
cot ( numeric_exp )
```

degrees

Returns numeric_exp radians converted to degrees.

Syntax

```
degrees ( numeric_exp )
```

radians

Return the degree equivalent of numeric_exp. Results are of the same type as numeric. For expressions of type numeric or decimal, the results have an internal precision of 77 and a scale equal to that of the numeric expression. When the money datatype is used, internal conversion to float may cause loss of precision.

Syntax

```
radians ( numeric_exp )
```

sin

Returns the sine of numeric_exp where numeric_exp is an angle expressed in radians.

Syntax

```
sin ( numeric_exp )
```

Report functions

AsOfDate

Returns the date value of the As of Time expression, if it is defined. Otherwise, AsofDate returns the report execution date.

Syntax

```
AsOfDate()
```

AsOfTime

Returns the time value of the As of Time expression, if it is defined. Otherwise, AsofTime returns the report execution time.

Syntax

```
AsOfTime()
```

BurstKey

Returns burst key.

Syntax

```
BurstKey()
```

BurstRecipients

Returns the distribution list of burst recipients.

Syntax

```
BurstRecipient()
```

GetLocale

Returns run locale (deprecated).

Syntax

```
GetLocale()
```

IsBursting

Returns boolean 1 (TRUE) when report will be distributed to given recipient; otherwise, 0 (FALSE).

Syntax

```
IsBursting(recipientName)
```

Locale

Returns run locale.

Syntax

```
Locale()
```

ModelPath

Returns model path.

Syntax

```
ModelPath()
```

Now

Returns current system time.

Syntax

```
Now()
```

PageCount

Returns the current page count. If you run the report, this function works only when the report output is PDF or Excel. If you save the report output, this function works for all formats.

Syntax

```
PageCount()
```

PageName

Returns the current page name.

Syntax

```
PageName()
```

PageNumber

Returns current page number.

Syntax

```
PageNumber()
```

ParamCount

Returns parameter count of the variable identified by parameterName.

Syntax

```
ParamCount(parameterName)
```

ParamDisplayValue

Returns a string that is the parameter display value of the variable identified by parameterName.

Syntax

```
ParamDisplayValue(parameterName)
```

ParamName

Returns parameter name of the variable identified by parameterName.

Syntax

```
ParamName(parameterName)
```

ParamNames

Returns all parameter names.

Syntax

```
ParamNames()
```

ParamValue

Returns a string that is the parameter value of the variable identified by parameterName.

Syntax

```
ParamValue(parameterName)
```

ReportAuthorLocale

Returns author locale.

Syntax

```
ReportAuthorLocale()
```

ReportCreateDate

Returns the date when the report was created.

Syntax

```
ReportCreateDate()
```

ReportDate

Returns report execution date and time.

Syntax

```
ReportDate()
```

ReportDescription

Returns report description. This function works only when the report is run from Cognos Connection.

Syntax

```
ReportDescription()
```

ReportID

Returns the report id.

Syntax

```
ReportID()
```

ReportLocale

Returns run locale.

Syntax

```
ReportLocale()
```

ReportName

Returns report name. This function works only when the report is run from Cognos Connection.

Syntax

```
ReportName()
```

ReportOption

Returns the value of run option variable identified by optionName. Possible values for optionName: attachmentEncoding, burst, cssURL, email, emailAsAttachment, emailAsURL, emailBody, emailSubject, emailTo, emailToAddress, history, metadataModel, outputEncapsulation, outputFormat, outputLocale, outputPageDefinition, outputPageOrientation, primaryWaitThreshold, print, printer, printerAddress, prompt, promptFormat, saveAs, saveOutput, secondaryWaitThreshold, verticalElements, xslURL.

Syntax

```
ReportOption(optionName)
```

ReportOutput

Returns the name of the output format. Possible return values: CSV, HTML, HTMLFragment, PDF, XHTML, XML.

Syntax

```
ReportOutput()
```

ReportPath

Returns report path. This function works only when the report is run from Cognos Connection.

Syntax

```
ReportPath()
```

ReportProductLocale

Returns product locale.

Syntax

```
ReportProductLocale()
```

ReportSaveDate

Returns the date when the report was last saved.

Syntax

```
ReportSaveDate()
```

RowNumber

Returns current row.

Syntax

```
RowNumber()
```

ServerLocale

Returns the locale of the server that runs the report.

Syntax

```
ServerLocale()
```

ServerName

Returns the name of the server that runs the report.

Syntax

```
ServerName()
```

Today

Returns current system date.

Syntax

```
Today()
```

URLEncode

Returns the url encoded value of the input text.

Syntax

```
URLEncode(text)
```

Data type casting functions

_add_days

Returns the datetime resulting from adding integer_exp days to timestamp_exp.

Syntax

```
_add_days ( timestamp_exp, integer_exp )
```

_add_months

Returns the datetime resulting from adding integer_exp months to timestamp_exp.

Syntax

```
_add_months ( timestamp_exp, integer_exp )
```

_add_years

Returns the datetime resulting from adding integer_exp years to timestamp_exp.

Syntax

```
_add_years ( timestamp_exp, integer_exp )
```

_age

Returns a number that is obtained from subtracting timestamp_exp from today's date in YYYYMMDD format (years, months, days).

Syntax

```
_age ( timestamp_exp )
```

_day_of_week

Returns the day of week (between 1 and 7), where 1 is the first day of the week as indicated by the second parameter.

Syntax

```
_day_of_week ( timestamp_exp, integer )
```

_day_of_year

Returns the ordinal for the day of the year in date_ exp (1 to 366). Also known as Julian day.

Syntax

```
_day_of_year ( timestamp_exp )
```

_days_between

Returns a positive or negative number representing the number of days between the two datetime expressions. If timestamp_exp1 < timestamp_exp2 then the result will be a negative number.

Syntax

```
_days_between ( timestamp_exp1, timestamp_exp2 )
```

_days_to_end_of_month

Returns a number representing the number of days remaining in the month represented by the datetime expression timestamp_exp.

Syntax

```
_days_to_end_of_month ( timestamp_exp )
```

_first_of_month

Returns a datetime that is the first day of the month represented by timestamp_exp.

Syntax

```
_first_of_month ( timestamp_exp )
```

_last_of_month

Returns a datetime that is the last day of the month represented by timestamp_exp.

Syntax

```
_last_of_month ( timestamp_exp )
```

_make_timestamp

Returns a timestamp constructed from integer_exp1 (the year), integer_exp2 (the month) and integer_exp3 (the day).

Syntax

```
_make_timestamp ( integer_exp1, integer_exp2, integer_exp3 )
```

_months_between

Returns a positive or negative number representing the number of months between the two datetime expressions. If timestamp_exp1 < timestamp_exp2 then the result will be a negative number.

Syntax

```
_months_between ( timestamp_exp1, timestamp_exp2 )
```

_week_of_year

Returns the number of the week of the year of the timestamp_exp according to ISO 8601, in which 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.

Syntax

```
_week_of_year ( timestamp_exp )
```

_years_between

Returns a positive or negative integer number representing the number of years from timestamp_exp1 to timestamp_exp2. If timestamp_exp1 < timestamp_exp2 then a negative value is returned.

Syntax

```
_years_between ( timestamp_exp1, timestamp_exp2 )
```

_ymdint_between

Returns a number representing the difference between the datetime expressions timestamp_exp1 and timestamp_exp2. This 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 ( timestamp_exp1, timestamp_exp2 )
```

abs

Returns the absolute value of numeric_exp. The sign of negative values is changed to positive.

Syntax

```
abs ( numeric_exp )
```

ceiling

Returns the smallest integer greater than or equal to numeric_exp.

Syntax

```
ceiling ( numeric_exp )
```

character_length

Returns the number of characters in string_exp.

Syntax

```
character_length ( string_exp )
```

exp

Returns e raised to the power of numeric_exp. The constant e is the base of the natural logarithm.

Syntax

```
exp ( numeric_exp )
```

extract

Returns an integer representing the value of datepart ("year", "month", "day", "hour", "minute", "second" (=default)) in datetime_exp.

Syntax

```
extract ( datepart , datetime_exp )
```

floor

Returns the largest integer less than or equal to numeric_exp.

Syntax

```
floor ( numeric_exp )
```

ln

Returns the natural logarithm of the numeric_exp.

Syntax

```
ln ( numeric_exp )
```

lower

Returns string_exp with all uppercase characters shifted to lowercase.

Syntax

```
lower ( string_exp )
```

mapNumberToLetter

The user should supply the initial letter and the number is added to it: e.g. mapNumberToLetter ('a', 1) will result in 'b'.

Syntax

```
mapNumberToLetter(String format, Int32 value,)
```

mod

Returns the integer value of the remainder (modulo) when the first number is divided by the second.

Syntax

```
mod(Int64 value, Int64 value)
```

octet_length

Returns the number of bytes in string_exp.

Syntax

```
octet_length ( string_exp )
```

position

Returns integer value representing the position of the first string_exp in the second string_exp or 0 when the first string_exp is not found.

Syntax

```
position ( string_exp , string_exp )
```

round

Returns numeric_exp rounded to the nearest value integer_exp places right of the decimal point. If integer_exp is negative, numeric_exp is rounded to the nearest value absolute (integer_exp) places to the left of the decimal point, e.g., round (125, -1) rounds to 130.

Syntax

```
round ( numeric_exp, integer_exp )
```

sqrt

Returns the square root of numeric_exp. numeric_exp must be non-negative.

Syntax

```
sqrt ( numeric_exp )
```

substring

Returns the substring of string_exp that starts at position integer_exp1 for integer_exp2 characters or to the end of string_exp if integer_exp2 is -1. The first character in string_exp is at position 1.

Syntax

```
substring ( string_exp , integer_exp1, integer_exp2 )
```

trim

Returns a string_exp trimmed of leading and/or trailing blanks or trimmed of a character specified in match_character_exp. trim_what may be: "LEADING", "TRAILING" or "BOTH" (default). match_character_exp = empty string to trim blanks, or specifes a character to be trimmed. string_exp = the string to be trimmed.

Syntax

```
trim ( trim_what, match_character_exp, string_exp )
```

upper

Returns string_exp with all lowercase characters shifted to uppercase.

Syntax

```
upper ( string_exp )
```

Appendix F: Producing Reports in Microsoft Excel Format

The following limitations exist when producing reports in Microsoft Excel format.

Microsoft Excel Limitations

The following issues were identified in Microsoft Excel and affect producing Cognos 8 reports in Excel format.

Unable to Load Images from the Cognos 8 Content Store in a Report

If a report contains an image whose URL points to the Cognos 8 content store, Microsoft Excel 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 2000 and 2002.

A Blank Worksheet is Opened

If Microsoft Excel cannot download a worksheet within a timeout period, Excel may instead open a blank worksheet.

A Warning Message Appears When Excel Opens a Cognos 8 Report

Each time Microsoft Excel opens a Cognos 8 report, the following message appears:

Some of the files in this Web page aren't in the expected location. Do you want to download them anyway? If you're 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. Cognos 8 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 a Cognos 8 report in Excel.

Using 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 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 Excel reports in 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.

Loading Excel Reports in Netscape 7.01 Is Not Supported

This version of Cognos 8 does not support loading Microsoft Excel reports in Netscape 7.01.

Nested Labels in Charts Are Not Supported

Currently, it is not possible to specify nested labels for the category axis via XML.

Data Series Are Truncated

Microsoft Excel may group data series or categories differently when compared to a chart produced by Cognos 8.

A 1 KB buffer limit in Excel limits the maximum number of data series per chart to 120. Data series over 120 are truncated.

Charts and Custom Colors

If a Cognos 8 report contains a chart that uses custom colors, Excel 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, Excel 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*

Discrete Axis Label Skip Control in Charts

In Cognos 8 charts, you can control the skipping of discrete axis labels. This feature is not supported in Excel charts.

Formatting Limitations

About 70% of the formatting functions available in Cognos 8 are supported in Microsoft Excel. The following table shows which formatting functions are supported in Excel and which are not.

Cognos 8 format	Supported in Excel	Notes
Currency	✔	
Currency Symbol	✔	
Decimal Separator		Excel does not allow changing locale-dependent formatting attributes.
Exponential Symbol		Excel does not allow changing locale-dependent formatting attributes.
Format Width		Not required. Excel automatically adjusts the width.
Group Separator		Excel does not allow changing locale-dependent formatting attributes.
Group Size	✔	
International Currency Symbol		
List Separator		Not required. Excel automatically adjusts the width.

Cognos 8 format	Supported in Excel	Notes
Maximum Fraction Digits	✔	
Maximum Integer Digits	✔	
Minimum Exponent Digits	✔	
Minimum Fraction Digits	✔	
Minimum Integer Digits	✔	
Minus Sign	✔	
Monetary Decimal Separator		Excel does not allow changing the locale-dependent formatting attributes.
Multiplier		Not supported by Excel.
Negative Prefix	✔	
Negative Suffix	✔	
Pad Character	✔	
Percent Symbol	✔	Supported for lists and crosstabs. The percent symbol is not supported for charts.
PerMill Symbol		Not supported by Excel.
Plus Sign		
Positive Prefix	✔	
Positive Suffix	✔	
Scale		Excel has a different scaling formula than Cognos 8.
Secondary Grouping Size	✔	
When Negative	✔	
WhenZero	✔	
Use Currency Symbol	✔	
Use Grouping	✔	
Use Scientific	✔	
Use Trailing Currency Symbol	✔	

Cognos 8 format	Supported in Excel	Notes
Use Trailing Sign	✔	
AM String		Excel does not allow changing locale-dependent formatting attributes.
Calendar		Excel does not allow changing the calendar.
Date Separator Symbol	✔	
Day Name		Excel does not allow changing locale-dependent formatting attributes.
Day Short Name		Excel does not allow changing locale-dependent formatting attributes.
Decimal Delimiter Symbol		Excel does not allow changing locale-dependent formatting attributes.
Era Name		Not supported by Excel.
First Day Of Week		Not supported by Excel.
Month Name		Excel does not allow changing locale-dependent formatting attributes.
Month Short Name		Excel does not allow changing locale-dependent formatting attributes.
PM String		Excel does not allow changing locale-dependent formatting attributes.
Symbol Display Order	✔	
Time Separator Symbol	✔	
Show AmPm Symbol	✔	
Show Century	✔	
Show Clock	✔	
Show Days	✔	
Show Era		Not supported by Excel.
Show Hours	✔	
Show Milliseconds	✔	
Show Minutes	✔	

Overline Text Format

Microsoft Excel does not support the overline text format.

Text Strings with More Than 255 Characters

Cells in Microsoft Excel 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 ######.

Reports with More Than 256 Columns

Microsoft Excel limits the size of a worksheet size to 65536 rows by 256 columns. If your report contains more than 65536 rows, it is split into multiple worksheets. The number of worksheets that your report can contain is limited by the physical memory of your computer. If your report contains more than 256 columns, the following error occurs:

Reports with more than 256 columns cannot be rendered in Excel.

Table and Column Width

Microsoft Excel 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 the workbook is saved, only a single table is saved per worksheet.

Excel Formats and Secure Socket Layer (SSL)

SSL is supported for the following formats and Microsoft Excel versions.

Format	Version
Excel 2000 single sheet	Microsoft Excel 2000, 2002, XP, 2003
Excel 2000	Microsoft Excel 2003
Excel 2002	Microsoft Excel 2002, XP, 2003

Number Format Becomes Currency Format 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.

Report Shows 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 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.

Avoid exceeding the Excel limit by redesigning the report to present the data using non-nested structures.

Cognos 8 Limitations

The following Cognos 8 limitations exist when producing reports in Microsoft Excel format.

Accessing Reports on a Remote Server

To access a report in Excel format on a remote server, 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 Cognos Configuration.

Drill-through Reports

Cognos 8 does not support drill-through for reports in Excel format.

Map Reports

Cognos 8 does not support map reports in Excel format.

Formats Not Supported for Reports in Excel Format

Cognos 8 does not support the following for reports in Excel formats:

- 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 Microsoft Excel limitations.

Hyperlink Buttons

Microsoft Excel does not support hyperlink buttons.

Emailing Reports in Excel Format

Cognos 8 can send Excel reports in HTML and XML format by email. However, the Excel email attachments must be saved to your computer before you can view them.

Charting Support in Excel and Cognos 8

The following Cognos 8 chart properties are not supported in Microsoft Excel:

- 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

In addition, Cognos 8 makes sure that Excel reuses the same color palette that is defined in Cognos 8. However, Excel can only use the first 16 colors from the Cognos 8 palette. If the number of categories in a chart exceeds 16, the rest of the colors are taken from the default Excel palette.

About 70% of the chart types available in Cognos 8 are matched in Microsoft Excel. The following table shows which chart types are supported in Microsoft Excel 2000 and later and which are not. Charts not supported will return a default column chart.

Chart group	Chart type	Supported in Excel	Notes
Column	Column	✔	Similar
	Column 3-D	✔	Results are better in HTML
	Stacked	✔	Similar
	Stacked 3-D	✔	Results are better in HTML
	100% stacked	✔	Similar
	100% stacked 3-D	✔	Chart is viewed from a different angle
	3-D axis	✔	Similar, but category data is presented in the reverse order
Bar	Bar	✔	Similar
	Bar 3-D	✔	Similar
	Stacked	✔	Similar
	Stacked 3-D	✔	Similar
	100% stacked	✔	Similar
	100% stacked 3-D	✔	Similar
Progressive	Column		
	Column with 3-D effect		
	Bar		
	Bar with 3-D effect		
Pareto	Stacked column		

Chart group	Chart type	Supported in Excel	Notes
	Stacked column 3-D		
	Stacked bar		
	Stacked bar 3-D		
Line	Line with markers	✔	Similar
	Line	✔	Similar
	Line 3-D	✔	Shows as a Line 3-D axis chart
	Stacked line with markers	✔	
	Stacked line	✔	Similar
	Stacked line 3-D	✔	Shows as a stacked line
	100% stacked line with markers	✔	Similar
	100% stacked line	✔	Similar
	100% stacked line 3-D	✔	Shows as 100% stacked line
	3-D Axis	✔	Similar
Pie	Pie	✔	Cognos 8 shows many types of pie charts, Microsoft Excel shows only one type
	Pie 3-D	✔	Microsoft Excel shows the pie chart at a different angle
	Donut	✔	Microsoft Excel may fill in the donut hole to accommodate extra measures
	Donut 3-D		Shows as a donut chart
Area	Area	✔	Similar
	Area 3-D	✔	Shows a smaller version of an Area 3-D axis chart
	Stacked area	✔	Similar
	Stacked area 3-D	✔	Shows at a different angle
	100% stacked area	✔	Similar

Chart group	Chart type	Supported in Excel	Notes
	100% stacked area 3-D	✔	Shows at a different angle
	3-D Axis	✔	Similar
Scatter, bubble, point	Scatter	✔	Similar
	Bubble	✔	Similar but actual details on the chart may differ slightly
	Quadrant		
	Polar		
Radar, polar	Radar with markers	✔	Similar
	Radar	✔	Returns Radar with markers
	Area radar	✔	Microsoft Excel names this chart type Filled radar
	Stacked area radar		
	Bipolar		
Combination	Combination	✔	
	Combination 3-D		
	Stacked combination	✔	
	Stacked combination 3-D		
	3-D axis		Returns Combination
Gauge	Dial gauge		

Appendix G: Report Studio Object and Property Reference

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

The following objects and properties are referenced:

Report Studio Objects

These objects are visible in the Report Studio work area. They can be inserted from the toolbox tab.

Report Studio Properties

These properties can be set on Report Studio objects. They are available in the lower-left pane of Report Studio.

Data Formatting Properties

These properties can be set on data values by selecting **Layout 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 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, Borders, Chart Type, Values

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, Borders, Chart Type, Values

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, Background Color, Background Image, Baselines, Border, Box Type, Class, Conditional Palette, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Margin, Master Detail Relationships, Name, Notes, Numerical Axis, Padding, Pagination, Palette, Query, Relative Alignment, Render Variable, Series Color, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Visible, X Axis, Y Axis

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, Borders, Chart Type, Values

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, Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Class, Conditional Palette, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Margin, Master Detail Relationships, Name, Notes, Padding, Pagination, Palette, Point Shape, Point Size (pt), Query, Relative Alignment, Render Variable, Show Feelers, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Values, Visible

Angular Axis

The angular numeric axis for a polar chart, including labels, titles, range, and scale.

Properties of Angular Axis

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Angular Measure

The measure that determines the angular position of each data marker on a polar chart.

Properties of Angular Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Area

The data marker used to represent data series in an area chart.

Properties of Area

Axis Assignment, Border Color, Borders, Chart Type, Grouping Type, Value Location, Values, Value Type

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

Axis Title

The title for an axis of the chart.

Properties of Axis Title

Box Type, Class, Default Title, Font, Foreground Color, Master Detail Relationships, Properties, Query, Style Variable, Visible

Bar

A chart in which members of a data series are represented by bars of varying length and color.

Properties of Bar

Axis Assignment, Border Color, Borders, Chart Type, Connecting Lines, Grouping Type, Value Location, Values, Value Type

Baseline

A baseline to be rendered on a chart.

Properties of Baseline

Aggregate Function, Axis Assignment, Bar, Line, or Area Index, Box Type, Data Item Value, Expression, Label, Label, Legend Label, Limit Type, Line Styles, Master Detail Relationships, Member Offset (%), Name, Numeric Value, Percentile, Percent of Axis, Properties, Query, Report Expression, Report Expression, Rollup Aggregate Function, Source Type, Standard Deviations

Baseline

A baseline for a polar chart, scatter chart, or bubble chart.

Properties of Baseline

Aggregate Function, Axis Assignment, Box Type, Data Item Value, Expression, Label, Label, Legend Label, Limit Type, Line Styles, Master Detail Relationships, Name, Numeric Value, Percentile, Percent of Axis, Properties, Query, Report Expression, Rollup Aggregate Function, Source Type, Standard Deviations

Baseline

A baseline for a three-dimensional combination chart.

Properties of Baseline

Aggregate Function, Bar, Line, or Area Index, Box Type, Data Item Value, Expression, Label, Limit Type, Line Styles, Master Detail Relationships, Name, Numeric Value, Percentile, Percent of Axis, Properties, Query, Report Expression, Rollup Aggregate Function, Source Type, Standard Deviations

Baseline

A baseline for a three-dimensional scatter chart.

Properties of Baseline

Aggregate Function, Box Type, Data Item Value, Expression, Label, Limit Type, Line Styles, Master Detail Relationships, Name, Numeric Value, Percentile, Percent of Axis, Properties, Query, Report Expression, Rollup Aggregate Function, Source Type, Standard Deviations

Block

A container into which you can insert other objects.

Properties of Block

Background Color, Background Image, Border, Box Type, Class, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Name, Padding, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible, White Space

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, Data Item Value, Label, Report Expression, Source Type

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 Tool Tips property of this object to Yes.

Properties of Bubble Chart

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Class, Conditional Palette, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Name, Notes, Padding, Pagination, Palette, Point Shape, Query, Regression Line, Relative Alignment, Render Variable, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible

Bubble Measure

The measure that determines the size of each bubble on a bubble chart.

Properties of Bubble Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

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, Data Item, Name

Caption

The caption on a Field Set object.

Properties of Caption

Background Color, Background Image, Border, Class, Font, Foreground Color, Margin, Padding, Style Variable

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

Background Color, Background Image, Fill Effects, Font, Foreground Color, Horizontal Alignment, Style Variable

Chart Footer

A footer for the chart.

Properties of Chart Footer

Box Type, Class, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

Chart Node Member

A data item, and its accompanying text, to render on the chart.

Properties of Chart Node Member

Aggregate Function, Custom Label, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Expression, HTML, HTML Source Variable, Label, Label, Name, Properties, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Sorting, Source Type, Source Type, Source Type, Source Type, Text, Text Source Variable, URL, URL Source Variable

Chart Subtitle

The subtitle for a chart.

Properties of Chart Subtitle

Box Type, Class, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

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

Aggregate Function, Data Format, Data Item Label, Data Item Value, Expression, Label, Name, Report Expression, Rollup Aggregate Function, Source Type, Style Variable, Text, Text Source Variable

Chart Title

The title text that appears at the top of the chart.

Properties of Chart Title

Box Type, Class, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

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

Background Color, Background Image, Border, Box Type, Description, Floating, Font, Foreground Color, Horizontal Alignment, Label, Label, Margin, Padding, Relative Alignment, Selector, Selector, Size & Overflow, Spacing & Breaking, Table Properties, Text Flow & Justification, Vertical Alignment, Visible, White Space

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

Background Color, Background Image, Baselines, Border, Box Type, Chart Orientation, Class, Conditional Palette, Depth, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Name, Notes, Ordinal Axis, Padding, Pagination, Palette, Query, Relative Alignment, Render Variable, Rotate Labels, Series Color, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Value Location, Visible, Visual Angle, Y1 Axis, Y2 Axis, Y2 Axis Position

Component Override

Overrides a child object of the Layout Component Reference object.

Properties of Component Override

Component Reference

Conditional Block

A block that can be used for conditional display.

Properties of Conditional Block

Background Color, Background Image, Block Variable, Border, Box Type, Class, Current Block, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible, White Space

Conditional Block

Contains the default set of layout objects to render based on a report variable.

Properties of Conditional Block

Background Color, Background Image, Block Variable, Border, Box Type, Class, Current Block, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible, White Space

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

Background Color, Background Image, Border, Box Type, Class, Default Measure, Default Measure Solve Order, Fact Cells Precedence, Floating, Font, Foreground Color, Has Fact Cells, Horizontal Alignment, Margin, Master Detail Relationships, Name, Pagination, Query, Relative Alignment, Render Variable, Rows Per Page, Size & Overflow, Style Variable, Table Properties, Text Flow & Justification, Visible

Crosstab Columns

Overrides the style for Crosstab Column Member objects that is defined in the GlobalReportStyles.css file.

Properties of Crosstab Columns

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Crosstab Columns

A list of columns in a crosstab.

Properties of Crosstab Columns

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, 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 Image, Border, Box Type, Class, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Expression, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, 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, Background Color, Background Image, Border, Box Type, Class, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill Throughs, Expression, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source 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 Image, Border, Box Type, Class, Column Intersection, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Define Contents, Drill Throughs, Expression, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Row Intersection, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source 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

Aggregate Function, Background Color, Background Image, Border, Box Type, Class, Column Intersection, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Define Contents, Drill Throughs, Expression, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Row Intersection, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Vertical Alignment, White Space

Crosstab Node Member

A member in the crosstab node.

Properties of Crosstab Node Member

Aggregate Function, Background Color, Background Image, Border, Box Type, Class, Data Format, Data Item, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill Throughs, Expression, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Indentation Length, Indentation Start Level, Intersection Name, Label, Label, Name, Padding, Pagination, Properties, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Size & Overflow, Sorting, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Vertical Alignment, White Space

Crosstab Rows

Overrides the style for Crosstab Row Member objects that is defined in the GlobalReportStyles.css file.

Properties of Crosstab Rows

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Crosstab Rows

A list of rows in a crosstab.

Properties of Crosstab Rows

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

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, Background Image, Border, Box Type, Class, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill Throughs, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Intersection Name, Label, Padding, Pagination, Report Expression, Report Expression, Report Expression, Report Expression, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Vertical Alignment, White Space

Cumulation Line

A line that shows the cumulative effect of multiple series members on a measure in a pareto chart.

Properties of Cumulation Line

Cumulation Axis, Cumulation Label, Line Styles, Point Shape, Point Size (pt), Properties, Value Location, Values

Cumulation Line Axis

The axis for the cumulation line in a pareto chart.

Properties of Cumulation Line Axis

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Cumulation Line Label

A label that is rendered with the cumulation line in a pareto chart.

Properties of Cumulation Line Label

Aggregate Function, Data Item Label, Data Item Value, Expression, Label, Name, Report Expression, Rollup Aggregate Function, Source Type, Text, Text Source Variable

Data Item

A set of data values or members.

Properties of Data Item

Aggregate Function, Calculation Intersection, Detail, Expression, Label, Name, Pre-Sort, Rollup Aggregate Function, Solve Order

Date & Time Prompt

A prompt control with which you can select a date and time value.

Properties of Date & Time Prompt

Allow Is Missing, Box Type, Calendar Type, Clock Mode, Default Selections, Display Milliseconds, Display Seconds, First Date, Floating, Hide Adornments, Last Date, Multi-Select, Name, Parameter, Range, Render Variable, Required, Select UI, Style Variable, Visible

Date Prompt

A prompt control with which you can to select a date value.

Properties of Date Prompt

Allow Is Missing, Box Type, Calendar Type, Default Selections, First Date, Floating, Hide Adornments, Last Date, Multi-Select, Name, Parameter, Range, Relative Alignment, Render Variable, Required, Select UI, Style Variable, Visible

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

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

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, Expression, Usage

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

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

Border Color, Borders, Fill Effects, Labels, Style Variable

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, Name

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

Background Color, Background Image, Border, Box Type, Class, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Name, Relative Alignment, Render Variable, Show Caption, Size & Overflow, Style Variable, Visible

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

Axis Title, Background Color, Background Image, Border, Border Color, Borders, Box Type, Class, Conditional Palette, Dial Outline Color, Drill Throughs, Face Color, Fill Effects, Floating, Font, Footer, Foreground Color, Gauge Labels, Gauge Palette, Legend, Margin, Master Detail Relationships, Name, Notes, Numerical Axis, Padding, Pagination, Palette, Query, Relative Alignment, Render Variable, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Visible

Gauge Labels

A label for each gauge in a multiple gauge chart.

Properties of Gauge Labels

Class, Drill Throughs, 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

Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

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, Name, Parameter, Render Variable, Required

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.

Properties of HTML Item

Aggregate Function, Data Item Label, Data Item Value, Description, Expression, HTML, HTML Source Variable, Label, Name, Name, Render Variable, Report Expression, Rollup Aggregate Function, Source Type

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

Background Color, Background Image, Border, Box Type, Class, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Floating, Font, Foreground Color, Margin, Name, Padding, Relative Alignment, Render Variable, Report Expression, Report Expression, Size & Overflow, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Visible

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

Background Color, Background Image, Border, Class, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Floating, Foreground Color, Margin, Name, Padding, Relative Alignment, Render Variable, Report Expression, Report Expression, Size & Overflow, Source Type, Source Type, Style Variable, Text, Text Source Variable, URL, URL Source Variable

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

Aggregate Function, Background Color, Background Image, Border, Box Type, Class, Data Item Label, Data Item Value, Drill Throughs, Expression, Floating, Label, Margin, Name, Name, Relative Alignment, Render Variable, Report Expression, Rollup Aggregate Function, Size & Overflow, Source Type, Style Variable, URL, URL Source Variable, Visible

Interval Prompt

An advanced prompt control that allows you to enter time duration values.

Properties of Interval Prompt

Allow Is Missing, Box Type, Default Selections, Display Milliseconds, Display Seconds, Floating, Hide Adornments, Multi-Select, Name, Parameter, Range, Render Variable, Required, Style Variable, Visible

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

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, Name

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

Component Reference, Embed, Overrides

Legend

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

Properties of Legend

Absolute Position, Auto Truncation, Background Color, Background Image, Border Color, Borders, Bottom Position (px), Box Type, Class, Drill Throughs, Fill Effects, Font, Font Auto-Sizing, Foreground Color, Horizontal Alignment, Left Position (px), Legend Title, Maximum Characters, Position, Right Position (px), Separator, Show Legend Values, Style Variable, Top Position (px), Truncation Text, Visible

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

Box Type, Class, Default Title, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

Level

A set of members with a predefined set of similar characteristics. For example, the members Year 1999 and Year 2000 in the Time dimension form a year level, while the corresponding quarters form a quarter level.

Properties of Level

Caption, Name, Sorting

Level Hierarchy

Specifies how the levels in a dimension are logically ordered.

Properties of Level Hierarchy

Name

Line

The data marker used to represent a data series in a line chart.

Properties of Line

Axis Assignment, Border Color, Borders, Chart Type, Grouping Type, Line, Line Style, Line Weight (pt), Point Shape, Point Size (pt), Value Location, Values, Value Type

List

A layout object that is used to present query results in a list fashion.

Properties of List

Background Color, Background Image, Border, Box Type, Class, Column Titles, Contents Height, Floating, Font, Foreground Color, Grouping & Sorting, Horizontal Alignment, Margin, Master Detail Relationships, Name, Pagination, Properties, Query, Relative Alignment, Render Variable, Rows Per Page, Size & Overflow, Style Variable, Table Properties, Text Flow & Justification, Visible

List Cell

A cell in a row, you can use for a list header or footer.

Properties of List Cell

Aggregate Function, Background Color, Background Image, Border, Box Type, Class, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Expression, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Vertical Alignment, White Space

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

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Column Body

The contents of a column in a list report.

Properties of List Column Body

Aggregate Function, Background Color, Background Image, Border, Box Type, Class, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill Throughs, Expression, Font, Foreground Color, Group Span, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Vertical Alignment, White Space

List Columns

A set of columns in a list.

Properties of List Columns

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Columns

Overrides the style for List Column objects that is defined in the GlobalReportStyles.css file.

Properties of List Columns

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

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

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

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

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Column Title

The title of a list column.

Properties of List Column Title

Aggregate Function, Background Color, Background Image, Border, Box Type, Class, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill Throughs, Expression, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Name, Padding, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Vertical Alignment, White Space

List Footer

The footer that appears at the end of a list on each page on which the list is rendered. It is useful for presenting page totals.

Properties of List Footer

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Properties, Push To Bottom, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

List Header

The header that appears at the start of a list on each page that the list is rendered. It is useful for presenting carry forward totals.

Properties of List Header

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Pagination, Properties, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

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.

Properties of List Page Footer

Push To Bottom

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.

Properties of List Page Header

Display After Overall Header

List Row

A row in a list.

Properties of List Row

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

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

Background Color, Background Image, Border, Box Type, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Map

A chart that uses a map to show data.

Properties of Map

Axis Title, Background Color, Background Image, Border, Box Type, Class, Data Language, Dictionary, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Ignore Data with No Features, Legend, Map & Layers, Margin, Master Detail Relationships, Name, No Data Features Size (pt), Notes, Padding, Pagination, Query, Relative Alignment, Render Variable, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Visible

Map Location

Associates a data series with regions on the region layer of the map.

Properties of Map Location

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Map Location

Associates a data series with points on the point layer of the map.

Properties of Map Location

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Map Refinement Location

Qualifies the data series members that are associated with regions on the map.

Properties of Map Refinement Location

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Map Refinement Location

Qualifies the data series members that are associated with points on the map.

Properties of Map Refinement Location

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Marker

An additional marker that can be placed at a static point on a chart.

Properties of Marker

Aggregate Function, Axis Assignment, Bar, Line, or Area Index, Box Type, Data Item Value, Expression, Label, Label, Master Detail Relationships, Name, Numeric Value, Percentile, Percent of Axis, Point Color, Point Shape, Point Size (pt), Properties, Query, Report Expression, Report Expression, Rollup Aggregate Function, Source Type, Standard Deviations

Marker

A marker on a point chart. Point charts include scatter charts, polar charts, radar charts, and bubble charts.

Properties of Marker

Box Type, Data Item Value, Data Item Value, Label, Master Detail Relationships, Numeric Value, Percentile, Percent of Axis, Point Color, Point Shape, Point Size (pt), Properties, Query, Report Expression, Source Type, Source Type, Standard Deviations

MDX

A multidimensional expression (MDX) query against an OLAP data source.

Properties of MDX

Catalog, Data Source, MDX, Name

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

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, Name

Member Set

A named member set from a data item object specified in the query.

Properties of Member Set

Data Item, Name

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

Background Color, Background Image, Baselines, Border, Box Type, Class, Conditional Palette, Depth, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Marker Color, Marker Label, Markers, Marker Text Location, Master Detail Relationships, Name, Notes, Ordinal Axis, Padding, Pagination, Palette, Performance Pattern, Query, Range Label, Relative Alignment, Render Variable, Series Color, Size & Overflow, Style Variable, Subtitle, Target Color, Target Marker, Target Marker Border Color, Target Marker Position, Target Range (%), Title, Tolerance Color, Tolerance Label, Tooltips, Upper Range Skew (%), Value Location, Visible, Visual Angle, Y1 Axis

Metric Studio Diagram

A Metric Studio history chart rendered as an image.

Properties of Metric Studio Diagram

Description, Diagram Identifier

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

Bottom Position (px), Box Type, Height (px), Left Position (px), Note Border, Style Variable, Width (px)

Note Content

The content and style of a note.

Properties of Note Content

Background Color, Box Type, Class, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

Numerical Axis

The numeric axis for a progressive chart, including labels, titles, range, scale, and gridlines.

Properties of Numerical Axis

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Numerical Axis

The numeric axis for a pareto chart, including labels, titles, and gridlines.

Properties of Numerical Axis

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Ordinal Axis

The axis line for an ordinal, or non-numeric, axis.

Properties of Ordinal Axis

Allow 45° Rotation, Allow 90° Rotation, Allow Skip, Allow Stagger, Axis Labels, Axis Line, Axis Title, Class, Display Frequency, Drill Throughs, First Label Index, Font, Foreground Color, Gridlines, Label Control, Maximum Truncation Characters, Minor Gridlines, Style Variable, Truncation, Truncation Text, Visible

Page

A page in a layout.

Properties of Page

Background Color, Background Image, Border, Class, Font, Foreground Color, Horizontal Alignment, Margin, Master Detail Relationships, Name, Padding, Pagination, Properties, Query, Render Variable, Size & Overflow, Style Variable, Text Flow & Justification

Page Body

The main body of a page.

Properties of Page Body

Background Color, Background Image, Border, Box Type, Class, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Page Footer

The footer of a page.

Properties of Page Footer

Background Color, Background Image, Border, Box Type, Class, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Page Header

The header of a page.

Properties of Page Header

Background Color, Background Image, Border, Box Type, Class, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Page Set

The set of pages to render according to a grouping structure.

Properties of Page Set

Grouping & Sorting, Master Detail Relationships, Name, Pagination, Properties, Query

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

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Chart Orientation, Class, Conditional Palette, Connecting Lines, Cumulative Line, Depth, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Name, Notes, Padding, Pagination, Palette, Query, Relative Alignment, Render Variable, Rotate Labels, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible, Visual Angle

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

Avoid Label Collision, Axis Title, Background Color, Background Image, Border, Border Color, Borders, Box Type, Class, Conditional Palette, Depth, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Hole Size (%), Labels, Legend, Margin, Master Detail Relationships, Name, Notes, Padding, Pagination, Palette, Pie Labels, Query, Relative Alignment, Render Variable, Size & Overflow, Style Variable, Subtitle, 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

Class, Drill Throughs, Font, Foreground Color, Horizontal Alignment, Maximum Truncation Characters, Style Variable, Truncation, Truncation Text, Visible

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

Border Color, Borders, Color Legend Title, Conditional Palette, Fill Effects, Labels, Map Drills, Palette, Show Data Range in Legend, Show Features with No Data, Size Legend Title, Style Variable, Values

Point Measure

The measure that determines the colors of points on a map chart.

Properties of Point Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Point Size Measure

The measure that determines the size of the points on a map chart.

Properties of Point Size Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Maximum Size (pt), Minimum Size (pt), Name, Rollup Aggregate Function, Style Variable

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

Angular Axis, Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Class, Conditional Palette, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Name, Notes, Padding, Pagination, Palette, Point Shape, Point Size (pt), Query, Radial Axis, Relative Alignment, Render Variable, Size & Overflow, Spider Effects, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible

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

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Chart Orientation, Class, Conditional Palette, Connecting Lines, Depth, Drill Throughs, Fill Effects, First Column Color, Floating, Font, Footer, Foreground Color, Margin, Markers, Marker Text Location, Master Detail Relationships, Name, Negative Column Color, Notes, Ordinal Axis, Padding, Pagination, Palette, Positive Column Color, Progressive Axis, Query, Relative Alignment, Render Variable, Rotate Labels, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Total Column, Value Location, Values, Visible, Visual Angle

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

Background Color, Background Image, Border, Class, Floating, Font, Foreground Color, Margin, Name, Padding, Relative Alignment, Size & Overflow, Style Variable, Type

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

Auto Group & Summarize, Auto-Sort, Avoid Division by Zero, Cross Product Allowed, Define Member Sets, Execution Method, Execution Optimization, Generated SQL/MDX, Maximum Execution Time, Maximum Rows Retrieved, Maximum Tables, Maximum Text Blob Characters, Name, Outer Join Allowed, Override Dimension Info, Processing, Rollup Processing, Suppress, Use 1.x Behavior, Use For Parameter Info, Use Local Cache, Use SQL With Clause

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

Duplicates, Name, Projection List, Set Operation

Query Reference

A reference to another query defined in the same query set.

Properties of Query Reference

Cardinality

Radar Chart

A chart that integrates multiple axes into a single radial figure as lines or stacked areas.

Properties of Radar Chart

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Class, Conditional Palette, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Name, Notes, Ordinal Axis, Padding, Pagination, Palette, Point Shape, Point Size (pt), Query, Radar Type, Radial Axis, Relative Alignment, Render Variable, Size & Overflow, Spider Effects, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible

Radial Axis

The radial numeric axis for a polar chart or radar chart, including labels, titles, range, and scale.

Properties of Radial Axis

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Radial Measure

The measure that determines the distance between the center of the chart and each data marker.

Properties of Radial Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Region Layer

A map layer that includes regions, such as provinces.

Properties of Region Layer

Border Color, Borders, Color Legend Title, Conditional Palette, Fill Effects, Labels, Map Drills, Palette, Show Data Range in Legend, Show Features with No Data, Style Variable, Values

Region Measure

The measure that determines the colors of regions on a map chart.

Properties of Region Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Regression Line

A regression line for a bubble chart or scatter chart.

Properties of Regression Line

Box Type, Line Styles, Number of Regression Lines, Polynomial Exponent, Properties, Regression Type

Repeater

A table into which you can insert items that will be repeated.

Properties of Repeater

Master Detail Relationships, Name, Pagination, Properties, Query, Render Variable, Rows Per Page

Repeater Table

Renders query data in a table.

Properties of Repeater Table

Across, Background Color, Background Image, Border, Box Type, Class, Down, Floating, Font, Foreground Color, Grouping & Sorting, Horizontal Alignment, Margin, Master Detail Relationships, Name, Pagination, Properties, Query, Relative Alignment, Render Variable, Repeater Direction, Size & Overflow, Style Variable, Table Properties, Text Flow & Justification, Visible

Repeater Table Cell

The contents of a repeater table object.

Properties of Repeater Table Cell

Background Color, Background Image, Border, Box Type, Class, Font, Foreground Color, Horizontal Alignment, Padding, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

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 Inserting Other Objects.

Properties of Rich Text Item

Aggregate Function, Data Item Label, Data Item Value, Description, Expression, HTML, HTML Source Variable, Label, Name, Name, Render Variable, Report Expression, Rollup Aggregate Function, Source Type

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

Background Color, Background Image, Baselines, Border, Border Color, Borders, Box Type, Class, Conditional Palette, Drill Throughs, Fill Effects, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Name, Notes, Padding, Pagination, Palette, Point Shape, Point Size (pt), Query, Regression Line, Relative Alignment, Render Variable, Size & Overflow, Style Variable, Subtitle, Title, Tooltips, Value Location, Values, Visible

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

Allow Is Missing, Box Type, Cascade Source, Case Insensitive, Data Format, Display Value, Floating, Hide Adornments, Multi-Select, Name, Parameter, Properties, Query, Render Variable, Required, Rows Per Page, Sorting, Static Choices, Style Variable, Use Value, Visible

Slicer Member Set

A member expression.

Properties of Slicer Member Set

Expression

SQL

An SQL query against a relational data source.

Properties of SQL

Data Source, Name, SQL, SQL Syntax

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

Expression, Scope, Usage

Table

A collection of cells in which objects can be organized in a grid fashion.

Properties of Table

Background Color, Background Image, Border, Box Type, Class, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Name, Pagination, Relative Alignment, Render Variable, Size & Overflow, Style Variable, Table Properties, Text Flow & Justification, Visible

Table Cell

The cells within a row.

Properties of Table Cell

Background Color, Background Image, Border, Box Type, Class, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

Table Row

The rows in a table.

Properties of Table Row

Background Color, Background Image, Box Type, Class, Font, Foreground Color, Horizontal Alignment, Style Variable, Vertical Alignment

Target Measure

The target measure for a metrics chart.

Properties of Target Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Text Box Prompt

A prompt control that allows you to type in a value.

Properties of Text Box Prompt

Allow Is Missing, Background Color, Border, Box Type, Default Selections, Floating, Font, Foreground Color, Hide Adornments, Hide Text, Multi-Line, Multi-Select, Name, Numbers Only, Parameter, Range, Relative Alignment, Render Variable, Required, Size & Overflow, Style Variable, Use Thousands Separator, Visible

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, Background Image, Border, Box Type, Class, Data Format, Data Item Label, Data Item Value, Drill Throughs, Expression, Floating, Font, Foreground Color, Label, Margin, Maximum Characters, Name, Name, Padding, Relative Alignment, Render Variable, Report Expression, Rollup Aggregate Function, Size & Overflow, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Use Detail Value On Page, Visible

Time Prompt

An advanced prompt control that allows you to select a time value.

Properties of Time Prompt

Allow Is Missing, Box Type, Clock Mode, Default Selections, Display Milliseconds, Display Seconds, Floating, Hide Adornments, Multi-Select, Name, Parameter, Range, Render Variable, Required, Select UI, Style Variable, Visible

Tolerance Measure

The tolerance measure for a metrics chart.

Properties of Tolerance Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, 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 Item Value, Expression, Label, Name, Properties, Report Expression, Rollup Aggregate Function, 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

Cascade Source, Default Selections, Floating, Hide Adornments, Multi-Select, Name, Parameter, Pre-populate If Parent Optional, Pre-populate Levels, Query, Render Variable, Required, Rows Per Page, Size & Overflow, Sorting, Style Variable, Use Value, Visible

Value Prompt

A prompt control that allows you to select one or more values from a list.

Properties of Value Prompt

Allow Is Missing, Auto-Submit, Background Color, Box Type, Cascade Source, Data Format, Default Selections, Display Value, Floating, Font, Foreground Color, Hide Adornments, Multi-Select, Name, Parameter, Pre-populate If Parent Optional, Properties, Query, Range, Relative Alignment, Render Variable, Required, Rows Per Page, Select UI, Size & Overflow, Sorting, Static Choices, Style Variable, Use Value, Visible

Variable

A report variable.

Properties of Variable

Name, Report Expression, Type

X Axis

The horizontal numeric axis for the chart, including labels, titles, range, and scale.

Properties of X Axis

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

X Axis

The axis line for an ordinal, or non-numeric, axis.

Properties of X Axis

Allow 45° Rotation, Allow 90° Rotation, Allow Skip, Allow Stagger, Axis Labels, Axis Line, Axis Title, Class, Display Frequency, Drill Throughs, First Label Index, Font, Foreground Color, Gridlines, Label Control, Maximum Truncation Characters, Minor Gridlines, Style Variable, Truncation, Truncation Text, Visible

X Axis Measure

The measure for the horizontal axis of a scatter chart or bubble chart.

Properties of X Axis Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Y Axis

The vertical numeric axis for the chart, including labels, titles, range, and scale.

Properties of Y Axis

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Y Axis

The axis line for an ordinal, or non-numeric, axis.

Properties of Y Axis

Allow 45° Rotation, Allow 90° Rotation, Allow Skip, Allow Stagger, Axis Labels, Axis Line, Axis Title, Class, Display Frequency, Drill Throughs, First Label Index, Font, Foreground Color, Gridlines, Label Control, Maximum Truncation Characters, Minor Gridlines, Style Variable, Truncation, Truncation Text, Visible

Y Axis 1

The numeric axis of a metrics chart, or the primary numeric axis of a combination chart.

Properties of Y Axis 1

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Y Axis 2

The secondary numeric axis of a combination chart.

Properties of Y Axis 2

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Y Axis Measure

The measure for the vertical axis of a scatter chart or bubble chart.

Properties of Y Axis Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Z Axis

The third numeric axis for a three-dimensional scatter chart, including labels, titles, range, and scale.

Properties of Z Axis

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Z Axis

The vertical numeric axis for a three-dimensional combination chart, including labels, titles, range, and scale.

Properties of Z Axis

Axis Labels, Axis Line, Axis Title, Class, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

Z Axis Measure

The measure for the third axis of a three-dimensional scatter chart.

Properties of Z Axis Measure

Aggregate Function, Custom Label, Data Format, Expression, Label, Name, Rollup Aggregate Function, Style Variable

Report Studio Properties

The following is a list of properties available in the lower left pane of Report Studio.

3-D Viewing Angle

Specifies the 3-D viewing angle of the chart.

Applies to

3-D Combination Chart, 3-D Scatter Chart

Absolute Position

Specifies that the legend will be positioned absolutely, by setting its pixel position.

Applies to

Legend

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

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. For more information about each summary function, see the Report Studio User Guide.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, HTML Item, Image, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Allow 45° Rotation

Specifies whether the labels can be rotated 45 degrees if the labels are long.

Applies to

Ordinal Axis, X Axis, Y Axis

Allow 90° Rotation

Specifies whether the labels can be rotated 90 degrees if the labels are long.

Applies to

Ordinal Axis, X Axis, Y Axis

Allow Is Missing

Allows missing values.

For example, you have a list with the columns Product line, Product type, and Quantity. For the personal accessory binoculars, no quantity exists. If you filter the report on Product line to show only personal accessories, binoculars will appear in the report.

Applies to

Date & Time Prompt, Date Prompt, Interval Prompt, Select & Search Prompt, Text Box Prompt, Time Prompt, Value Prompt

Allow Skip

Specifies whether some labels can be skipped if they are long.

Applies to

Ordinal Axis, X Axis, Y Axis

Allow Stagger

Specifies whether the labels can be staggered if they are long.

Applies to

Ordinal Axis, X Axis, Y Axis

Angular Axis

Specifies whether the axis is rendered.

Applies to

Polar Chart

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

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

Auto-Sort

When running the report, specifies whether to automatically sort based on data type.

Applies to

Query

Auto-Submit

Specifies whether the application submits the prompt page automatically, as soon as a value is changed.

Applies to

Value Prompt

Auto Truncation

Specifies whether to allow truncation of text.

Applies to

Legend

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

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

Axis Assignment

Specifies the numeric axis to use.

Applies to

Area, Bar, Baseline, Line, Marker

Axis Assignment

Specifies which numeric axis to use.

Applies to

Baseline

Axis Labels

Specifies whether to show or hide axis labels.

Applies to

Angular Axis, Cumulation Line Axis, Numerical Axis, Numerical Axis, Ordinal Axis, Radial Axis, X Axis, X Axis, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Axis Line

Specifies the properties of the axis line in a chart.

Applies to

Angular Axis, Cumulation Line Axis, Numerical Axis, Numerical Axis, Ordinal Axis, Radial Axis, X Axis, X Axis, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Axis Title

Specifies whether an axis title will be rendered.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Chart, Map, Numerical Axis, Numerical Axis, Ordinal Axis, Pie Chart, Radial Axis, X Axis, X Axis, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Background Color

Specifies the background color for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Chart Body, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, Legend, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Note Content, Page, Page Body, Page Footer, Page Header, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Table Row, Text Box Prompt, Text Item, Value Prompt

Background Image

Specifies an image to be used as the background for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Chart Body, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, Legend, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Page, Page Body, Page Footer, Page Header, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Table Row, Text Item

Bar, Line, or Area Index

Specifies which combination object to use when calculating the position.

Applies to

Baseline, Baseline, Marker

Baselines

Specifies baselines.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Metrics Range Chart, Pareto Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Block Variable

Specifies a variable based on which the block can be conditionally rendered.

Applies to

Conditional Block, Conditional Block

Border

Specifies the width, style, and color for the border of the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Page, Page Body, Page Footer, Page Header, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Text Box Prompt, Text Item

Border Color

Specifies the color of the border.

Applies to

3-D Area, 3-D Bar, 3-D Line, 3-D Scatter Chart, Area, Bar, Bubble Chart, Display Layer, Gauge Chart, Legend, Line, Pareto Chart, Pie Chart, Point Layer, Polar Chart, Progressive Chart, Radar Chart, Region Layer, Scatter Chart

Borders

Specifies whether borders are rendered.

Applies to

3-D Area, 3-D Bar, 3-D Line, 3-D Scatter Chart, Area, Bar, Bubble Chart, Display Layer, Gauge Chart, Legend, Line, Pareto Chart, Pie Chart, Point Layer, Polar Chart, Progressive Chart, Radar Chart, Region Layer, Scatter Chart

Bottom Position (px)

Specifies the bottom position of the note, in pixels.

Applies to

Note

Bottom Position (px)

Specifies the pixel position of the bottom edge of the legend.

Applies to

Legend

Box Type

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. 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

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Field Set, Gauge Chart, Hyperlink, Image, List, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater Table, Scatter Chart, Table, Text Item

Box Type

Specifies whether to override the default box type for the object.

Applies to

Axis Title, Baseline, Baseline, Baseline, Baseline, Chart Footer, Chart Subtitle, Chart Title, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Date & Time Prompt, Date Prompt, Interval Prompt, Legend, Legend Title, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Marker, Marker, Note, Note Content, Page Body, Page Footer, Page Header, Regression Line, Repeater Table Cell, Select & Search Prompt, Table Cell, Table Row, Text Box Prompt, Time Prompt, Value Prompt

Calculation Intersection

Specifies that if an edge member intersects with a calculated member, the resulting cell value should be N/A.

Applies to

Data Item

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, Date Prompt

Caption

Specifies the caption for the level.

Applies to

Level

Caption

Specifies the caption.

Applies to

Calculated Member

Cardinality

Specifies the cardinality for this join operand.

Applies to

Query Reference

Cascade Source

Specifies the parameter whose value is used to filter the values displayed in this control.

Applies to

Select & Search Prompt, Tree Prompt, Value Prompt

Case Insensitive

Specifies whether to perform a case insensitive search by default.

Applies to

Select & Search Prompt

Catalog

Specifies the OLAP catalog.

Applies to

MDX

Chart Orientation

Specifies whether the chart is rendered vertically or horizontally.

Applies to

Combination Chart, Pareto Chart, Progressive Chart

Chart Type

Specifies whether the data may be rendered as either a bar, line, or area.

Applies to

Area, Bar, Line

Chart Type

Specifies whether the data may be rendered as either a bar, line or area.

Applies to

3-D Area, 3-D Bar, 3-D Line

Class

Specifies a class to apply to the object. The Class provides a default style.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Axis Title, Block, Bubble Chart, Caption, Chart Footer, Chart Subtitle, Chart Title, Combination Chart, Conditional Block, Conditional Block, Crosstab, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Axis, Field Set, Gauge Chart, Gauge Labels, Hyperlink, Hyperlink Button, Image, Legend, Legend Title, List, List Cell, List Column Body, List Column Title, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page, Page Body, Page Footer, Page Header, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Radial Axis, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Table Row, Text Item, X Axis, X Axis, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Clock Mode

Specifies whether the arms of the clock move.

Applies to

Date & Time Prompt, Time Prompt

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

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

Column Intersection

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, Crosstab Member Fact Cells

Column Titles

Specifies where or whether column titles may be rendered.

Applies to

List

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

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

Conditional Palette

Specifies a conditional palette for the chart.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Metrics Range Chart, Pareto Chart, Pie Chart, Point Layer, Polar Chart, Progressive Chart, Radar Chart, Region Layer, Scatter Chart

Connecting Lines

Specifies the properties of the lines that connect the segments of a stacked bar. This property is ignored for clustered bars.

Applies to

Bar, Pareto Chart, Progressive Chart

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

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

Cumulation Axis

Specifies whether the cumulation axis may be rendered.

Applies to

Cumulation Line

Cumulation Label

Specifies whether a label for the cumulation line may be rendered.

Applies to

Cumulation Line

Cumulative Line

Specifies whether the cumulation line is rendered.

Applies to

Pareto Chart

Current Block

Specifies which block is currently being authored.

Applies to

Conditional Block, Conditional Block

Custom Label

Overrides the default label for the data item.

Applies to

Angular Measure, Bubble Measure, Chart Node Member, Default Measure, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Point Measure, Point Size Measure, Radial Measure, Region Measure, Target Measure, Tolerance Measure, X Axis Measure, Y Axis Measure, Z Axis Measure

Data Format

Specifies the data format of the object.

Applies to

Angular Axis, Angular Measure, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Columns, Crosstab Columns, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Cumulation Line Axis, Default Measure, Gauge Numerical Axis, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Numerical Axis, Numerical Axis, Point Measure, Point Size Measure, Radial Axis, Radial Measure, Region Measure, Select & Search Prompt, Target Measure, Text Item, Tolerance Measure, Value Prompt, X Axis, X Axis Measure, Y Axis, Y Axis 1, Y Axis 2, Y Axis Measure, Z Axis, Z Axis, Z Axis Measure

Data Item

Specifies a reference to a data item. You cannot modify this value.

Applies to

Crosstab Node Member, Member Set

Data Item

Specifies a reference to a data item.

Applies to

Calculated Member, Fact, Key, Member Property

Data Item Label

Specifies the data item label that defines the text to render.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item

Data Item Label

Specifies the data item label that defines the HTML to render.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

Data Item Label

Specifies the data item label that defines the URL.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

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, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Data Item Value

Specifies the data item value that defines the text to render.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item, Total Column

Data Item Value

Specifies the data item value that defines the HTML to render.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

Data Item Value

Specifies the data item value that defines the URL.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

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, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Data Item Value

Specifies the numeric position by using a data item value.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker

Data Item Value

Specifies the numeric position of a data item.

Applies to

Marker

Data Item Value

Specifies the numeric position from a data item.

Applies to

Marker

Data Language

Specifies the language of the data.

Applies to

Map

Data Source

Specifies the query data source.

Applies to

MDX, SQL

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

Default Measure 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

Crosstab

Default Selections

Specifies the collection of default selections for a prompt control.

Applies to

Date & Time Prompt, Date Prompt, Interval Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

Default Title

Specifies whether the default title may be generated.

Applies to

Axis Title, Legend Title

Define Contents

Overrides the content of the selected crosstab intersection.

Applies to

Crosstab Intersection, Crosstab Member Fact Cells

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

Depth

Specifies the depth of the chart. A value of zero indicates a flat chart.

Applies to

Combination Chart, Metrics Range Chart, Pareto Chart, Pie Chart, Progressive Chart

Description

Specifies a description for the object, that is used to assist authoring.

Applies to

Class, HTML Item, Metric Studio Diagram, Rich Text Item

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

Data Item

Diagram Identifier

Identifies a diagram in Metric Studio.

Applies to

Metric Studio Diagram

Dial Outline Color

Specifies the dial outline color in a gauge chart.

Applies to

Gauge Chart

Dictionary

Specifies the aliases to use when matching data values to feature names in the map.

Applies to

Map

Display After Overall Header

Specifies whether the list page header is to be rendered after the overall header.

Applies to

List Page Header

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.

Applies to

Ordinal Axis, X Axis, Y Axis

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, Interval Prompt, Time Prompt

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, Interval Prompt, Time Prompt

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, Value Prompt

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

Drill Throughs

Specifies a drill-through target for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Gauge Chart, Gauge Labels, Image, Legend, List Column Body, List Column Title, Map, Metrics Range Chart, Ordinal Axis, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart, Text Item, X Axis, Y Axis

Duplicates

Specifies whether duplicate rows will be preserved or removed.

Applies to

Query Operation

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

Execution Method

Specifies whether the query is allowed to run concurrently or must run sequentially. Concurrent query execution may improve performance in some cases.

Applies to

Query

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

Expression

Specifies the expression to evaluate when filtering the data.

Applies to

Detail Filter, Summary Filter

Expression

Specifies the expression that defines the slicer member.

Applies to

Slicer Member Set

Expression

Specifies the expression used to populate the data item.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, HTML Item, Image, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Face Color

Specifies the color to show on the face of each gauge in a gauge chart.

Applies to

Gauge Chart

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

Fill Effects

Specifies the fill effects for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Chart Body, Combination Chart, Display Layer, Gauge Chart, Legend, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Point Layer, Polar Chart, Progressive Chart, Radar Chart, Region Layer, Scatter Chart

First Column Color

Specifies the color, gradient, or pattern to be used for the first column in the progressive chart.

Applies to

Progressive Chart

First Date

Specifies the earliest date to render in the control, and the earliest date that can be selected. The date entered must be in YYYY-MM-DD format.

Applies to

Date & Time Prompt, Date Prompt

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.

Applies to

Ordinal Axis, X Axis, Y Axis

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

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Date & Time Prompt, Date Prompt, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, Interval Prompt, List, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Scatter Chart, Select & Search Prompt, Table, Text Box Prompt, Text Item, Time Prompt, Tree Prompt, Value Prompt

Font

Specifies the font family, size, weight, style, and effects used to display the object's text.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Axis Title, Block, Bubble Chart, Caption, Chart Body, Chart Footer, Chart Subtitle, Chart Title, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Cumulation Line Axis, Field Set, Gauge Chart, Gauge Labels, Gauge Numerical Axis, Hyperlink, Legend, Legend Title, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page, Page Body, Page Footer, Page Header, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Radial Axis, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Table Row, Text Box Prompt, Text Item, Value Prompt, X Axis, X Axis, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Font Auto-Sizing

Specifies whether to allow automatic resizing of the font.

Applies to

Legend

Footer

Specifies whether a chart footer is rendered.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Foreground Color

Specifies the color of the object's text.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Axis Title, Block, Bubble Chart, Caption, Chart Body, Chart Footer, Chart Subtitle, Chart Title, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Cumulation Line Axis, Field Set, Gauge Chart, Gauge Labels, Gauge Numerical Axis, Hyperlink, Hyperlink Button, Legend, Legend Title, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page, Page Body, Page Footer, Page Header, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Radial Axis, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Table Row, Text Box Prompt, Text Item, Value Prompt, X Axis, X Axis, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Gauge Labels

Specifies whether gauge labels are rendered.

Applies to

Gauge Chart

Gauge Palette

Specifies the palette that controls the look of the dial portion of a gauge.

Applies to

Gauge Chart

Generated SQL/MDX

Shows the generated SQL or MDX.

Applies to

Query

Gridlines

Specifies the properties of the gridlines in a chart.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Ordinal Axis, Radial Axis, X Axis, X Axis, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Grouping & Sorting

Specifies the grouping and sorting structure.

Applies to

List, Page Set, Repeater Table

Grouping Type

Specifies whether the absolute, stacked, or 100 percent stacked data may be drawn.

Applies to

Area, Bar, Line

Group Span

Specifies the group that this cell should visually span.

Applies to

List Column Body

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

Height (px)

Specifies the height of the note, in pixels.

Applies to

Note

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, Date Prompt, Generated Prompt, Interval Prompt, Select & Search Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

Hide Text

Specifies whether to replace characters entered in the prompt control with asterisk (*) characters.

Applies to

Text Box Prompt

Hole Size (%)

Specifies the size of the hole in a donut chart. A value of zero indicates a pie chart.

Applies to

Pie Chart

Horizontal Alignment

Specifies how the contents of the selected object is aligned horizontally.

Applies to

Block, Chart Body, Chart Footer, Chart Subtitle, Chart Title, Class, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Field Set, Gauge Labels, Legend, Legend Title, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Note Content, Page, Page Body, Page Footer, Page Header, Pie Labels, Repeater Table, Repeater Table Cell, Table, Table Cell, Table Row

HTML

Specifies the static text used as HTML.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

HTML Source Variable

Specifies a variable based on which the HTML source is chosen.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

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

Include Zero For Auto Scale

Specifies whether the value zero is included in the automatic calculation of the numeric scale. It is only relevant if the chart is involved in a master detail relationship.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Indentation Length

If specified, the members are indented according to their level in the hierarchy. The string value represents a CSS length (for example, 10 px, 0.5 in, etc.). If not specified, then the members are rendered in a flat list with no indenting.

Applies to

Crosstab Node Member

Indentation Start Level

If specified, indentation of members will only start at the indicated level member. The default level (usually All) is at index zero (0).

Applies to

Crosstab Node Member

Intersection Name

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, Crosstab Space

Join Relationships

Specifies how to join the two queries.

Applies to

Join

Label

Specifies the static text that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to

Bookmark, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Label

Specifies the class label for a local class.

Applies to

Class

Label

Specifies the class label for a global class. You cannot modify this label.

Applies to

Class

Label

Specifies whether a label is rendered for the baseline.

Applies to

Baseline, Baseline

Label

Specifies whether a label is rendered for the marker.

Applies to

Marker, Marker

Label

Specifies the label of the object.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, HTML Item, Image, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

Label Control

Controls how the labels in a chart are rendered.

Applies to

Ordinal Axis, X Axis, Y Axis

Labels

Specifies whether labels are rendered in the chart.

Applies to

Display Layer, Point Layer, Region Layer

Labels

Specifies whether labels are rendered.

Applies to

Pie Chart

Last Date

Specifies the latest date rendered in the control, and the last date that can be selected. The date entered must be in YYYY-MM-DD format.

Applies to

Date & Time Prompt, Date Prompt

Left Position (px)

Specifies the left position of the note, in pixels.

The position is measured from the left edge of the note.

Applies to

Note

Left Position (px)

Specifies the pixel position of the left edge of the legend.

Applies to

Legend

Legend

Specifies whether the legend is rendered.

Applies to

Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Radar Chart, Scatter Chart

Legend Label

Specifies whether to render the baseline in the legend.

Applies to

Baseline, Baseline

Legend Title

Specifies whether a legend title is rendered.

Applies to

Legend

Limit Type

Specifies whether the limit is the maximum or minimum.

Applies to

Baseline, Baseline, Baseline, Baseline

Line

Specifies whether a line may be rendered. This allows you to show markers without lines.

Applies to

Line

Line Style

Specifies the style of the line.

Applies to

Line

Line Styles

Specifies the line style, color, and weight of the line.

Applies to

Baseline, Baseline, Baseline, Baseline, Cumulation Line, Regression Line

Line Weight (pt)

Specifies the line thickness in points. A value of zero indicates the thinnest possible line.

Applies to

Line

Map & Layers

Sets the map and layers for a map chart.

Applies to

Map

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, Region Layer

Margin

Specifies the margin properties for the object.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, List, Map, Metrics Range Chart, Page, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Scatter Chart, Table, Text Item

Marker Color

Specifies a color for the target value markers in a metrics chart.

Applies to

Metrics Range Chart

Marker Label

Specifies whether the label for the Target Marker will appear in the legend.

Applies to

Metrics Range Chart

Markers

Specifies markers.

Applies to

Bubble Chart, Combination Chart, Metrics Range Chart, Pareto Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Marker Text Location

Specifies where the text of the marker is rendered.

Applies to

Bubble Chart, Combination Chart, Metrics Range Chart, Pareto Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Master Detail Relationships

Specifies how a data container, or detail, is rendered inside another data container, or master.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Axis Title, Baseline, Baseline, Baseline, Baseline, Bubble Chart, Chart Footer, Chart Subtitle, Chart Title, Combination Chart, Crosstab, Gauge Chart, Legend Title, List, Map, Marker, Marker, Metrics Range Chart, Note Content, Page, Page Set, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Scatter Chart

Maximum Characters

Specifies the maximum number of characters to show before the text is truncated.

Applies to

Legend, Text Item

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 the value 0 is specified, no error is returned and the query runs until complete.

Applies to

Query

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 the value 0 is specified, no error is returned and the query returns all rows.

Applies to

Query

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

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 Cognos SQL exceeds the specified value. If the value 0 is specified, no error is returned and there is no restriction on the number of tables that can be queried.

Applies to

Query

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 the value 0 is specified, no error is returned and text BLOBs can be of any size.

Applies to

Query

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 this property, the application will automatically determine 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

Gauge Labels, Ordinal Axis, Pie Labels, X Axis, Y Axis

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

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

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

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.

Applies to

Baseline

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

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

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Minor Gridlines

Specifies the properties of the minor gridlines in a chart.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Ordinal Axis, Radial Axis, X Axis, X Axis, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Multi-Line

Specifies whether to allow multi-line editing in the text control

Applies to

Text Box Prompt

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

Date & Time Prompt, Date Prompt, Interval Prompt, Select & Search Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

Name

Specifies the name of the object.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, Dimension, Fact, HTML Item, Image, Level, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Member Set, Page, Page Set, Point Measure, Point Size Measure, Query, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, Variable, X Axis Measure, Y Axis Measure, Z Axis Measure

Name

Specifies a unique name that allows layout objects to be reused, usually to take advantage of any applied styling.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Combination Chart, Crosstab, Date & Time Prompt, Date Prompt, Field Set, Gauge Chart, Generated Prompt, HTML Item, Hyperlink, Hyperlink Button, Image, Interval Prompt, List, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater, Repeater Table, Rich Text Item, Scatter Chart, Select & Search Prompt, Table, Text Box Prompt, Text Item, Time Prompt, Tree Prompt, Value Prompt

Name

Specifies the unique name identifier for a query operation.

Applies to

Query Operation

Name

Specifies the unique name identifier for an SQL object.

Applies to

SQL

Name

Specifies the unique name identifier for an MDX object.

Applies to

MDX

Name

Specifies a unique name that allows layout objects to be reused, usually to take advantage of any applied formatting.

Applies to

Calculated Member, Key, Level Hierarchy, Member Hierarchy, Member Property

Negative Column Color

Specifies the color, gradient, or pattern to be used for columns in the progressive chart that represent negative values.

Applies to

Progressive Chart

No Data Features Size (pt)

Specifies the point size used for map point features that do not have corresponding data.

Applies to

Map

Note Border

Specifies the properties for the border of a note.

Applies to

Note

Notes

Specifies notes.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

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

Numbers Only

Specifies whether the Text Box Prompt allows numbers only.

Applies to

Text Box Prompt

Numerical Axis

Specifies whether the axis is rendered.

Applies to

3-D Combination Chart

Numerical Axis

Specifies whether the numeric axis of a gauge chart is rendered.

Applies to

Gauge Chart

Numeric Value

Specifies the numeric position by using a number.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker

Numeric Value

Specifies the value of the numeric position.

Applies to

Marker

Ordinal Axis

Specifies whether the axis is rendered.

Applies to

Combination Chart, Metrics Range Chart, Progressive Chart, Radar Chart

Outer Join Allowed

Specifies whether outer joins are allowed on the object.

Applies to

Query

Override Dimension Info

Specifies dimension information for a query.

Applies to

Query

Overrides

Specifies whether to override child objects. Before you can override child objects, they must have a name.

Applies to

Layout Component Reference

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

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Gauge Chart, Hyperlink, Hyperlink Button, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Page, Page Body, Page Footer, Page Header, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table Cell, Scatter Chart, Table Cell, Text Item

Pagination

Specifies pagination rules, such as page breaks, keep-with properties, page counts, and numbering.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Crosstab, Crosstab Node Member, Crosstab Space, Gauge Chart, List, List Header, Map, Metrics Range Chart, Page, Page Set, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Scatter Chart, Table

Palette

Specifies the palette to use for the chart.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

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, Region Layer

Parameter

Specifies the parameter that is satisfied by values chosen in the prompt control.

Applies to

Date & Time Prompt, Date Prompt, Generated Prompt, Interval Prompt, Select & Search Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

Percentile

Specifies a position based on a data percentile value. This value must be greater than zero.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker, Marker

Percent of Axis

Specifies a position based on a percentage along the numeric axis. This value must be greater than zero.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker, Marker

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

Pie Labels

Specifies whether pie labels are rendered.

Applies to

Pie Chart

Point Color

Specifies the color of the markers.

Applies to

Marker, Marker

Point Shape

Specifies the shape of the markers.

Applies to

Cumulation Line, Marker, Marker

Point Shape

Specifies the shape of the markers. If you choose a value of series or category, the marker shape varies accordingly.

Applies to

3-D Scatter Chart, Bubble Chart, Line, Polar Chart, Radar Chart, Scatter Chart

Point Size (pt)

Specifies the size of markers in points. A value of zero means do not show markers.

Applies to

3-D Scatter Chart, Cumulation Line, Line, Marker, Marker, Polar Chart, Radar Chart, Scatter Chart

Polynomial Exponent

Specifies the highest exponential value to use in the regression calculation.

Applies to

Regression Line

Position

Specifies where to position the legend.

Applies to

Legend

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

Pre-populate If Parent Optional

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, Value Prompt

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

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

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.

Applies to

Query

Progressive Axis

Specifies whether the axis is rendered.

Applies to

Progressive Chart

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

Properties

Specifies a list of properties for an object.

Applies to

Axis Title, Baseline, Baseline, Baseline, Baseline, Chart Footer, Chart Node Member, Chart Subtitle, Chart Title, Crosstab Node Member, Cumulation Line, Legend Title, List, List Footer, List Header, Marker, Marker, Note Content, Page, Page Set, Regression Line, Repeater, Repeater Table, Select & Search Prompt, Total Column, Total Column Label, Value Prompt

Push To Bottom

Specifies whether to position the footer as low as possible inside the parent object.

Applies to

List Footer, List Page Footer

Query

Specifies a reference to a query.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Axis Title, Baseline, Baseline, Baseline, Baseline, Bubble Chart, Chart Footer, Chart Subtitle, Chart Title, Combination Chart, Crosstab, Gauge Chart, Legend Title, List, Map, Marker, Marker, Metrics Range Chart, Note Content, Page, Page Set, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Scatter Chart, Select & Search Prompt, Tree Prompt, Value Prompt

Radar Type

Specifies how the radar chart is rendered.

Applies to

Radar Chart

Radial Axis

Specifies whether the axis is rendered.

Applies to

Polar Chart, Radar Chart

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, Date Prompt, Interval Prompt, Text Box Prompt, Time Prompt, Value Prompt

Range Label

Specifies whether the label for the Target Range will appear in the legend.

Applies to

Metrics Range Chart

Regression Line

Specifies whether a regression line is rendered.

Applies to

Bubble Chart, Scatter Chart

Regression Type

Specifies the type of regression used.

Applies to

Regression Line

Relative Alignment

Specifies how to vertically align this object, relative to its siblings.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Class, Combination Chart, Crosstab, Date Prompt, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, List, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Scatter Chart, Table, Text Box Prompt, Text Item, Value Prompt

Render Variable

Specifies a variable based on which the object can be conditionally rendered.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Combination Chart, Crosstab, Date & Time Prompt, Date Prompt, Field Set, Gauge Chart, Generated Prompt, HTML Item, Hyperlink, Hyperlink Button, Image, Interval Prompt, List, List Column, Map, Metrics Range Chart, Page, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Repeater Table Cell, Rich Text Item, Scatter Chart, Select & Search Prompt, Table, Text Box Prompt, Text Item, Time Prompt, Tree Prompt, Value Prompt

Repeater Direction

Specifies the direction in which to populate the rendered repeater cells.

Applies to

Repeater Table

Report Expression

Specifies the report expression that defines the text to render.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item, Total Column

Report Expression

Specifies the report expression used to define the HTML to render.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

Report Expression

Specifies the report expression that defines the URL.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

Report Expression

Specifies the report expression that defines the bookmark. The value used as the bookmark reference must match this value.

Applies to

Bookmark, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Report Expression

Specifies the numeric position by using a report expression.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker

Report Expression

Specifies the report expression for the member position.

Applies to

Baseline, Marker

Report Expression

Specifies a report expression.

Applies to

As of Time Expression

Report Expression

Specifies the expression to evaluate when determining the value for this variable.

Applies to

Variable

Report Expression

Specifies the numeric position from a report expression.

Applies to

Marker

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 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, check the setting of the associated filter for the Required property.

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

Date & Time Prompt, Date Prompt, Generated Prompt, Interval Prompt, Select & Search Prompt, Text Box Prompt, Time Prompt, Tree Prompt, Value Prompt

Right Position (px)

Specifies the pixel position of the right edge of the legend.

Applies to

Legend

Rollup Aggregate Function

Specifies the type of aggregation to apply to summarized values. These values appear at the higher levels of lists and crosstabs.

Applies to

Angular Measure, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, HTML Item, Image, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Target Measure, Text Item, Tolerance Measure, Total Column, X Axis Measure, Y Axis Measure, Z Axis Measure

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

Rotate Labels

Controls whether or not labels are rotated with the chart. May help make labels easier to read on rotated charts.

Applies to

Combination Chart, Pareto Chart, Progressive Chart

Row Intersection

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, Crosstab Member Fact Cells

Rows Per Page

Specifies the maximum number of rows to show at one time.

Applies to

Crosstab, List, Repeater, Select & Search Prompt, Tree Prompt, Value Prompt

Scale

Specifies whether the numeric scale is logarithmic or linear.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Scale Interval

Specifies the interval between ticks on the numeric scale. This value must be greater than zero. If no value is specified, one will be calculated based on the data.

Applies to

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Scope

Specifies the scope of the filter in terms of the number of levels.

Applies to

Summary Filter

Selector

Specifies the class name in the report specification. We recommend that you do not modify this name.

Applies to

Class

Selector

Specifies the class name for a global class. You cannot modify this name.

Applies to

Class

Select UI

Specifies which interface the prompt control renders.

Applies to

Date & Time Prompt, Date Prompt, Time Prompt, Value Prompt

Separator

Specifies a separator to use when showing legend entries.

Applies to

Legend

Series Color

Specifies whether the graphs for a series will have the same color for each combination measure.

Applies to

3-D Combination Chart, Combination Chart, Metrics Range Chart

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

Show Caption

Specifies whether, or where, to show the caption.

Applies to

Field Set

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, Region Layer

Show Features with No Data

Indicates whether to show the features of a map that do not have corresponding data.

Applies to

Point Layer, Region Layer

Show Feelers

Specifies whether feeler lines are rendered for each marker.

Applies to

3-D Scatter Chart

Show Legend Values

Specifies whether and how to show legend values.

Applies to

Legend

Size & Overflow

Specifies the height and width of the object, as well as the treatment of page overflow, using scroll bars and clipping.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Metrics Range Chart, Page, Page Body, Page Footer, Page Header, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Repeater Table Cell, Scatter Chart, Table, Table Cell, Text Box Prompt, Text Item, Tree Prompt, Value Prompt

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

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

Data Item

Sorting

Specifies the desired sort sequence.

Applies to

Chart Node Member, Crosstab Node Member, Level, Select & Search Prompt, Tree Prompt, Value Prompt

Source Type

Specifies the source type of the text.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item

Source Type

Specifies the source type of the HTML text.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

Source Type

Specifies the source type of the URL.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

Source Type

Specifies the source type of the bookmark.

Applies to

Bookmark, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

Source Type

Specifies the source type for the numeric position.

Applies to

Baseline, Baseline, Baseline, Baseline, Marker

Source Type

Specifies the type of numeric position.

Applies to

Marker

Source Type

Specifies a type of numeric position.

Applies to

Marker

Spacing & Breaking

Specifies text properties such as line height, letter spacing, and word breaking.

Applies to

Block, Class, Conditional Block, Conditional Block, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Hyperlink, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Page Body, Page Footer, Page Header, Repeater Table Cell, Table Cell, Text Item

Spider Effects

Specifies whether the chart is rendered with spider effects.

Applies to

Polar Chart, Radar Chart

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

SQL Syntax

Specifies the syntax of the SQL in the query. A value of Cognos indicates that Cognos extended SQL-92 syntax is used. Alternatively, a value of Native indicates that native database SQL is used.

Applies to

SQL

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, Baseline, Baseline, Baseline, Marker, Marker

Static Choices

Represents a collection of static choices used by the prompt object.

Applies to

Select & Search Prompt, Value Prompt

Style Variable

Specifies a variable based on which the object can be conditionally styled.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Angular Measure, Axis Title, Block, Bubble Chart, Bubble Measure, Caption, Chart Body, Chart Footer, Chart Subtitle, Chart Text Item, Chart Title, Combination Chart, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Cumulation Line Axis, Date & Time Prompt, Date Prompt, Default Measure, Display Layer, Field Set, Gauge Chart, Gauge Labels, Gauge Numerical Axis, Hyperlink, Hyperlink Button, Image, Interval Prompt, Legend, Legend Title, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Map, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Metrics Range Chart, Note, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Page, Page Body, Page Footer, Page Header, Pareto Chart, Pie Chart, Pie Labels, Point Layer, Point Measure, Point Size Measure, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Radial Axis, Radial Measure, Region Layer, Region Measure, Repeater Table, Repeater Table Cell, Scatter Chart, Select & Search Prompt, Table, Table Cell, Table Row, Target Measure, Text Box Prompt, Text Item, Time Prompt, Tolerance Measure, Tree Prompt, Value Prompt, X Axis, X Axis, X Axis Measure, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Y Axis Measure, Z Axis, Z Axis, Z Axis Measure

Subtitle

Specifies whether a chart subtitle is rendered.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Suppress

For SAP BW data sources, specifies the type of suppression to apply to the query results. This property overrides the corresponding governor in the model. If unspecified, the value of the governor in the model is used.

Applies to

Query

Table Properties

Specifies the properties for the table object.

Applies to

Class, Crosstab, List, Repeater Table, Table

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

Target Marker

Specifies whether the status indicators will appear in the legend.

Applies to

Metrics Range Chart

Target Marker Border Color

Specifies a color for the borders around target value markers in a metrics chart.

Applies to

Metrics Range Chart

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

Target Range (%)

Specifies target ranges centered around target measure values.

Applies to

Metrics Range Chart

Text

Specifies the static text to render.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item, Total Column

Text Flow & Justification

Specifies text flow properties, such as direction, writing mode, and justification.

Applies to

Block, Class, Conditional Block, Conditional Block, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Hyperlink, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Page, Page Body, Page Footer, Page Header, Repeater Table, Repeater Table Cell, Table, Table Cell, Text Item

Text Source Variable

Specifies a variable based on which the text source can be chosen.

Applies to

Chart Node Member, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item

Title

Specifies whether a chart title is rendered.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

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

Tolerance Label

Specifies whether the label for the Target Tolerance will appear in the legend.

Applies to

Metrics Range Chart

Tooltips

Specifies whether tooltips are rendered in the chart. Tooltips are not supported in PDF documents.

Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Combination Chart, Gauge Chart, Map, Metrics Range Chart, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Top Position (px)

Specifies the pixel position of the top edge of the legend.

Applies to

Legend

Total Column

Specifies whether a total column is rendered.

Applies to

Progressive Chart

Total Column Color

Represents the color, gradient, or pattern of the total column on the progressive chart.

Applies to

Total Column

Truncation

Specifies whether labels can be truncated.

Applies to

Gauge Labels, Ordinal Axis, Pie Labels, X Axis, Y Axis

Truncation Text

Specifies the text to append when a label is truncated.

Applies to

Gauge Labels, Legend, Ordinal Axis, Pie Labels, X Axis, Y Axis

Type

Specifies the type of variable.

Applies to

Variable

Type

Specifies the behavior of the prompt control.

Applies to

Prompt Button

Upper Range Skew (%)

Specifies a percentage that affects the positioning of tolerance bar and range with respect to the target value.

Applies to

Metrics Range Chart

URL

Specifies the URL, using static text.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

URL Source Variable

Specifies a variable based on which the URL source can be chosen.

Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

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, Summary Filter

Use 1.x Behavior

Specifies that Cognos ReportNet query semantics are used if they differ from Cognos 8 query rules.

Applies to

Query

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 attribute 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.

Applies to

Text Item

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.

Applies to

Query

Use Local Cache

Specifies whether a query is a candidate for query reuse. If set to Yes, the query engine can reuse an existing SQL result. If set to No, the query is executed rather than using cached results.

Applies to

Query

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

Angular Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

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

Use Thousands Separator

Specifies whether to delimit digit groups with the thousands separator.

Applies to

Text Box Prompt

Use Value

Specifies the values used by the prompt object.

Applies to

Tree Prompt

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, Value Prompt

Value Location

Specifies where values and labels are to be rendered in the chart.

Applies to

Area, Bar, Bubble Chart, Combination Chart, Cumulation Line, Line, Metrics Range Chart, Pareto Chart, Polar Chart, Progressive Chart, Radar Chart, Scatter Chart

Value Representation

Specifies whether values are rendered as percentages.

Applies to

Pie Chart

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, Bubble Chart, Polar Chart, Scatter Chart

Values

Specifies whether values are rendered in the chart.

Applies to

3-D Area, 3-D Bar, 3-D Line, Area, Bar, Cumulation Line, Line, Pareto Chart, Point Layer, Progressive Chart, Radar Chart, Region Layer

Values

Specifies whether values are rendered.

Applies to

Pie Chart

Value Type

Specifies whether absolute values are rendered rather than cumulative values.

Applies to

Area, Bar, Line

Vertical Alignment

Specifies how objects contained in this object are vertically aligned.

Applies to

Class, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Page Body, Page Footer, Page Header, Repeater Table Cell, Table Cell, Table Row

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

3-D Combination Chart, 3-D Scatter Chart, Angular Axis, Axis Title, Block, Bubble Chart, Chart Footer, Chart Subtitle, Chart Title, Class, Combination Chart, Conditional Block, Conditional Block, Crosstab, Cumulation Line Axis, Date & Time Prompt, Date Prompt, Field Set, Gauge Chart, Gauge Labels, Gauge Numerical Axis, Hyperlink, Image, Interval Prompt, Legend, Legend Title, List, Map, Metrics Range Chart, Note Content, Numerical Axis, Numerical Axis, Ordinal Axis, Pareto Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Radar Chart, Radial Axis, Repeater Table, Scatter Chart, Select & Search Prompt, Table, Text Box Prompt, Text Item, Time Prompt, Tree Prompt, Value Prompt, X Axis, X Axis, Y Axis, Y Axis, Y Axis 1, Y Axis 2, Z Axis, Z Axis

Visual Angle

Specifies the angle, in degrees, in which the chart objects will be displayed when the chart has 3-D effects.

Applies to

Combination Chart, Metrics Range Chart, Pareto Chart, Progressive Chart

White Space

Specifies how white space inside the object is handled.

Applies to

Block, Class, Conditional Block, Conditional Block, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Page Body, Page Footer, Page Header, Repeater Table Cell, Table Cell

Width (px)

Specifies the width of the note, in pixels.

Applies to

Note

X Axis

Specifies whether the axis is rendered.

Applies to

3-D Combination Chart

Y1 Axis

Specifies whether the axis is rendered.

Applies to

Combination Chart, Metrics Range Chart

Y2 Axis

Specifies whether the axis is rendered.

Applies to

Combination Chart

Y2 Axis Position

Specifies how the second Y axis is rendered.

Applies to

Combination Chart

Y Axis

Specifies whether the axis is rendered.

Applies to

3-D Combination Chart

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. If no value is entered for this property, an empty string will be displayed. 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 #!Error. 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 sign. 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.

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 notations, using exponents. If this property is set to No, no scientific notations will be used. If it is not specified, scientific notations 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 Months

Specifies whether to display the month.

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 #!Overflow. 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.

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 formatted according 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: 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.

Number of symbols	Meaning	Example
4 or more	Full text form	EEEE produces Monday
Less than 4	Abbreviated form	EEE produces Mon

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 1997, and yy produces 97.

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.

Number of symbols	Meaning	Example
3 or more	Text	MMMM produces January MMM produces Jan

Number of symbols	Meaning	Example
Less than 3	Numbers	MM produces 01 M produces 1

Date and Time Symbols

Locale ID	Locale group
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-za, fo-fo, gl-es, id, id-id, is, is-is, it, it-ch, it-it, kk-kz, ms, ms-bn, 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, sw-ke, sw-tz	Locale Group A
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	Locale Group B
ca-es, cs, cs-cz, da, da-dk, es, es-ar, es-bo, es-cl, es-co, es-cr, es-do, es-ec, es-es, es-gt, es-hn, es-mx, es-ni, es-pa, es-pe, es-pr, es-py, es-sv, es-us, es-uy, es-ve, eu-es, mk-mk	Locale Group C
de, de-at, de-be, de-ch, de-de, de-lu	Locale Group D
fr, fr-be, fr-ca, fr-ch, fr-fr, fr-lu	Locale Group E
ga-ie	Locale Group F

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-za, fo-fo, gl-es, id, id-id, is, is-is, it, it-ch, it-it, kk-kz, ms, ms-bn, 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, sw-ke, sw-tz

Meaning	Symbol	Presentation	Example
Era	G	Text	AD
Year	y	Number	1996
Year (of 'Week of Year')	Y	Number	1996
Month in year	M	Text and number	July and 07
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	d	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	a	Text	pm
Hour in day (1 to 24)	k	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	h	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

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

Meaning	Symbol	Presentation	Example
Era	G	Text	AD
Year	a	Number	1996
Year (of 'Week of Year')	A	Number	1996
Month in year	n	Text and number	July and 07
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	j	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	x	Text	pm
Hour in day (1 to 24)	h	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Locale Group C

Locales: ca-es, cs, cs-cz, da, da-dk, es, es-ar, es-bo, es-cl, es-co, es-cr, es-do, es-ec, es-es, es-gt, es-hn, es-mx, es-ni, es-pa, es-pe, es-pr, es-py, es-sv, es-us, es-uy, es-ve, eu-es, mk-mk

Meaning	Symbol	Presentation	Example
Era	G	Text	AD
Year	u	Number	1996
Year (of 'Week of Year')	U	Number	1996
Month in year	M	Text and number	July and 07
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	t	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	a	Text	pm
Hour in day (1 to 24)	h	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Locale Group D

Locales: de, de-at, de-be, de-ch, de-de, de-lu

Meaning	Symbol	Presentation	Example
Era	G	Text	AD
Year	j	Number	1996
Year (of 'Week of Year')	J	Number	1996
Month in year	M	Text and number	July and 07
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	t	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	a	Text	pm
Hour in day (1 to 24)	h	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Locale Group E

Locales: fr, fr-be, fr-ca, fr-ch, fr-fr, fr-lu

Meaning	Symbol	Presentation	Example
Era	G	Text	AD

Meaning	Symbol	Presentation	Example
Year	a	Number	1996
Year (of 'Week of Year')	A	Number	1996
Month in year	M	Text and number	July and 07
Week in year	w	Number	27
Week in month	W	Number	2
Day in year	D	Number	189
Day in month	j	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	x	Text	pm
Hour in day (1 to 24)	h	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	H	Number	0
Minute in hour	m	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	z	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Locale Group F

Locales: ga-ie

Meaning	Symbol	Presentation	Example
Era	R	Text	AD
Year	b	Number	1996
Year (of 'Week of Year')	B	Number	1996
Month in year	M	Text and number	July and 07

Meaning	Symbol	Presentation	Example
Week in year	t	Number	27
Week in month	T	Number	2
Day in year	l	Number	189
Day in month	L	Number	10
Day of week in month	F	Number	2 (2nd Wed in July)
Day of Week (1=first day)	e	Number	2
Day in week	E	Text	Tuesday
a.m. or p.m. marker	a	Text	pm
Hour in day (1 to 24)	u	Number	24
Hour in a.m. or p.m. (0 to 11)	K	Number	0
Hour in a.m. or p.m. (1 to 12)	k	Number	12
Hour in day (0 to 23)	U	Number	0
Minute in hour	n	Number	30
Second in minute	s	Number	55
Millisecond	S	Number	978
Time zone	c	Text	Pacific Standard Time
Escape used in text	'	n/a	n/a
Single quote	''	n/a	'

Decimal Format Symbols

All locales

Symbol	Meaning
0	A digit that is shown even if the value is zero.
#	A digit that is suppressed if the value is zero.
.	A placeholder for decimal separator.
,	A placeholder for thousands grouping separator.
E	Separates mantissa and exponent for exponential formats.
;	Separates formats for positive numbers and formats for negative numbers.
-	The default negative prefix.

Symbol	Meaning
%	Multiplied by 100, as percentage.
‰	Multiplied by 1000, as per mille.
¤	The currency symbol. If this symbol is present in a pattern, the monetary decimal separator is used instead of the decimal separator.
¤¤	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.
X	Other characters that can be used in the prefix or suffix.
'	Used to quote special characters in a prefix or suffix.
/u221E	Infinity symbol.
/uFFFD	Not a Number symbol.

Appendix I: Drilling Through from Cognos Series 7 to Cognos 8

You can set up drill-through access from Cognos Series 7 to Cognos 8. Specifically, you can drill through to a Cognos 8 report from a PowerPlay Web report or Cognos Visualizer report. Drill through is supported for both PowerCubes and third-party cubes.

You must complete steps in both Cognos Series 7 and Cognos 8 to enable drill through. Refer to the Cognos Series 7 documentation for the specific steps you must complete in the Cognos Series 7 components. Cross references to the appropriate Cognos Series 7 documents are included below.

Setting Up Drill-through Access from PowerPlay Web

Setting up drill-through access from PowerPlay Web to Cognos 8 involves

- For PowerCubes, specifying drill-through targets for Cognos 8 reports in the Transformer model

  For more information, see the Transformer documentation.
- For third-party cubes, specifying drill-through targets for Cognos 8 reports in PowerPlay Connect.

  For more information, see the PowerPlay Connect.
- configuring drill through in PowerPlay Enterprise Server Administration

  In addition to enabling drill through to Cognos 8, you must specify the location of the Cognos 8 server and the Cognos 8 folder that contains the target reports. For more information, see the PowerPlay *Enterpriser Server Guide*.
- selecting the filters to add to the target report

  In PowerPlay Enterprise Server Administration, enable and use Cognos 8 Assistance to identify the filter expressions required in the target report. For more information, see the PowerPlay *Enterpriser Server Guide*.
- creating and testing the target report (p. 447)

Setting Up Drill-through Access from Cognos Visualizer

Setting up drill-through access from Cognos Visualizer to Cognos 8 involves

- specifying the Cognos 8 target and selecting the filters to add to the target report

  You must configure drill through to Cognos 8 for individual Cognos Visualizer reports. For more information, see the Cognos Visualizer *User Guide*.
- creating and testing the target report (p. 447)

Create and Test the Target Report

The target report must be based on a package published from a Framework Manager model that contains the metadata items that you want to filter on, or contains items that are mapped to those metadata items.

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 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`.

Steps

1. Start Report Studio and create a new report.
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, third-party cube, or Cognos Visualizer report.
10. Test the drill through in the PowerPlay report or Cognos Visualizer report.

Glossary

burst

To create many report results by running a single report once. For example, you can create a report that shows sales for each employee, and run it once, sending different results to regional managers by bursting on region. You set up bursting in Report Studio and enable it in the portal.

calculated member

A member of a dimension whose measure values are not stored but are calculated at run time using an expression.

cardinality

For OLAP data sources, cardinality is the number of members in a hierarchy. The cardinality property for a hierarchy is used to assign solve orders to expressions. For relational data sources, cardinality indicates the nature of the relationship between two query subjects, query items, or other model objects.

cascading prompt

A prompt that uses values from a previous prompt to filter the values in the current prompt or picklist.

condition

An expression that yields a boolean value. Conditions are used in query expressions, query filters, and boolean report variables that can be used for conditional formatting, styles, data sources, layouts and blocks.

Content Manager

The Cognos 8 service that manages the storage of customer applications, including application-specific security, configuration data, models, metrics, reports, and report output. Content Manager is needed to publish models, retrieve or store report specifications, manage scheduling information, and manage the Cognos namespace.

cube

A physical data source containing a multidimensional representation of data. A cube contains information organized into dimensions and optimized to provide faster retrieval and navigation in reports. In Cognos Planning, a cube (see also D-Cube) corresponds to a tab on Contributor client user interface.

data source

A relational database, dimensional cube, file, or other physical data store that can be accessed though Cognos 8.

data tree

Within a studio, 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.

dimension

In Cognos Planning, the rows, columns, and pages of a cube are created from dimensions. Dimensions are lists of related items such as Profit and Loss items, months, products, customers, and cost centers. Dimensions also contain all the calculations. One dimension can be used by many cubes.

In Cognos 8 BI a dimension is a broad grouping of descriptive data about a major aspect of a business, such as products, dates, or markets. Each dimension includes different levels of members in one or more hierarchies and an optional set of calculated members.

drill down

In Cognos Planning, drill down is a technique used to analyze D-Cube data that was imported by a D-Link. You can drill down on any single cell in a D-Cube. If the cell contains data transferred by a D-Link, drill down opens a view of the source data. If the data was imported from another D-Cube, drill down opens the appropriate selection from the source D-Cube. If the data was imported from an external source (a mapped ASCII file or an ODBC database), drill down extracts the relevant data from the source file and displays it in a special drill-down results dialog box.

In Cognos 8 BI, drill down refers to the act of navigating from one level of data to a more detailed level. The levels are set by the structure of the data. See also drill up..

drill up

The act of navigating from one level of data to a less detailed level. The levels are set by the structure of the data.

See also drill down.

group

In security, a list of users or other groups that can be used to assign access permissions and capabilities.

Groups can be referenced from third-party authentication sources or can be local to Cognos 8. Local groups are managed from the administration portal. The list of groups that an authentication user is a member of is part of the user's passport for a Cognos 8 session.

In reporting, grouping is the action of organizing common values of query item together and only displaying the value once. Headers and footers often appear after each instance of a common value in a grouped column

hierarchy

A hierarchy represents a collection of dimensional members organized into a tree structure, with each member having one or more parent members and an arbitrary number of child members.

The root of a hierarchy has no parent, and leaf members of a hierarchy have no children.

layout

In reporting, layout defines the appearance of the report, including formatting, style, and design.

In report specifications, layout is the portion of the specification that defines how the data returned by queries is presented.

level

A level is a set of members that have common attributes. For example, a geographical dimension might contain levels for country, region, and city. Levels are used to define the structure of hierarchies in a dimension.

locale

A code that is used to set the language or dialect used for browsers, report text, and so on; and the regional preferences, such as formats for time, date, money, and money expressions.

For Cognos products, you can specify a locale for the product interface (product locale) and for the data in the report (content locale).

MDX

An abbreviation for multidimensional expression language, MDX is the multidimensional equivalent of SQL. Cognos 8 allows report authors to type in their queries using MDX expressions in Report Studio for dimensional data sources.

measure

A performance indicator that is quantifiable and used to determine how well a business is operating. For most business purposes, the aggregate values of a measure are more useful than individual values. For example, measures can be Revenue, Revenue/Employee, and Profit Margin %. In relational modeling, this is also called a "fact."

member

A member is a unique item within a hierarchy. For example, Camping Equipment and 4 Man tent are members of the Products Hierarchy.

See also member unique name.

Metric Studio application

A set of metric types, metrics, and scorecards that monitor the performance of an organization. Each organization may have several Metric Studio applications monitoring distinct units, such as operating companies or divisions that have different user communities, metric types or reporting calendars.

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 8 BI, a design model is created and maintained in Framework Manager. The design model or a subset of the design model must be published to the Cognos 8 server as a package for users to create and run reports.

In Cognos Planning, a model is a group of D-Cubes, D-Lists, D-Links, and other objects stored in a library. A model may reside in one or more libraries, with a maximum of two for Contributor.

package

A subset of a model, which can be the whole model, to be made available to the Cognos 8 server.

For Metric Studio users, see metric package.

page set

A set of pages associated with a specified group or level that indicates the page set is to be repeated for that group or level. For example, render a specified set of pages for each customer in the query.

prompt

A report element that asks for parameter values before the report is run.

properties pane

Within a studio, the properties pane provides an overview of the properties for selected data. You can also use the properties pane to make several changes and apply them at the same time, instead of repeating several different commands.

query

A specification for a set of data to retrieve from a data source. A report specification can contain one or more queries.

The type of object created and edited by Query Studio. A query is a subtype of report.

query item

A representation of a column of data in a data source. It contains a reference to a database column, a reference to another query item, or a calculation. Query items may appear in a model or in a report.

query subject

One of the types of objects inside a model. A query subject can be defined as a collection of references to items of other query subjects, or expressed as an SQL expression that represents selected query items. These query items will be retrieved from objects such as tables, synonyms, views, and so on. Query subjects contain query items.

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 for row in the associated query.

report

A set of data deliberately laid out to communicate business information. Depending on the context, "report" may refer to report specification or report output.

Report (more precisely, report specification) is the generic term for the objects created or edited by Query Studio, Report Studio, and Analysis Studio.

report output

A report output combines data at the point in time when the report was run with a report specification. It is a document that can be displayed, printed, or emailed. Cognos 8 can produce report outputs in HTML, PDF, Excel, or CSV formats.

report specification

The definition of queries, prompts, layouts, and styles that make up a report. A report specification is combined with data by a run operation to create report outputs. You create report specifications by using Report Studio, Query Studio, Analysis Studio, or through the Software Development Kit.

style sheet

A file that defines the formatting and appearance of data or a document. In XML, style sheets may be extensible style sheet language (XSL) files or extensible style sheet language transformation (XSLT) files.

In HTML, style sheets are cascading style sheet (CSS) files.

XSL and CSS style sheets can be embedded inside any XML document or referenced as an external, separate file.

summary

In reporting and analysis, summaries are aggregate values that are calculated for all the values of a particular level or dimension. Examples of summaries include total, minimum, maximum, average, and count.

template

A reusable report layout or style that can be used to set the presentation of a query or report.

work area

The area within a studio that contains the report, analysis, query or agent currently being used.

Index

Symbols

"Not Applicable" Characters, 431

Numerics

100% stacked charts, 226
3-D Area, 351
3-D Bar, 351
3-D charts, 226
3-D Combination Chart, 351
3-D Line, 352
3-D Scatter Chart, 352
3-D Viewing Angle, 379

A

absolute charts, *See* standard charts
Absolute Position, 379
Across, 379
adding
  bookmarks, 85
  borders, 83
  calculations, 113
  colors, 84
  conditional formatting, 159
  data, 31
  footers, 82
  headers, 82
  languages, 163
  layouts, 162
  multimedia files, 88
  pages, 94
  prompt buttons, 127
  prompts, 119
  query macros, 140
  summaries, 108
  text, 84
  variables, 158
aggregate button, 109
Aggregate Function, 379
aggregate functions
  using in data item expressions, 110
aggregating values
  charts, 111
  crosstabs, 111
aggregation modes, 28
aggregation properties
  model, 109
aliases
  in maps, 73
aligning
  objects, 91
alignment
  unexpected results in Report Studio report, 207
Allow 45° Rotation, 380
Allow 90° Rotation, 380
Allow Is Missing, 380
Allow Skip, 380
Allow Stagger, 380
Analysis Studio analyses
  opening, 146
Angular Axis, 352, 380
Angular Measure, 352
Any Error Characters, 431
Application, 380
applying
  padding, 92
  table styles, 91
Area, 352
area charts, 219
As of Time Expression, 352
Auto Group & Summarize, 109, 381
Auto Truncation, 381
automating
  Cognos 8 using the SDK, 42
Auto-Sort, 381
Auto-Submit, 381
Avoid Division by Zero, 381
Avoid Label Collision, 381
axes
  bipolar, 184
  customizing titles, 68
  dual, 184
  title in maps, 76
Axis Assignment, 381, 382
Axis Labels, 382
Axis Line, 382
Axis Title, 353, 382

B

Background Color, 382
background color
  adding, 84
Background Image, 382
background images
  inserting, 85
Bar, 353
bar charts, 54
Bar, Line, or Area Index, 383
Baseline, 353
Baselines, 383
baselines
  chart, 61
bipolar axis, 184
Block, 354

Block Variable, 383
blocks
  inserting, 88
Bookmark, 354
bookmarks
  adding, 85
  creating for drill-through, 165
Border, 383
Border Color, 383
Borders, 384
borders
  adding, 83
  formatting, 83
Bottom Position (px), 384
Box Type, 384
browser
  report does not display, 207
browsing data
  in expression editor, 234
Bubble Chart, 354
bubble charts, 221
Bubble Measure, 354
Build Prompt Page tool, 119
burst groups
  defining, 154
burst options
  setting, 155
burst recipients
  creating calculated fields, 152
  defining, 152
burst reports
  creating, 152
  creating against dimensional data sources, 156
burst tables
  creating, 153
bursting
  enabling, 155
bursts
  definition, 449
business keys
  using for drill-through, 167

C

calculated fields
  creating for bursting, 152
Calculated Member, 354
calculated members
  definition, 449
Calculation Intersection, 115, 385
calculations
  adding, 113
  inserting, 88
  resolving in crosstabs and charts, 114
  resolving in dimensional data sources, 115
Calendar Type, 385, 431
CAMID, *See* Cognos Access Manager ID
Caption, 354, 385
Cardinality, 385
cardinality
  definition, 449
Cascade Source, 385
cascading prompts
  adding, 127
  definition, 449
cascading style sheets
  classes, 101
Case Insensitive, 385
Catalog, 385
changes in model not in report, 206
changing
  chart scale intervals, 61
  package connections, 150
  referenced objects, 93
chart background
  changing, 60
Chart Body, 355
chart configurations, 216
  100% stacked, 226
  3-D, 226
  absolute, *See* standard charts
  changing, 181
  stacked, 225
  standard
Chart Footer, 355
Chart Node Member, 355
Chart Orientation, 386
Chart Subtitle, 355
Chart Text Item, 355
Chart Title, 355
Chart Type, 386
chart types, 215
charts, 54
  adding data, 180
  adding notes, 62
  aggregating values, 111
  area, 219
  axis titles, 68
  bar
  baselines, 61
  bubble, 221
  changing colors, 58
  changing configuration, 181
  changing lines, 61
  changing presentation, 184
  changing the background, 60
  circle, *See* pie charts, 216
  column
  combination, 220
  combining sets of data
  conditional palettes, 69
  converting, 186
  creating, 179
  curve, *See* line charts
  customizing, 56
  customizing values, 180
  defining query contexts, 67
  dial, *See* gauge charts
  drill-down, 66
  drill-through, 64
  drill-up, 66
  formatting axis values, 63
  gauge
  grouping data, 181

horizontal, *See* bar charts
legends, 66
line
markers, 62
metric range, 223
Microsoft Excel limitations, 346
nesting data, 186
palettes, 58, 59
Pareto, 218
pie, 216
point, 222
polar, 223
progressive column
properties, 56
quadrant, 222
radar
resizing, 100
resolving multiple calculations, 114
scales, 61
scatter
sector, *See* pie charts, 216
showing values, 180
spider, *See* radar charts
stacked, 54
star, *See* radar charts
vertical bar, *See* column charts
waterfall, *See* progressive column charts
XY, *See* scatter charts
choosing
database connections, 139
templates, 31
circle charts, *See* pie charts, 216
Class, 356, 386
classes
changing the default for a report, 101
Clear Parameter Values, 35
clipboards
copying reports, 146
opening reports, 146
Clock, 431
Clock Mode, 386
Cognos 8 Go! Office
creating reports for, 41
Cognos 8 Software Development Kit, 42
Cognos Access Manager ID, 153
Cognos Connection
java.lang.OutOfMemory error, 207
Cognos SQL, 138
Cognos Visualizer reports
setting up drill through to Cognos 8, 447
Color Legend Title, 386, 387
colors
adding, 84
changing in charts, 58
custom, 84
column charts, 54
Column Intersection, 387
Column Titles, 387
column widths
Microsoft Excel limitations, 345
columns, 31
adding multiple items in a single column, 150
calculated, 113
changing order, 96
grouping, 43
renaming, 97
setting the group span, 45
sorting, 107
swapping, 97
Combination Chart, 356
combination charts, 220
combining reports, *See* master-detail relationships
Component Override, 356
Component Reference, 387
concatenating
strings, 235
Condition Explorer, 24
Conditional Block, 356
conditional block lists
inserting, 88
conditional formatting
adding, 159
Conditional Palette, 387
conditions
defining, 157
definition, 449
configuring
Web browsers, 29
Connecting Lines, 387
Content Manager
definition, 449
Contents Height, 388
converting
charts, 186
lists to crosstabs, 52
lists to repeaters, 79
reports to templates, 39
to MDX, 138
to SQL, 138
copied objects
updating, 93
copyright, 2
Count
changes between Cognos ReportNet and Cognos 8, 231
Count Distinct, 231
creating
accessible reports, 40
burst tables, 153
charts, 179
crosstab nodes, 49
crosstabs, 178
dashboards, 197
drill-through reports, 168
drill-up/drill-down reports, 172
filter expressions for drill through from Cognos Series 7 to Cognos 8, 447
invoices, 195
lists, 177
mailing labels, 79
master-detail relationships, 175
member sets, 174
multilingual reports, 163
multiple-page reports, 192
prompt pages, 119

prompts, 119, 191
Query Studio template, 40
report specifications, 30
reports, 30
reports for Cognos 8 Go! Office, 41
section headings, 107
single-edge crosstabs, 49
target reports for drill through from Cognos Series 7, 447
templates, 39, 194
Cross Product Allowed, 388
Crosstab, 356
Crosstab Columns, 357
Crosstab Corner, 357
Crosstab Fact Cells, 357
crosstab fact cells
formatting, 51
Crosstab Intersection, 358
crosstab intersections
formatting, 51
Crosstab Member Fact Cells, 358
crosstab member fact cells
formatting, 51
Crosstab Node Member, 358
crosstab nodes, 48
creating, 49
members, 48
moving, 49
crosstab reports, 48
Crosstab Rows, 358
Crosstab Space, 359
crosstabs
aggregating values, 111
aggregation modes, 28
applying table styles, 91
creating, 178
creating single-edge, 49
crosstab node members, 48
crosstab nodes, 48
discontinuous, 53
formatting, 51
intersections, 51
nesting data, 49
resolving multiple calculations, 114
CSV format
producing reports, 37
cubes
definition, 449
Cumulation Axis, 388
Cumulation Label, 388
Cumulation Line, 359
Cumulation Line Axis, 359
Cumulation Line Label, 359
Cumulative Line, 388
currencies
formatting, 116
Currency, 431
currency data format
Microsoft Excel limitations, 345
Currency Display, 431
Currency Symbol, 431
Currency Symbol Position, 431
Current Block, 388
curve charts, *See* line charts
Custom Label, 388
customizing
charts, 56
maps, 76

D

dashboards
creating, 197
data
adding, 31
adding to charts, 180
aggregating in charts, 111
aggregating in crosstabs, 111
browsing in expression editor, 234
combining sets in charts
filtering, 103
filtering date columns, 105
filtering SAP BW data source, 105
formatting, 116
formatting using patterns, 437
grouping in charts, 181
highlighting, 160
showing for specific time periods, 118
sorting, 106
summarizing, 108
Data Format, 389
data formats
Microsoft Excel limitations, 345
Data Item, 359, 389
data item expressions
using aggregate functions, 110
Data Item Label, 389, 390
Data Item Value, 390, 391
data items
adding, 31
adding multiple items in a single column, 150
browsing data in expression editor, 234
Data Language, 391
Data Source, 391
data sources
conformed dimensions, 129
definition, 449
dimensionally modeled relational, 141, 235
Dimensionally Modeled Relational (DMR), 32
non-conformed dimensions, 129
OLAP, 32, 235
data stores
conformed dimensions, 167
data trees
definition, 449
database connections
choosing, 139
Date & Time Prompt, 359
date and time prompts, 119
Date Ordering, 431
Date Prompt, 360
date prompts, 119
Date Separator, 432
Date Style, 432
dates

filtering, 105
formatting, 116
using for drill-through, 167
Decimal Separator, 432
Default Measure, 360, 391
Default Measure Solve Order, 391
Default Selections, 392
default selections
specifying in prompts, 126
Default Title, 392
Define Contents, 392
Define Member Sets, 392
defining
burst groups, 154
burst recipients, 152
Depth, 392
derived column, 106
Description, 392
Detail, 392
Detail Filter, 360
Diagram Identifier, 392
dial charts, *See* gauge charts
Dial Outline Color, 393
Dictionary, 393
dictionary property in maps, 73
Dimension, 360
dimensional coercion rules, 236
dimensional data sources
creating burst reports, 156
creating group reports, 43
linking members to create a master-detail relationship, 176
resolving calculations, 115
running a report against, 37
Dimensionally Modeled Relational (DMR) data sources, 32
dimensionally modeled relational data sources, 141, 235
dimensions, 32
conformed for drill-through, 167
definition, 450
searching, 34
working with, 32
discontinuous crosstabs
creating, 53
Display After Overall Header, 393
Display AM / PM Symbols, 432
Display As Exponent, 432
Display Days, 432
Display Eras, 432
Display Frequency, 393
Display Hours, 432
Display Layer, 360
Display Milliseconds, 393, 432
Display Minutes, 432
Display Months, 433
Display Seconds, 393, 433
Display Time Zone, 433
Display Value, 393
Display Weekdays, 433
Display Years, 433
Divide By Zero Characters, 433
DMR, *See* Dimensionally Modeled Relational (DMR) data sources
double counting, 231
Down, 394
drill through
setting up for Cognos Visualizer reports, 447
setting up for PowerPlay reports, 447
Drill Throughs, 394
drilling down
charts, 66
definition, 450
drilling through
charts, 64
maps, 77
Microsoft Excel limitations, 346
See also drill-through
drilling up
charts, 66
definition, 450
drill-through
between packages, 167
between PowerCubes and packages, 167
business keys, 167
concepts, 165
conformed dimensions, 167
formats, 165
members, 166
model-based access, 165
report-based access, 165
using bookmarks, 165
using dates, 167
using scope in model-based, 168
using URLs, 165
values, 166
drill-through access
concepts, 164
drill-through definition, 165
drill-through reports
creating, 168
specifying text, 170
drill-up/drill-down reports
creating, 172
member sets, 173
dual axis, 184
Duplicates, 394

E

editing
MDX, 138
SQL, 138
emailing reports
Microsoft Excel format reports, 346
Embed, 394
enabling
bursting, 155
error messages
501 not implemented, 205
java.lang.OutOfMemory error, 207
Web server error, 205
escape character, 235
examples
adding a note to a chart, 63
adding data to charts, 180
adding interval prompts, 121

changing chart configuration, 181
changing chart presentation, 184
changing palettes, 59
combining sets of data in charts
converting charts, 186
create column charts, 55
create drill-through charts, 64
create map reports, 71
create metrics range charts, 188
creating conditional chart palettes, 69
creating discontinuous crosstabs, 53
creating mailing labels, 79
customizing chart palettes, 59
customizing values in chart axis titles, 68
customizing values in charts, 180
defining data values for region layers, 72
formatting axis values, 63
grouping data in charts, 181
nesting data in charts, 186
showing values in chart legends, 66
showing values in charts, 180
Execution Method, 394
Execution Optimization, 394
Explorer Bar, 24
Exponent Symbol, 433
Expression, 395
expression editor
Block Functions, 261
browsing data, 234
Business Date/Time Functions, 258
Common Functions, 266
Constants, 256
Constructs, 257
creating calculations, 113
creating filters, 103
Data type casting functions, 335
DB2, 280
DB2 Cast, 288
DB2 Math, 290
DB2 Trigonometry, 290
dimensional coercion rules, 236
Informix, 292
Informix Math, 295
Informix Trigonometry, 295
Macro Functions, 262
Member Summaries, 253
MS Access, 296
MS Access Cast, 300
MS Access Math, 301
MS Access Trigonometry, 301
Operators, 239
Oracle, 302
Oracle Math, 307
Oracle Trigonometry, 307
prompts, 123
quotation marks in literal strings, 235
Red Brick, 309
Report functions, 331
SAP BW, 324
SAP BW Math, 325
SAP BW Trigonometry, 324
SQL Server, 313
SQL Server Cast, 316
SQL Server Math, 317
SQL Server Trigonometry, 318
Summaries, 244
Sybase, 325
Sybase Math, 329
Sybase Trigonometry, 330
Teradata, 319
Teradata Trigonometry, 322
expressions
functions not available, 235
quotation marks in literal strings, 235

F

Face Color, 395
Fact, 360
Fact Cells Precedence, 395
fact, *See* measure
Field Set, 361
field sets
inserting, 88
Fill Effects, 395
filter expressions
creating for drill through from Cognos Series 7 to Cognos 8, 447
filtering
data, 103
data from SAP BW data sources, 105
date columns, 105
filters
using with multiple-fact queries, 129
values not recognized in multilingual query items, 206
First Column Color, 396
First Date, 396
First Label Index, 396
first-rows optimization, 139
Floating, 100, 396
Font, 396
Font Auto-Sizing, 397
fonts
in maps, 76
specifying, 84
Footer, 397
footers
adding, 82
formatting, 82
in maps, 75, 76
Foreground Color, 397
foreground colors
adding, 84
formats
drill-through, 165
not supported for Microsoft Excel, 346
reports, 35
formatting
borders, 83
chart axis values, 63
colors, 84
crosstabs, 51
currencies, 116
data, 116

data using patterns, 437
dates, 116
footers, 82
headers, 82
limitations for Microsoft Excel outputs, 342
lists, 47
reports in page structure view, 82
single object, 117
text, 84
values in map legends, 76

G

Gauge Chart, 361
gauge charts
Gauge Labels, 361, 397
Gauge Numerical Axis, 361
Gauge Palette, 398
Generated Prompt, 361
generated prompts, 119
Generated SQL/MDX, 398
graphs, *See* charts
Gridlines, 398
gridlines
major, 61
minor, 61
Group Size (digits), 433
Group Span, 398
group spans
setting, 45
grouped reports have incorrect running totals, 207
grouping
data, 43
Grouping & Sorting, 398
Grouping Type, 398
groups
definition, 450
guidelines
laying out reports, 82

H

Has Fact Cells, 398
headers
adding, 82
formatting, 82
Height (px), 398
Hide Adornments, 399
Hide Text, 399
hiding
objects, 159
hierarchies
definition, 450
inserting, 34
highlighting
data, 160
Hole Size (%), 399
Horizontal Alignment, 399
horizontal charts, *See* bar charts
HTML, 399
HTML format
producing a report, 35
HTML Item, 361
HTML items
inserting, 88
HTML Source Variable, 399
Hyperlink, 362
Hyperlink Button, 362
hyperlink buttons
inserting, 88
not supported for Microsoft Excel, 346
hyperlinks
inserting, 88

I

icons, 31
Ignore Data with No Features, 400
ignoring data with no features
in maps, 73
Image, 362
images
inserting, 85
Microsoft Excel limitations, 341
Include Zero For Auto Scale, 400
Indentation Length, 400
Indentation Start Level, 400
inheritance
objects, 27
Insertable Objects pane, 24
inserting
background images, 85
blocks, 88
calculations, 88
conditional block lists, 88
dates, 88
field sets, 88
hierarchies, 34
HTML items, 88
hyperlink buttons, 88
hyperlinks, 88
images, 85
layout component references, 88
member properties, 34
page sets, 94
repeaters, 88
rich text items, 88
row numbers, 88
tables, 88, 91
International Currency Symbol, 434
Internet Explorer 7
configuration settings for Report Studio, 205
Intersection Name, 400
Interval Prompt, 362
interval prompts, 119
adding, 121
invoices
creating, 195

J

Japanese
Microsoft Excel limitations, 345
java.lang.OutOfMemory error, 207
Join, 362
Join Relationships, 400

joining
  nested page sets, 95

K

Key, 362

L

Label, 401
Label Control, 401
Labels, 402
labels
  specifying, 65
language
  values not recognized in query, 206
languages
  adding to reports, 163
Last Date, 402
laying out reports, 81
  guidelines, 82
layout component cache, 93
Layout Component Reference, 363
layout component references
  inserting, 88
layouts
  adding, 162
  definition, 450
  description, 25
  pages, 26
  relating to queries, 129
  report objects, 26
Left Position (px), 402
Legend, 363, 402
Legend Label, 402
Legend Title, 363, 403
legends
  customizing, 66
  in maps, 74, 75, 76
  positioning, 66
Level, 363
level hierarchies, 32
Level Hierarchy, 363
levels, 32
  definition, 450
  member unique names, 166
LFA.dll file, 145
Limit Type, 403
limitations
  Microsoft Excel output, 341
Line, 363, 403
line charts, 54
Line Style, 403
Line Styles, 403
Line Weight (pt), 403
linear scales
  showing, 61
lines
  changing in charts, 61
List, 364
List Cell, 364
List Column, 364
List Column Body, 364
List Column Title, 365
List Columns, 364, 365
List Columns Body Style, 365
List Columns Title Style, 365
List Footer, 365
list footers
  adding, 82
List Header, 366
list headers
  adding, 82
List Page Footer, 366
List Page Header, 366
List Row, 366
List Row Cells Style, 366
lists, 43
  applying table styles, 91
  converting to crosstabs, 52
  converting to repeaters, 79
  creating, 177
  formatting, 47
  grouping data, 43
  headers and footers, 82
literal strings
  in expressions, 235
locales
  definition, 450
  locale-sensitive properties, 117
logarithmic scales
  showing, 61

M

mailing labels
  creating, 79
managing
  changes in packages, 149
  reports, 42
Mantissa (digits), 434
Map, 366
Map & Layers, 403
Map Drills, 403
Map Location, 367
Map Refinement Location, 367
maps, 70
  adding to legends, 189
  aliases, 73
  changing the size of points, 191
  creating, 71, 188
  customizing, 76
  dictionary property, 73
  drill through, 77
  footers, 75
  format legends, 189
  ignoring data with no features, 73
  legends, 74
  matching data values, 73, 189
  Microsoft Excel limitations, 346
  notes, 75
  parts of a map report, 71
  point layers, 74, 190
  region layers, 72
  resizing, 100

size of points, 75
subtitles, 75
titles, 75
Margin, 404
margins
setting, 92
Marker, 367
Marker Color, 404
Marker Label, 404
Marker Text Location, 404
Markers, 404
markers
charts, 62
Master Detail Relationships, 404
master-detail relationships
creating, 175
linking members from two dimensional data sources, 176
using to join page sets, 95
Maximum Characters, 404
Maximum Execution Time, 405
Maximum No. of Digits, 434
Maximum Rows Retrieved, 405
Maximum Size (pt), 405
Maximum Tables, 405
Maximum Text Blob Characters, 405
Maximum Truncation Characters, 405
Maximum Value, 406
MDX, 367, 406
adding your own, 138
converting to, 138
definition, 451
editing, 138
viewing, 138
working with, 138
measures, 31
definition, 451
specifying default, 50
Member Hierarchy, 368
Member Offset (%), 406
member properties, 32
inserting, 34
Member Property, 368
Member Set, 368
member sets
creating, 174
member unique names, 235
relational metadata, 166
members
definition, 451
drill-through, 166
inserting properties, 34
linking from two dimensional data sources to create a master-detail relationship, 176
members folders, 32
metric range charts, 223
Metric Studio applications
definition, 451
Metric Studio Diagram, 368
Metrics Range Chart, 368
Microsoft Excel
report limitations, 341
Microsoft Excel format
producing a report, 38
Microsoft Excel limitations
nested report objects, 345
Minimum No. of Digits, 434
Minimum Size (pt), 406
Minimum Value, 406
Minor Gridlines, 406
Missing Value Characters, 434
model
aggregation properties, 109
model versioning, 206
model-based drill-through
using scope, 168
models, 31
definition, 451
using design filters, 35
modifying
prompts, 124
properties in page structure view, 82
moving
crosstab nodes, 49
Multi-Line, 407
multimedia files
adding, 88
multiple values
selecting in prompts, 125
multiple-fact queries
filters, 129
multiple-page reports, 94
Multi-Select, 407
MUN, *See* member unique names

N

Name, 407, 408
names
updating references, 150
native SQL, 138
Negative Column Color, 408
Negative Pattern, 434
Negative Sign Position, 434
Negative Sign Symbol, 434
nested report objects
Microsoft Excel limitations, 345
nesting
data in charts, 186
data in crosstabs, 49
No Data Features Size (pt), 408
No. of Decimal Places, 434
Note, 368
Note Border, 408
Note Content, 369
Notes, 408
notes
adding to charts, 62
in maps, 76
maps, 75
number data format
Microsoft Excel limitations, 345
number of decimals
rounding mode used by Cognos 8, 117
Number of Regression Lines, 409

Numbers Only, 409
Numeric Overflow Characters, 434
Numeric Value, 409
Numerical Axis, 369, 409

O

objects
  adding, 26
  aligning, 91
  as containers, 27
  formatting, 117
  hiding, 159
  hierarchies, 27
  inheritance, 27
  inserting, 88
  reusing, 92
  setting properties, 98
  showing, 159
  types, 27
  updating, 93
OLAP
  data sources, 32, 235
opening
  Analysis Studio analyses, 146
  Query Studio reports, 146
  reports locally, 145
options
  setting, 28
Ordinal Axis, 369, 409
Outer Join Allowed, 410
overline text formats
  Microsoft Excel limitations, 344
Override Dimension Info, 410
Overrides, 410

P

packages, 32
  changing connections, 150
  definition, 451
  drilling-through, 167
  managing changes, 149
  referencing items in child queries, 137
  refreshing, 30
  specifying, 30
Padding, 410
padding
  applying, 92
Padding Character, 435
Page, 369
Page Body, 369
page breaks
  controlling, 101
page breaks, *See* page sets
Page Explorer, 24
Page Footer, 369
Page Header, 370
Page Set, 370
page sets
  definition, 451
  inserting, 94
  joining nested page sets, 95
page structure view, 82
pages, 26
  adding, 94
  associating queries, 94
Pagination, 98, 410
Palette, 410, 411
palettes
  charts, 58, 59
  conditional, 69
Parameter, 411
parameters
  for prompts, 119, 123
  using in master-detail relationships, 175
Pareto Chart, 370
Pareto charts, 218
Pattern, 435
patterns
  using to format data, 437
PDF manual printing, 206
PDFs
  producing a report, 35
Percent of Axis, 411
Percent Scale (integer), 435
Percentage Symbol, 435
Percentile, 411
Performance Pattern, 411
Pie Chart, 370
pie charts, 216
  limitations for Microsoft Excel outputs, 342
Pie Labels, 370, 411
planning
  reports, 23
point charts, 222
Point Color, 411
Point Layer, 371
point layers, 74
Point Measure, 371
Point Shape, 411, 412
Point Size (pt), 412
Point Size Measure, 371
points layers, 75
Polar Chart, 371
polar charts, 223
Polynomial Exponent, 412
pop-up blocking, 207
Position, 412
Positive Column Color, 412
PowerPlay reports
  setting up drill through to Cognos 8, 447
Pre-populate If Parent Optional, 412
Pre-populate Levels, 412
presentation
  changing in charts, 184
Pre-Sort, 413
printing
  reports, 39
printing PDF manuals, 206
Processing, 413
Progressive Axis, 413
Progressive Chart, 371
progressive column charts
Projection List, 413

Prompt Button, 372
prompt buttons
adding, 127
prompt controls
changing interfaces, 125
date and time prompts, 119
date prompts, 119
generated prompts, 119
interval prompts, 119
select and search prompts, 119
text box prompts, 119
time prompts, 119
tree prompts, 119
value prompts, 119
See also prompts
prompt pages
creating your own, 119
prompts
adding, 119
adding prompt buttons, 127
cascading, 127
changing prompt control interfaces, 125
creating, 119, 191
creating in a report page, 124
creating using query macros, 141
definition, 451
interval, 121
modifying, 124
pages, 119
parameters, 119, 123
required, 125
requiring user input, 125
selecting multiple values, 125
showing and hiding status, 126
specifying default selections, 126
specifying values, 126
tree, 128
unable to delete, 208
using the Build Prompt Page tool, 119
Properties, 413
properties
charts, 56
locale-sensitive, 117
Properties pane, 24
properties panes
definition, 451
Push To Bottom, 413

Q

quadrant charts, 222
queries, 27
associating to pages, 94
definition, 451
master-detail, *See* master-detail relationships
referencing package items in child queries, 137
relating to layouts, 129
working with, 128
working with dimensional data sources, 129
working with MDX, 138
working with SQL, 138
Query, 372, 413
query context
defining in charts, 67
Query Explorer, 24, 128
query items
definition, 452
query macros
adding, 140
using to create prompts, 141
Query Operation, 372
Query Reference, 372
Query Studio
creating templates in Report Studio, 40
Query Studio reports
opening, 146
query subjects
definition, 452
quotation marks
in literal strings, 235

R

Radar Chart, 372
radar charts
Radar Type, 414
Radial Axis, 373, 414
Radial Measure, 373
Range, 414
Range Label, 414
reference
package items in child queries, 137
referenced objects
changing, 93
updating, 93
refreshing
packages, 30
Region Layer, 373
region layers, 72
Region Measure, 373
Regression Line, 373, 414
Regression Type, 414
Relative Alignment, 415
removing
section headings, 108
renaming
columns, 97
Render Variable, 415
rendering problems in reports, 208
reordering
columns, 96
Repeater, 373
Repeater Direction, 415
Repeater Table, 373
Repeater Table Cell, 374
repeater tables
definition, 452
repeaters, 79
converting from lists, 79
definition, 452
inserting, 88
report does not display
browser limitations, 207
Report Expression, 415, 416

report objects
as containers, 27
hierarchy, 27
locking and unlocking, 27
types, 27
working with, 27
report outputs
definition, 452
report properties
changing, 101
report specifications
creating, 30
definition, 452
showing for a selected object, 42
viewing, 42
Report Studio
changing default classes, 101
does not start, 207
Insertable Objects pane, 24
planning reports, 23
Properties pane, 24
saving reports, 34
setting options, 28
unexpected alignment in reports, 207
user interface, 23
using with Internet Explorer 7, 205
visual aids, 25
work area, 25
report types, 43
charts, 54
crosstab, 48
lists, 43
repeaters, 79
ReportNet
preserving report appearance, 101
reports
accessible, 40
adding colors, 84
basic structure, 25
building using your own MDX, 138
building using your own SQL, 138
bursting, 152
combining, *See* master-detail relationships
controlling rows per page, 37
converting to templates, 39
copying to the clipboard, 146
creating, 30
creating for Cognos 8 Go! Office, 41
creating multiple-page, 192
creating prompts in a report page, 124
creating section headings, 107
defining conditions, 157
definition, 452
drill-through, 168
drill-through text, 170
drill-up/drill-down, 172
formats, 35
guidelines for layouts, 82
laying out, 81
managing, 42
multilingual, 163
not showing changes to model, 206
opening and saving locally, 145
opening from the clipboard, 146
pages, 26
planning, 23
printing, 39
producing in CSV format, 37
producing in Microsoft Excel format, 38
producing in XML format, 38
removing section headings, 108
removing upgrade messages, 146
running, 35
running against dimensional data sources, 37
running against SAP BW, 37
running in Report Studio, 35
samples, 209
saving in Report Studio, 34
validating, 35
Web server error, 205
Required, 417
required prompts, 125
resizing
charts, 100
maps, 100
reusing
objects, 92
Rich Text Item, 374
rich text items
inserting, 88
supported elements, 90
Right Position (px), 417
Rollup Aggregate Function, 417
Rollup Processing, 417
Rotate Labels, 417
rounding mode
used by Cognos 8 for number of decimals, 117
Row Intersection, 418
row numbers
inserting, 88
rows
swapping, 97
Rows Per Page, 35, 418
rows per page
controlling, 37
run options
setting, 35
running reports, 35
against dimensional data sources, 37
against SAP BW, 37
unable to delete a prompt, 208
running totals incorrect, 207

S

sample reports
included with Cognos 8, 209
SAP BW data sources
creating burst reports against, 156
running reports against, 37
SAP variable properties
unsupported, 37
saving
reports, 34

reports locally, 145
Scale, 418, 435
Scale Interval, 418
scales
changing interval, 61
linear, 61
logarithmic, 61
Scatter Chart, 374
scatter charts
Scope, 418
scope
setting in summary filters, 103
using in drill-through, 168
SDK, *See* Cognos 8 Software Development Kit
searching
dimensions, 34
Secondary Group Size (digits), 435
section headings
creating, 107
removing, 108
sector charts, *See* pie charts, 216
Secure Socket Layer
Microsoft Excel limitations, 345
Security Error Characters, 435
Select & Search Prompt, 374
select and search prompts, 119
Select UI, 419
Selector, 418
Separator, 419
Series Color, 419
Set Operation, 419
setting
burst options, 155
margins, 92
object properties, 98
run options, 35
Show Caption, 419
Show Data Range in Legend, 419
Show Features with No Data, 419
Show Feelers, 419
Show Legend Values, 420
showing
objects, 159
the report specification, 42
single-edge crosstabs
creating, 49
Size & Overflow, 420
Size and Overflow, 100
Size Legend Title, 420
Slicer Member Set, 374
Solve Order, 420
solve order, 114
combined with calculated rollup aggregation function, 115
Sorting, 420
sorting
data, 106
multiple columns, 107
Source Type, 420, 421
Spacing & Breaking, 422
Spacing and Breaking, 99
specifying
default measures, 50
fonts, 84
labels, 65
packages, 30
spider charts, *See* radar charts
Spider Effects, 422
SQL, 375, 422
adding your own, 138
Cognos, 138
converting to, 138
editing, 138
first-rows optimization, 139
native, 138
viewing, 138
working with, 138
SQL Syntax, 422
SSL, See Secure Socket Layer, 345
stacked charts, 225
standard charts
Standard Deviations, 422
star charts, *See* radar charts
startup
Report Studio does not start, 207
Static Choices, 422
status
showing and hiding in prompts, 126
strings
concatenating, 235
style sheets
definition, 452
Style Variable, 422
Subtitle, 423
summaries
adding, 108
automatic, 227
average, 227
calculated, 115, 227
count, 227
count distinct, 227
custom, 227
definition, 452
maximum, 227
median, 227
minimum, 227
none, 227
not applicable, 227
standard deviation, 227
summarize, 227
total, 227
variance, 227
summarizing
data, 108
Summary Filter, 375
Suppress, 423
swapping
columns and rows, 97

T

Table, 375
Table Cell, 375
Table Properties, 423
Table Row, 375

table widths
  Microsoft Excel limitations, 345
tables
  applying styles, 91
  inserting, 88, 91
  setting properties, 101
tabular data
  viewing, 35
Target Color, 423
Target Marker, 423
Target Marker Border Color, 424
Target Marker Position, 424
Target Measure, 375
Target Range (%), 424
target reports
  creating for drill through from Cognos Series 7, 447
templates
  choosing, 31
  converting from reports, 39
  creating, 39, 194
  definition, 452
Text, 424
text
  adding, 84
  formatting, 84
Text Box Prompt, 375
text box prompts, 119
Text Flow & Justification, 424
Text Flow and Justification, 99
Text Item, 376
Text Source Variable, 424
Thousands Separator, 435
Time Prompt, 376
time prompts, 119
Time Separator, 435
Time Style, 435
Time Unit, 436
Title, 425
titles
  chart axis, 68
  in maps, 76
Tolerance Color, 425
Tolerance Label, 425
Tolerance Measure, 376
Tooltips, 425
tooltips
  in maps, 76
Top Position (px), 425
Total Column, 376, 425
Total Column Color, 425
Total Column Label, 376
Tree Prompt, 376
tree prompts, 119
  controlling the data that appears, 128
Truncation, 426
Truncation Text, 426
Type, 426

U

ungrouping
  data, 43
union
  combining sets in charts
units of measure, 116
units of measure notation, 37
unsupported SAP variable properties, 37
updating
  copied objects, 93
  name references, 150
  referenced objects, 93
upgrade messages
  removing, 146
upgrading reports
  removing upgrade messages, 146
Upper Range Skew (%), 426
URL, 426
URL Source Variable, 426
Usage, 427
Use 1.x Behavior, 427
Use Detail Value On Page, 427
Use For Parameter Info, 427
Use Local Cache, 427
Use Same Range For All Instances, 427
Use SQL With Clause, 428
Use Thousands Separator, 428, 436
Use Value, 428
user input
  requiring in prompts, 125
using
  model design filters, 35

V

validating
  reports, 35
Value Location, 428
Value Prompt, 377
value prompts, 119
Value Representation, 428
Value Type, 429
Values, 428, 429
values
  customizing in charts, 180
  showing in charts, 180
  specifying in prompts, 126
Variable, 377
variables
  adding, 158
version of document, 2
Vertical Alignment, 429
vertical bar charts, *See* column charts
viewing
  MDX, 138
  SQL, 138
  tabular data, 35
  the report specification, 42
viewing reports
  incorrect running totals in grouped reports, 207
  report does not render properly, 208
Visible, 429
visual aids, 25
Visual Angle, 429

W

waterfall charts, *See* progressive column charts
Web browsers
  configuring, 29
Web-safe
  color palettes, 84
White Space, 430
Width (px), 430
widths
  Microsoft Excel limitations, 345
work areas, 25
  definition, 452
working
  with dimensions, 32
  with report objects, 27

X

X Axis, 377, 430
X Axis Measure, 377
XLS format
  limitations, 341
XML
  producing a report, 38
XY charts, *See* scatter charts

Y

Y Axis, 377, 378, 430
Y Axis 1, 378
Y Axis 2, 378
Y Axis Measure, 378
Y1 Axis, 430
Y2 Axis, 430
Y2 Axis Position, 430

Z

Z Axis, 378
Z Axis Measure, 379
Zero Value Characters, 436