

Advanced Reporting Guide:

Enhancing Your Business Intelligence Application

Version 10

To ensure that you are using the documentation that corresponds to the software you are licensed to use, compare this version number with the software version shown in "About MicroStrategy..." in the Help menu of your software.

Document number: 09451000

Copyright © 2015 by MicroStrategy Incorporated. All rights reserved.

If you have not executed a written or electronic agreement with MicroStrategy or any authorized MicroStrategy distributor (any such agreement, a "Separate Agreement"), the following terms apply:

This software and documentation are the proprietary and confidential information of MicroStrategy Incorporated and may not be provided to any other person. Copyright © 2001-2015 by MicroStrategy Incorporated. All rights reserved.

THIS SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" AND WITHOUT EXPRESS OR LIMITED WARRANTY OF ANY KIND BY EITHER MICROSTRATEGY INCORPORATED OR ANYONE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DISTRIBUTION OF THE SOFTWARE OR DOCUMENTATION, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, GOOD TITLE AND NON-INFRINGEMENT, QUALITY OR ACCURACY. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE AND DOCUMENTATION IS WITH YOU. SHOULD THE SOFTWARE OR DOCUMENTATION PROVE DEFECTIVE, YOU (AND NOT MICROSTRATEGY, INC. OR ANYONE ELSE WHO HAS BEEN INVOLVED WITH THE CREATION, PRODUCTION, OR DISTRIBUTION OF THE SOFTWARE OR DOCUMENTATION) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

In no event will MicroStrategy, Incorporated, or any other person involved with the creation, production, or distribution of the Software be liable to you on account of any claim for damage, including any lost profits, lost savings, or other special, incidental, consequential, or exemplary damages, including but not limited to any damages assessed against or paid by you to any third party, arising from the use, inability to use, quality, or performance of such Software and Documentation, even if MicroStrategy, Inc. or any such other person or entity has been advised of the possibility of such damages, or for the claim by any other party. In addition, MicroStrategy, Inc. or any other person involved in the creation, production, or distribution of the Software shall not be liable for any claim by you or any other party for damages arising from the use, inability to use, quality, or performance of such Software and Documentation, based upon principles of contract warranty, negligence, strict liability for the negligence of indemnity or contribution, the failure of any remedy to achieve its essential purpose, or otherwise. The entire liability of MicroStrategy, Inc. and your exclusive remedy, shall not exceed, at the option of MicroStrategy, Inc., either a full refund of the price paid, or replacement of the Software. No oral or written information given out expands the liability of MicroStrategy, Inc. beyond that specified in the above limitation of liability. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages, so the above limitation may not apply to you.

The information contained in this manual (the Documentation) and the Software are copyrighted and all rights are reserved by MicroStrategy, Inc. MicroStrategy, Inc. reserves the right to make periodic modifications to the Software or the Documentation without obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of the Software or Documentation without prior written consent of an authorized representative of MicroStrategy, Inc. are prohibited. U.S. Government Restricted Rights. It is acknowledged that the Software and Documentation were developed at private expense, that no part is public domain, and that the Software and Documentation are Commercial Computer Software provided with RESTRICTED RIGHTS under Federal Acquisition Regulations and agency supplements to them. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFAR 252.227-7013 et. seq. or subparagraphs (c)(1) and (2) of the Commercial Computer Software-Restricted Rights at FAR 52.227-19, as applicable. Contractor is MicroStrategy, Incorporated., 1850 Towers Crescent Plaza, Tysons Corner, VA 22182. Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.

The following terms and notices apply regardless of whether you have executed a Separate Agreement:

Trademark Information

MicroStrategy, MicroStrategy 9, MicroStrategy 9s, MicroStrategy Analytics Platform, MicroStrategy Desktop, MicroStrategy Analytics Express, MicroStrategy Analytics Enterprise, MicroStrategy Evaluation Edition, MicroStrategy Suite, MicroStrategy Web, MicroStrategy Mobile, MicroStrategy Server, MicroStrategy Parallel Relational In-Memory Engine (MicroStrategy PRIME), MicroStrategy MultiSource, MicroStrategy OLAP Services, MicroStrategy Intelligence Server, MicroStrategy Intelligence Server Universal, MicroStrategy Distribution Services, MicroStrategy Report Services, MicroStrategy Transaction Services, MicroStrategy Visual Insight, MicroStrategy Web Reporter, MicroStrategy Web Analyst, MicroStrategy Web Universal, MicroStrategy Office, MicroStrategy Data Mining Services, MicroStrategy Narrowcast Server, MicroStrategy Health Center, MicroStrategy Power User, MicroStrategy Analyst, MicroStrategy Developer, MicroStrategy Web Professional, MicroStrategy Architect, MicroStrategy SDK, MicroStrategy Command Manager, MicroStrategy Enterprise Manager, MicroStrategy Object Manager, MicroStrategy Integrity Manager, MicroStrategy System Manager, MicroStrategy Analytics App, MicroStrategy Mobile App, MicroStrategy Analytics for iPad® App, MicroStrategy Analytics Express App, MicroStrategy Tech Support App, MicroStrategy Mobile App Platform, MicroStrategy Mobile App Developer Academy, MicroStrategy Cloud, MicroStrategy Cloud Platform Services, MicroStrategy Cloud Data Hosting Services, MicroStrategy Cloud Data Warehouse Services, MicroStrategy Cloud Data Integration Services, MicroStrategy Virtual Business Intelligence (VBI) Appliance, MicroStrategy Cloud Paid Pilot, MicroStrategy R Integration, MicroStrategy Usher, Usher Badge, Usher Security, Usher Security Server, Usher Mobile, Usher Analytics, Usher Network Manager, MicroStrategy Trela for Retail (Alert backwards / used for demonstrations, MicroStrategy Services, MicroStrategy Professional Services, MicroStrategy Consulting, MicroStrategy Customer Services, MicroStrategy Education, MicroStrategy University, MicroStrategy Managed Services, MicroStrategy Business Intelligence QuickStrike, BI QuickStrike, Mobile QuickStrike, Transaction Services QuickStrike, Retail Vendor Portal, Perennial Education Pass, MicroStrategy Web Based Training (WBT), MicroStrategy World, Office Intelligence, Best in Business Intelligence, Pixel Perfect, Global Delivery Center, MicroStrategy Identity Platform, MicroStrategy Loyalty Platform, Direct Connect, Enterprise Grade Security For Every Business, Build Your Own Business Apps, Code-Free, Welcome to Ideal, The World's Most Comprehensive Analytics Platform, The World's Most Comprehensive Analytics Platform. Period.

Other product and company names mentioned herein may be the trademarks of their respective owners.

Specifications subject to change without notice. MicroStrategy is not responsible for errors or omissions. MicroStrategy makes no warranties or commitments concerning the availability of future products or versions that may be planned or under development.

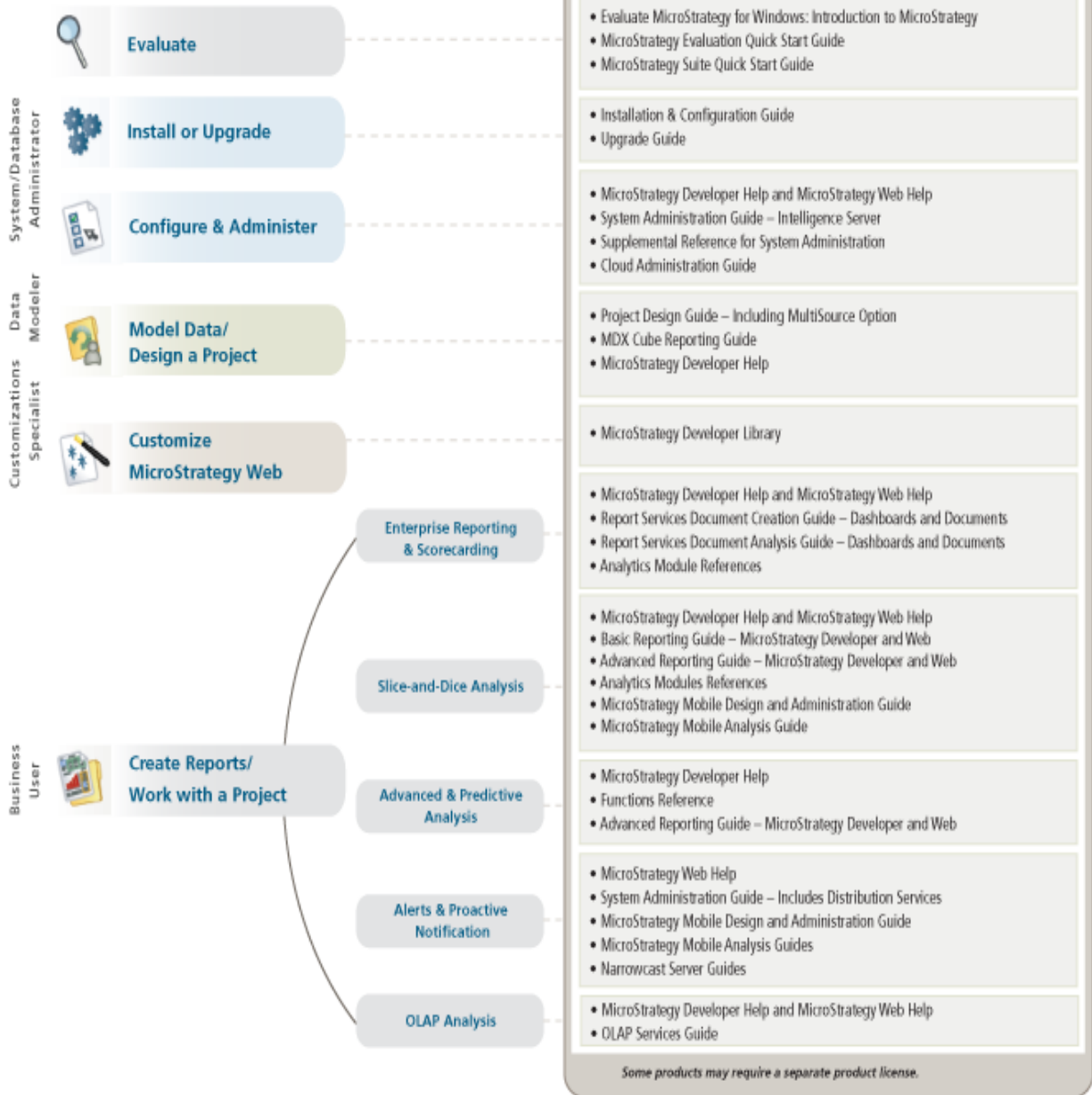
Patent Information

This product is patented. One or more of the following patents may apply to the product sold herein: U.S. Patent Nos. 5,321,520, 5,416,602, 5,748,560, 6,154,766, 6,173,310, 6,260,050, 6,263,051, 6,269,393, 6,279,033, 6,501,832, 6,567,796, 6,587,547, 6,606,596, 6,658,093, 6,658,432, 6,662,195, 6,671,715, 6,691,100, 6,694,316, 6,697,808, 6,704,723, 6,707,889, 6,741,980, 6,765,997, 6,768,788, 6,772,137, 6,788,768, 6,792,086, 6,798,867, 6,801,910, 6,820,073, 6,829,334, 6,836,537, 6,850,603, 6,859,798, 6,873,693, 6,885,734, 6,888,929, 6,895,084, 6,940,953, 6,964,012, 6,977,992, 6,996,568, 6,996,569, 7,003,512, 7,010,518, 7,016,480, 7,020,251, 7,039,165, 7,082,422, 7,113,474, 7,113,993, 7,127,403, 7,174,349, 7,181,417, 7,194,457, 7,197,461, 7,228,303, 7,260,577, 7,266,181, 7,272,212, 7,302,639, 7,324,942, 7,330,847, 7,340,040, 7,356,758, 7,356,840, 7,415,438, 7,428,302, 7,430,562, 7,440,898, 7,457,397, 7,486,780, 7,509,671, 7,516,181, 7,559,048, 7,574,376, 7,617,201, 7,725,811, 7,801,967, 7,836,178, 7,861,161, 7,861,253, 7,881,443, 7,925,616, 7,945,584, 7,970,782, 8,005,870, 8,035,382, 8,051,168, 8,051,369, 8,094,788, 8,130,918, 8,296,287, 8,321,411, 8,452,755, 8,521,733, 8,522,192, 8,577,902, 8,606,813, 8,607,138, 8,645,313, 8,761,659, 8,775,807, 8,782,083, 8,812,490, 8,832,588, 8,943,044, and 8,943,187. Other patent applications are pending.

Third Party Software

Various MicroStrategy products contain the copyrighted technology or software of third parties ("Third Party Software"). A list of Third Party Software, as well as links to any terms and conditions associated with such Third Party Software ("Third Party Terms"), can be found at www.microstrategy.com/third-party-notice. Your use of MicroStrategy products is subject to all applicable Third Party Terms.

WHAT DO YOU WANT TO DO WITH MICROSTRATEGY?



CONTENTS

Book Overview and Additional Resources	Description of this guide..... xvii
	About this book xix
	How to find business scenarios and examples xix
	What's new in this guide xx
	Prerequisites xx
	Who should use this guide xxi
	Resources xxi
	Documentation xxi
	Education xxix
	Consulting xxx
	Technical Support xxx
	Feedback xxxi
 1. Introduction to Advanced Reporting	 Introduction 1
	Before you begin 2
	Basic MicroStrategy terminology 2
	Moving to advanced reporting 5
	Importing data from different data sources 6
	Analyzing system information 8
 2. Advanced Metrics	 Introduction 11
Report Data and Calculations	Before you begin 12
	Level metrics: Modifying the context of data calculations 14
	Elements of a metric level 16

Metric level symbols: Defining expression syntax.....	17
Overview of level metrics	18
Target: The context of a calculation.....	22
Grouping: How aggregation is performed	23
Filtering: Interaction with report filters	30
Report level: Interaction with the context of reports.....	34
Level metrics review: Standard grouping.....	36
Level metrics review: No grouping.....	42
Using a hierarchy as the target of a metric level.....	47
Level metrics: Applying the metric condition to the metric calculation	53
Level metrics accepting dimensionality units to emulate MicroStrategy 6.x behavior	59
Nested metrics	60
Creating a nested metric.....	61
Conditional metrics	63
Conditional metric example.....	64
Conditional metrics with multiple qualifications.....	65
Example of a conditional metric with a level	65
Creating a conditional metric	67
Conditional metric and report filter interactions.....	70
Advanced options for metric filter and report filter interaction	89
Transformation metrics: Time-based and other comparisons	90
Year-to-date transformation example	93
Transformation metrics in the Metric Editor	94
Compound metrics.....	95
Compound metric example	96
Smart subtotals	98
Formula join type for compound metrics.....	99
Creating metrics by combining metrics, filters, and transformations	99
Metric subtotals.....	102
Displaying subtotals on reports.....	102
Defining subtotal defaults	103
Disabling subtotals.....	111
Metric functions.....	112
Rank	112
Count	114
Running and moving sums and averages.....	115
N-tile	117
First and Last	119

	Creating your own plug-in functions.....	119
	Integrating R analytics into MicroStrategy	120
	Apply functions: Using database-specific functions in metrics	120
	Join specifications.....	121
	Inner joins versus outer joins	121
	Joins between metrics	122
	Metric-specific VLDB properties: SQL and analytical customization	123
	Metric column aliases: SQL identifiers and data types	123
3. Advanced Filters		
Filtering Data on Reports	Introduction.....	125
	Before you begin.....	126
	Types of qualifications	127
	Advanced filter topics.....	128
	Attribute-to-attribute qualification	130
	Creating attribute-to-attribute qualification filters	130
	Example: Customers who live in the same city as call centers	131
	Example: Filtering on an income bracket.....	133
	Dynamic dates: Filtering by offsetting the current date.....	135
	Dynamic date example	136
	Dynamic date based on a prompt answer	139
	Importing and exporting elements for attribute qualifications	143
	The attribute level of set qualifications: Output level	146
	The output level of relationship qualifications	147
	The output level of metric qualifications.....	148
	Resetting rank and percent metric qualifications: Break by	149
	Metric-to-metric comparisons	150
	Merging attribute qualifications	151
	Applying qualifications independently of relationship filters.....	153
	Custom expressions and database-specific functions	154
	Custom expressions	154
	Apply functions in metrics: Using database-specific functions	164
	Joint element lists: Grouping elements from multiple attributes	172

Prompted filters: Asking for user input for filter conditions.....	174
Prompted attribute qualifications.....	175
Prompted metric qualifications.....	175
Report object prompts	176
Filter object prompts	176
Shortcut vs. embedded filters	177
Adding a stand-alone filter to a report.....	177
Saving a report with a shortcut to a stand-alone filter.....	178
 4. Custom Groups and Consolidations	
Grouping Attribute Elements	
Introduction.....	181
About custom groups: Grouping attribute elements with banding and filtering	182
About custom group elements	183
Custom group examples	185
Custom group effects on SQL generation.....	187
Benefits of using a custom group	187
Custom group structure on reports	191
Creating a custom group	193
Editing and formatting a custom group	194
Changing the position of element headers	195
Enabling subtotals for custom groups.....	196
Sorting custom groups	197
Determining custom group interaction with the report filter	200
Prompted custom groups.....	201
Editing custom group elements.....	202
Editing the conditions in a custom group element	204
Deleting a custom group, element, or condition	206
Consolidations: Grouping attribute elements with arithmetic and logical operators	207
Create a virtual attribute	207
Perform row level math	208
Consolidation elements	209
Evaluation order.....	213
Consolidations and SQL	214
Consolidations and subtotals.....	215
Consolidation example	220
Custom group and consolidation comparison.....	221
Arithmetic operations support (row level math).....	222

	Where and how data is calculated	222
	SQL efficiency	223
	Re-using consolidation elements	223
	Hierarchical display support.....	223
	Subtotal support.....	223
5. Drill Maps		
Customizing Drill Paths	Introduction.....	225
	What is drilling?	225
	Drill maps and drill paths.....	226
	Default drill paths: System hierarchy drill paths	227
	Filters and drilling.....	227
	Creating custom drill maps and paths	228
	The destination of the drill.....	229
	Drill path types: Drilling up, down, across, or to a template	229
	Drill path settings	230
	Drill map association: Objects containing the drill map	238
	Order of precedence of drill maps: Levels	239
	Removing associations	240
6. Advanced Prompts		
Enabling User Input	Introduction.....	241
	What is a prompt?.....	242
	Prompts in scheduled reports	242
	Types of prompts	243
	Filtering data on attributes, attribute forms, attribute elements, or metrics: Filter definition prompts	244
	Filtering data based on metrics, attributes, or other objects: Object prompts	246
	Value prompts.....	249
	Defining the dimensionality of metrics: Level prompts.....	251
	Prompting for a user's login name: System prompts	258
	Defining how a prompt is displayed in MicroStrategy Web.....	258
	Display styles for prompts in MicroStrategy Web	259
	Configuring how the qualification can be defined in MicroStrategy Web prompts: Expression settings	265

7. Designing Reports

Understanding data
evaluation order and
distributing reports to users

Introduction.....	271
Before you begin.....	272
Report design versus report creation.....	272
Accessing the Basic Report: Foundation of examples	273
Filters and order of calculation on reports	274
About filters.....	274
The difference between report filters and report limits.....	276
About metric qualifications.....	280
About the report-as-filter	283
Defining a filter for a drilled-to report.....	284
The basics of report execution.....	286
Data definition and view definition objects: Data versus display.....	287
Intelligent Cubes	288
Totaling and subtotalling data on reports	288
About subtotals	289
Custom report subtotals.....	294
Subtotals and memory usage	297
Sorting data on reports	298
Formatting a report	309
Formatting report cell data	310
Understanding how formatting impacts report display	317
Order of layers	322
Autostyles: Report formatting styles	328
Find and replace report, template, and metric formatting	329
Displaying attribute and attribute form headers in a grid	330
Determining which attribute elements are displayed: Attribute joins	335
Understanding attribute join types	336
Selecting an attribute join type.....	339
Evaluation order of objects on reports	342
Default evaluation order.....	343
Defining an evaluation order	344
Evaluation order in data definition and view definition	344
Adding features for users.....	351
Distributing reports to a user community	352
Re-using objects via shortcuts and object templates.....	352
Making reports accessible	361
Caching.....	361

	Exporting large reports as delimited text files	363
8. Graphing	Introduction.....	367
	Graphing best practices and troubleshooting	368
	Viewing graph reports.....	371
	Adding an object to a graph report	372
	Choosing an appropriate graph style.....	374
	Minimum object requirements for each graph style	374
	Placement of report objects and choice of graph styles	376
	Moving objects on a graph report	378
	Displaying graph data without scrollbars	380
	Graph style usage and examples	381
	Area	382
	Bar	384
	Line	392
	Pie chart.....	395
	Scatter plot.....	398
	Polar chart	400
	Radar chart	402
	Bubble chart.....	405
	Combination.....	407
	Custom	408
	Advanced graph style usage and examples	408
	Stock.....	409
	Histogram	414
	3D Surface	416
	Gauge	418
	Funnel.....	421
	Pareto chart	424
	Boxplot.....	427
	Gantt chart	430
	Formatting a graph report.....	434
	Controlling how a graph report appears.....	435
	Formatting a graph using colors, shading, and effects	435
	Manually formatting graph series and objects	451
	Creating and formatting titles.....	454
	Formatting the axes on a graph	461
	Formatting numeric values on a graph	468
	Displaying and formatting data labels.....	474
	Displaying and formatting a graph legend	476

	Formatting the labels on a graph's axes	480
	Creating tooltips in a graph	481
	Adjusting the layout, size, and scale of a graph.....	484
	Saving and applying a custom graph style	499
	Choosing a graph image format for HTML documents and Web	503
	Exporting and sharing graphs.....	504
	Exporting graphs.....	504
	Copying graphs into third-party applications.....	505
	Using interactive graph visualizations in documents: Widgets ..	507
9. Displaying Information about Reports: Report Details	Introduction.....	509
	Displaying report details	510
	Enabling and disabling report details	511
	Levels of report details configuration	511
	Interaction of report details preferences and report details for reports	512
	Configuring report details settings	513
	Configuring the display of object name delimiters for report details	514
	Configuring report details.....	516
	Configuring the display of template details	520
	Including objects from the base or view report	521
	Displaying the base template name.....	523
	Interaction of report details preferences and template details for reports	526
	Configuring the prompt details	530
	Prompt details: Examples	531
	Interaction of report details preferences and prompt details for reports	533
	Configuring the filter details	537
	Interaction of report details preferences and filter details for reports	538
	Configuring the content of the filter details.....	538
	Configuring how attribute element list qualifications are displayed.....	544
	Configuring how attribute form and set qualifications are displayed.....	546
	Configuring how logical operators are displayed	548
	Configuring whether to use aliases in the filter details	551
	Configuring the display of filter details: Procedure.....	551

	Enabling and customizing report details preferences	554
10. Linking Reports and Documents	Introduction	559
	Linking reports and documents.....	559
	Prerequisites for working with links.....	562
	About links	563
	Components of a link	564
	Specifying how prompts are answered in the target.....	567
	Creating links	583
	Copying links to other objects in the report.....	587
	Creating linked reports.....	589
	Creating linked reports that answer prompts with the selected objects	589
	Creating linked reports that answer prompts using metric conditions.....	592
11. Custom SQL Queries: Freeform SQL and Query Builder	Introduction	603
	Customize your SQL statements: Freeform SQL	604
	Usage scenarios and SQL standards	605
	Access and analyze multiple data sources with Freeform SQL.....	613
	Reporting analysis features	623
	Security for data access	632
	Reporting on third-party web services with XQuery.....	638
	Allowing connections to web services in a project.....	639
	Using XQuery to retrieve data from a web service	641
	Creating a report to analyze web service data.....	659
	Supporting Grid/Graph filtering for document datasets based on large XQuery reports.....	661
	Retrieving web folder resources with XQuery.....	664
	Creating an external mobile folder to allow web folder browsing	667
	Creating XQuery reports to retrieve web folder resources...	668
	Updating data with Transaction Services	675
	Creating a Transaction Services report	676
	Customize your SQL queries: Query Builder.....	694
	Usage scenarios	696
	Access and analyze multiple data sources with Query Builder.....	699
	Reporting analysis features	717

	Security for data access	727
	Connect to databases, Excel files, and text files	731
	Connect to databases	732
	Connect to Excel files	733
	Connect to text files	734
	Map data to non-project objects: Managed objects	735
	Access managed objects	737
	Edit managed objects	739
	Delete managed objects	739
	Creating Intelligent Cubes	740
12. Accessing Subsets of Data: Data Marts	Introduction	743
	Getting started with data marts	744
	Creating data marts	744
	Using a data mart table as a source table	749
	Enabling data mart optimization	754
	Running a standard report against a data mart	756
	Specifying governors, table creation options, and custom SQL statements	757
	Specifying governors and table creation options	758
	Specifying custom SQL statements	759
13. Data Mining Services	Introduction	763
	Data Mining Services	764
	Approaches for data mining with MicroStrategy	764
	The Data Mining Services workflow	769
	Predictive metrics and performance	770
	Creating a dataset report	771
	Data mining dataset reports	772
	Guidelines for creating a dataset report	774
	Inputs for predictive metrics	776
	Using non-MicroStrategy dataset reports	784
	Creating a predictive model	785
	Using MicroStrategy	785
	Using third-party data mining applications	823
	Importing the predictive model	823
	Importing predictive models for MDX cubes	825
	Aggregating predictive metrics	828

Using the predictive metric	829
Using the predictive metric in reports and documents	829
Using the predictive metric in other objects	830
Predictive Model Viewer	830
Data mining examples	831
Revenue forecasting example (using linear and seasonal regression)	832
Campaign management example (using logistic regression)	836
Segmentation example (using cluster analysis)	838
Telco churn example (using decision tree analysis)	841
Campaign management example: Importing a third-party neural network model	844
Market basket analysis (using association rules analysis)	848
 14. VLDB Properties	
Introduction	859
Levels of VLDB properties	860
VLDB properties at the metric level	861
Integer Constant In Metric	861
Metric Join Type	862
Null Check	862
Zero Check	863
Null checking for Analytical Engine	863
Subtotal Dimensionality Aware	863
VLDB properties at the report level	865
Cartesian Join Warning	866
Full Outer Join Support	868
Preserve All Final Pass Result Elements	868
Preserve All Lookup Table Elements	869
Report Post Statement	872
Report Pre Statement	872
Table Post Statement	873
Sub Query Type	873
WHERE Clause Driving Table	874
Attribute and Attribute Form Selection Option for Intermediate Passes properties	875
GROUP BY ID Attribute	877
GROUP BY Non-ID Attribute	878
SQL Hint	878
UNION Multiple INSERT	878
Drop Temp Table Method	878
FallBack Table Type	879

	Intermediate Table Type	879
	Table Option and Table Space properties	880
	VLDB properties at the database instance level	882
	Maximum SQL/MDX Size	882
	SQL Date Format.....	883
A. Logical and Mathematical Operators for Filtering	Introduction.....	885
	What is an operator?	885
	Logical operators	886
	Comparison operators	889
	Rank and percent operators	890
	Pattern operators	891
B. Formatting Default Values	Introduction.....	893
	Default values	894
	Number	894
	Alignment.....	894
	Font.....	895
	Border	895
	Patterns	895
	Banding.....	896
C. MicroStrategy Developer Commands	Introduction.....	899
	Basics	900
	Enabling the MicroStrategy Developer homepage	900
	Viewing the MicroStrategy Developer commands	902
	Commands	902
	ChangeView	903
	Editor	904
	Execute	905
	ExecuteDocument	906
	ExecuteReport	906
	Open	907
	Reset	908
	Shortcut	909

D. Sorting and Browsing Standards	Introduction..... 911
	Sort order of characters in MicroStrategy 912
	Project order in MicroStrategy Developer and Web 913
Glossary.....	915
Index.....	939

BOOK OVERVIEW AND ADDITIONAL RESOURCES

Description of this guide

Thank you for using MicroStrategy 9—Platform for Mobile Intelligence.

The *MicroStrategy Advanced Reporting Guide* provides comprehensive information on advanced topics for using the MicroStrategy query and reporting products. This guide builds on a basic understanding of information provided in the *MicroStrategy Basic Reporting Guide*. Advanced topics include the following:

- [Chapter 1, Introduction to Advanced Reporting](#), reviews the tasks and objects to create a basic report and previews advanced reporting concepts.
- [Chapter 2, Advanced Metrics](#), explains level metrics, conditional metrics, transformation metrics, and compound metrics, which provide complex business measures and key performance indicators. Numerous examples provide real-life context for the explanations.
- [Chapter 3, Advanced Filters](#), describes how to create complex filters and how to apply them to reports.
- [Chapter 4, Custom Groups and Consolidations](#), introduces and compares custom groups and consolidations and why to use each.

- [*Chapter 5, Drill Maps*](#), describes how to customize drill paths to allow users to access additional information from reports.
- [*Chapter 6, Advanced Prompts*](#), explains how to create dynamic report definitions through prompts.
- [*Chapter 7, Designing Reports*](#), guides you through advanced reporting concepts in a hands-on way, providing a tutorial experience. The sample reports illustrate various reporting concepts such as the difference between report filters and report limits, while showing you how reports are built and generated. The chapter also introduces the concepts of the view definition and the data definition of a report, detailing the report execution steps that belong to each.
- [*Chapter 8, Graphing*](#), discusses how to view, format, and manipulate MicroStrategy graph reports, introduces the graph styles available in MicroStrategy, and provides tips on how you can most effectively present your report data in a graph.
- [*Chapter 9, Displaying Information about Reports: Report Details*](#), discusses how to customize the report details, which is the information that appears in the Report Details pane. For example, you can select whether to include view filter information or the attribute name in the filter details. For report details, you can choose whether to include information on prompts or filters.
- [*Chapter 10, Linking Reports and Documents*](#), discusses how to create and use links. Links allow analysts to explore additional, related data by executing another document or report from within a document or report, in either MicroStrategy Developer or MicroStrategy Web. A link allows context to be passed to any prompted report or document, whether the destination report or document is related to the original.
- [*Chapter 11, Custom SQL Queries: Freeform SQL and Query Builder*](#), discusses how to use Freeform SQL to access data from various data sources, including relational databases, Excel files, and flat files.
- [*Chapter 12, Accessing Subsets of Data: Data Marts*](#), explains how to use relational tables known as data mart tables to create or update warehouse tables or integrate with third-party tools.
- [*Chapter 13, Data Mining Services*](#), presents the data mining process, which leverages existing and historical data to create predictive models to project potential outcomes for business activities and transactions.

- [Chapter 14, VLDB Properties](#), describes the most commonly-used VLDB properties relating to metrics and reports. VLDB properties allow you to customize the SQL that MicroStrategy generates.



You can also report on data returned from MDX cube sources such as SAP BW, Microsoft Analysis Services, and Hyperion Essbase. For information on integrating MicroStrategy with these MDX cube sources, see the *MDX Cube Reporting Guide*.

The appendixes contain the following additional reference information:

- [Appendix A, Logical and Mathematical Operators for Filtering](#), discusses operators and how to use them in filters and report limits.
- [Appendix B, Formatting Default Values](#), provides the default values for all the formatting options, to assist you in formatting reports and creating autostyles.
- [Appendix C, MicroStrategy Developer Commands](#), specifies the Developer commands used in MicroStrategy products, focusing on Developer home page usage and describing the commands from an HTML perspective.
- [Appendix D, Sorting and Browsing Standards](#), describes the sorting and browsing standards for characters and projects in MicroStrategy Developer and Web.

About this book

The following sections provide the location of examples, list prerequisites for using this book, and describe the user roles the information in this book was designed for.



The sample documents and images in this guide, as well as some example steps, were created with dates that may no longer be available in the MicroStrategy Tutorial project. If you are re-creating an example, replace the year(s) shown in this guide with the most recent year(s) available in the software.

How to find business scenarios and examples

Within this guide, many of the concepts discussed are accompanied by business scenarios or other descriptive examples. Many of the examples use

the MicroStrategy Tutorial, which is MicroStrategy's sample warehouse and project. Information about the MicroStrategy Tutorial can be found in the *Basic Reporting Guide*.

What's new in this guide

Analytics Enterprise

The name of MicroStrategy Desktop has been changed to MicroStrategy Developer.

MicroStrategy 9.4

- The examples for creating and formatting stock graphs have been updated for additional best practices information and requirements, which are provided in [Hi-Lo-Open-Close Stock graph example, page 411](#) and [Using Stock graphs effectively, page 412](#).
- Report Services document designers can enable Grid/Graphs as report condition selectors to filter the data in a dataset, allowing users to view subsets of large amounts of data, rather than loading and displaying all the data at once. This can be supported for datasets created using Freeform SQL and XQuery, as described in [Supporting Grid/Graph filtering of document datasets based on Freeform SQL reports, page 628](#) and [Supporting Grid/Graph filtering for document datasets based on large XQuery reports, page 661](#) respectively.

Prerequisites

Before working with this document, you should be familiar with the information in the *Basic Reporting Guide*, including:

- MicroStrategy Developer (formerly MicroStrategy Desktop)
- Projects, attributes, and facts (covered in the *Project Design Guide*)
- Simple metric and report creation
- Report manipulation such as formatting, drilling, and subtotals
- Prompts and filters

- SQL statements (basic understanding)

Who should use this guide

This document is designed for:

- Report designers who create advanced reports and advanced reporting objects such as metrics, filters, custom groups, consolidations, and drill maps.
- Analysts who use advanced report manipulations such as data marts, Freeform SQL reports, MDX cube reports, and data mining.

Resources

This section provides details on how to access books, online help, MicroStrategy Education and Consulting resources, and how to contact MicroStrategy Technical Support.

Documentation

MicroStrategy provides both manuals and online help; these two information sources provide different types of information, as described below:

- **Manuals:** MicroStrategy manuals provide:
 - Introductory information and concepts
 - Examples and images
 - Checklists and high-level procedures to get started

The steps to access the manuals are described in [Accessing manuals and other documentation sources, page xxvii](#).

Most of these manuals are also available printed in a bound, soft cover format. To purchase printed manuals, contact your MicroStrategy Account Executive with a purchase order number.

- **Help:** MicroStrategy online help provides:
 - Detailed steps to perform procedures

- Descriptions of each option on every software screen

Additional formats

MicroStrategy manuals are available as electronic publications, downloadable on the Apple iBooks Store or Google Play, and can be read on your iOS or Android device respectively. To download a book, search for the book's title in the iBookstore or Google Play. To view a list of manuals that are currently available, scan the following QR codes using your device's camera:

- For iOS devices, scan the following QR code:



- For Android devices, scan the following QR code:



For new MicroStrategy releases, it may take several days for the latest manuals to be available on the iBookstore or Google Play.

Translations

For the most up-to-date translations of MicroStrategy documentation, refer to the MicroStrategy Knowledge Base. Due to translation time, manuals in languages other than English may contain information that is one or more releases behind. You can see the version number on the title page of each manual.

Finding information

You can search all MicroStrategy books and Help for a word or phrase, with a simple Google™ search at <http://www.google.com>. For example, type “MicroStrategy derived metric” or “MicroStrategy logical table” into a Google search. As described above, books typically describe general concepts and examples; Help typically provides detailed steps and screen options. To limit your search to MicroStrategy books, on Google’s main page you can click **More**, then select **Books**.

Manuals for MicroStrategy overview and evaluation

- *Introduction to MicroStrategy: Evaluation Guide*

Instructions for installing, configuring, and using the MicroStrategy Evaluation Edition of the software. This guide includes a walkthrough of MicroStrategy features so you can perform reporting with the MicroStrategy Tutorial project and its sample business data.

- *MicroStrategy Evaluation Edition Quick Start Guide*

Overview of the installation and evaluation process, and additional resources.

Resources for security

- *Usher Help*

Steps to perform mobile identity validation using the Usher mobile security network to issue electronic badges for identifying users.

Manuals for query, reporting, and analysis

- *MicroStrategy Installation and Configuration Guide*

Information to install and configure MicroStrategy products on Windows, UNIX, Linux, and HP platforms, and basic maintenance guidelines.

- *MicroStrategy Upgrade Guide*

Steps to upgrade existing MicroStrategy products.

- *MicroStrategy Project Design Guide*

Information to create and modify MicroStrategy projects, and create the objects that present your organization's data, such as facts, attributes, hierarchies, transformations, advanced schemas, and project optimization.

- *MicroStrategy Basic Reporting Guide*

Steps to get started with MicroStrategy Web, and how to analyze and format data in a report. Includes the basics for creating reports, metrics, filters, and prompts.

- *MicroStrategy Advanced Reporting Guide: Enhancing Your Business Intelligence Application*

Steps to create Freeform SQL reports, Query Builder reports, complex filters and metrics, use Data Mining Services, and create custom groups, consolidations, and complex prompts.

- *Document and Dashboard Analysis Guide*

Steps to execute, analyze, and format a dashboard in MicroStrategy Web.

- *MicroStrategy Report Services Document Creation Guide: Creating Boardroom Quality Documents*

Steps to create Report Services documents, add objects, and format the document and its objects.

- *MicroStrategy Dashboards and Widgets Creation Guide: Creating Interactive Dashboards for Your Data*

Steps to create MicroStrategy Report Services dashboards and add interactive visualizations.

- *MicroStrategy In-memory Analytics Guide*

Information to use MicroStrategy OLAP Services features, including Intelligent Cubes, derived metrics, derived elements, dynamic aggregation, view filters, and dynamic sourcing.

- *MicroStrategy Office User Guide*

Instructions to use MicroStrategy Office to work with MicroStrategy reports and documents in Microsoft® Excel, PowerPoint, and Word, to analyze, format, and distribute business data.

- *MicroStrategy Mobile Analysis Guide: Analyzing Data with MicroStrategy Mobile*

Steps to use MicroStrategy Mobile to view and analyze data, and perform other business tasks with MicroStrategy reports and documents on a mobile device.

- *MicroStrategy Mobile Design and Administration Guide: A Platform for Mobile Intelligence*

Information and instructions to install and configure MicroStrategy Mobile, as well as steps for a designer working in MicroStrategy Developer or MicroStrategy Web to create effective reports and documents for use with MicroStrategy Mobile.

- *MicroStrategy System Administration Guide: Tuning, Monitoring, and Troubleshooting Your MicroStrategy Business Intelligence System*

Steps to implement, deploy, maintain, tune, and troubleshoot a MicroStrategy business intelligence system.

- *MicroStrategy Supplemental Reference for System Administration: VLDB Properties, Internationalization, User Privileges, and other Supplemental Information for Administrators*

Steps for administrative tasks such as configuring VLDB properties and defining data and metadata internationalization, and reference material for other administrative tasks.

- *MicroStrategy Functions Reference*

Function syntax and formula components; instructions to use functions in metrics, filters, attribute forms; examples of functions in business scenarios.

- *MicroStrategy MDX Cube Reporting Guide*

Information to integrate MicroStrategy with MDX cube sources. You can integrate data from MDX cube sources into your MicroStrategy projects and applications.

- *MicroStrategy Operations Manager Guide*

Instructions for managing, monitoring, and setting alerts for all of your MicroStrategy systems from one console. This guide also includes instructions for setting up and using Enterprise Manager to analyze your MicroStrategy system usage.

Manual for the Human Resources Analytics Module

- *Human Resources Analytics Module Reference*

Software Development Kits

- *MicroStrategy Developer Library (MSDL)*

Information to understand the MicroStrategy SDK, including details about architecture, object models, customization scenarios, code samples, and so on.

- *MicroStrategy Web SDK*



The Web SDK is available in the MicroStrategy Developer Library, which is part of the MicroStrategy SDK.

Documentation for MicroStrategy Portlets

- *Enterprise Portal Integration Help*

Information to help you implement and deploy MicroStrategy BI within your enterprise portal, including instructions for installing and configuring out-of-the-box MicroStrategy Portlets for several major enterprise portal servers.

This resource is available from <http://www.microstrategy.com/producthelp>.

Documentation for MicroStrategy GIS Connectors

- *GIS Integration Help*

Information to help you integrate MicroStrategy with Geospatial Information Systems (GIS), including specific examples for integrating with various third-party mapping services.

This resource is available from <http://www.microstrategy.com/producthelp>.

Help

Each MicroStrategy product includes an integrated help system to complement the various interfaces of the product as well as the tasks that can be accomplished using the product.

Some of the MicroStrategy help systems require a web browser to be viewed. For supported web browsers, see the MicroStrategy Readme.

MicroStrategy provides several ways to access help:

- **Help button:** Use the Help button or ? (question mark) icon on most software windows to see help for that window.
- **Help menu:** From the Help menu or link at the top of any screen, select MicroStrategy Help to see the table of contents, the Search field, and the index for the help system.
- **F1 key:** Press F1 to see context-sensitive help that describes each option in the software window you are currently viewing.



For MicroStrategy Web, MicroStrategy Web Administrator, and MicroStrategy Mobile Server, pressing the F1 key opens the context-sensitive help for the web browser you are using to access these MicroStrategy interfaces. Use the Help menu or ? (question mark) icon to access help for these MicroStrategy interfaces.

Accessing manuals and other documentation sources

The manuals are available from <http://www.microstrategy.com/producthelp>, as well as from your MicroStrategy disk or the machine where MicroStrategy was installed.



Adobe Reader is required to view these manuals. If you do not have Adobe Reader installed on your computer, you can download it from <http://get.adobe.com/reader/>.

The best place for all users to begin is with the *MicroStrategy Basic Reporting Guide*.

To access the installed manuals and other documentation sources, see the following procedures:

- [To access documentation resources from any location, page xxviii](#)
- [To access documentation resources on Windows, page xxviii](#)

- *To access documentation resources on UNIX and Linux, page xxviii*

To access documentation resources from any location

- 1 Visit <http://www.microstrategy.com/producthelp>.

To access documentation resources on Windows

- 1 From the Windows **Start** menu, choose **Programs** (or **All Programs**), **MicroStrategy Documentation**, then **Product Manuals**. A page opens in your browser showing a list of available manuals in PDF format and other documentation sources.
- 2 Click the link for the desired manual or other documentation source.



If bookmarks are not visible on the left side of a product manual, from the **View** menu click **Bookmarks and Page**. This step varies slightly depending on your version of Adobe Reader.

To access documentation resources on UNIX and Linux

- 1 Within your UNIX or Linux machine, navigate to the directory where you installed MicroStrategy. The default location is `/opt/MicroStrategy`, or `$HOME/MicroStrategy/install` if you do not have write access to `/opt/MicroStrategy`.
- 2 From the MicroStrategy installation directory, open the `Help` folder.
- 3 Open the `Product_Manuals.htm` file in a web browser. A page opens in your browser showing a list of available manuals in PDF format and other documentation sources.
- 4 Click the link for the desired manual or other documentation source.



If bookmarks are not visible on the left side of a product manual, from the **View** menu click **Bookmarks and Page**. This step varies slightly depending on your version of Adobe Reader.

Documentation standards

MicroStrategy online help and PDF manuals (available both online and in printed format) use standards to help you identify certain types of content. The following table lists these standards.



These standards may differ depending on the language of this manual; some languages have rules that supersede the table below.

Type	Indicates
bold	Button names, check boxes, options, lists, and menus that are the focus of actions or part of a list of such GUI elements and their definitions Example: Click Select Warehouse.
<i>italic</i>	- Names of other product manuals and documentation resources - When part of a command syntax, indicates variable information to be replaced by the user Example: Type <i>copy c:\filename d:\foldername\filename</i>
Courier font	- Calculations - Code samples - Registry keys - Path and file names - URLs - Messages displayed in the screen - Text to be entered by the user Example: <code>Sum(revenue)/number of months.</code> Example: Type <code>cmdmgr -f scriptfile.scp</code> and press Enter.
+	A keyboard command that calls for the use of more than one key (for example, SHIFT+F1).
	A note icon indicates helpful information for specific situations.
	A warning icon alerts you to important information such as potential security risks; these should be read before continuing.

Education

MicroStrategy Education Services provides a comprehensive curriculum and highly skilled education consultants. Many customers and partners from over 800 different organizations have benefited from MicroStrategy instruction.

Courses that can help you prepare for using this manual or that address some of the information in this manual include:

- MicroStrategy Developer: Reporting Essentials

- MicroStrategy Developer: Advanced Reporting
- MicroStrategy Web for Reporters and Analysts
- MicroStrategy Web for Professionals
- MicroStrategy Freeform SQL Essentials

For the most up-to-date and detailed description of education offerings and course curricula, visit <http://www.microstrategy.com/Education>.

Consulting

MicroStrategy Consulting Services provides proven methods for delivering leading-edge technology solutions. Offerings include complex security architecture designs, performance and tuning, project and testing strategies and recommendations, strategic planning, and more. For a detailed description of consulting offerings, visit <http://www.microstrategy.com/services-support/consulting>.

Technical Support

If you have questions about a specific MicroStrategy product, you should:

- 1 Consult the product guides, Help, and readme files. Locations to access each are described above.
- 2 Consult the MicroStrategy Knowledge Base online at <https://resource.microstrategy.com/support>.



A technical administrator in your organization may be able to help you resolve your issues immediately.

- 3 MicroStrategy Technical Support can be contacted by your company's Support Liaison. Contact information and the Technical Support policy information is available at <http://www.microstrategy.com/services-support/support/contact>.

Feedback

Please send any comments or suggestions about user documentation for MicroStrategy products to:

`documentationfeedback@microstrategy.com`

Send suggestions for product enhancements to:

`support@microstrategy.com`

When you provide feedback to us, please include the name and version of the products you are currently using. Your feedback is important to us as we prepare for future releases.

INTRODUCTION TO ADVANCED REPORTING

Introduction

Advanced reporting allows you to create more sophisticated reports using advanced report functionality such as data marting and Freeform SQL, and advanced objects such as level metrics, conditional metrics, prompted filters, custom groups, and drill maps. Examples of advanced reporting in action include:

- The contribution to revenue and profit of products bought by your top customers
- The quarterly revenue, split by regions
- The inventory for time periods you select at report run-time
- A comparison of new customers this year to last year
- The revenue for electronics in the first quarter of 2003 and music in the third quarter of 2003, or other attribute combinations you can select at report runtime
- Allowing users to drill only to the call center level, not the employee level, for a salary report

An overview of the advanced reporting features is provided in [Moving to advanced reporting, page 5](#). By the end of this chapter, you should understand what is involved in creating a basic report and have an idea of what advanced reporting can do for you.



Warehouse data in the sample MicroStrategy projects is updated regularly, and these changes are reflected in the documentation whenever possible. However, the sample reports, documents, objects, and images in this guide may display warehouse data that no longer appears in the software.

Before you begin

Before you use this guide, you should be familiar with the report concepts and processes described in the *Basic Reporting Guide*. Advanced reporting features build on the concepts and procedures presented there by providing more technical details and advanced options for report design.

The *Project Design Guide* also contains useful reference material, focused on schema objects such as facts and attributes.

Basic MicroStrategy terminology

Facts

Facts are the MicroStrategy objects on which metrics are based. A fact has two characteristics: it is numerical and aggregatable. Examples of facts include revenue, inventory, and account balances.

Facts are stored in tables in the data warehouse. These fact tables comprise different columns, each cell representing a specific piece of information. You build facts in a MicroStrategy project that point to the columns. Metrics, which are business measures, are then created from the facts.

SQL aggregations, such as `SUM` and `AVG`, are performed on the facts in the database tables. For example, in the following SQL statement, the `ORDER_AMT` column in the warehouse might correspond to the Order Amount fact in the MicroStrategy environment:

```
SELECT sum(a21.ORDER_AMT) REGION
```



```
FROM ORDER_FACTa21
JOIN LU_EMPLOYEEa22
ON (a21.EMP_ID = a22.EMP_ID)
WHERE a22.CALL_CTR_ID in (5, 9, 12)
```

In this example, ORDER_AMT is the fact, whereas sum(a21.ORDER_AMT) represents a metric.

For background information on facts and steps to create them, see the *Project Design Guide*.

Attributes

Attributes act as holders of information, allowing you to add context to your facts in a report. For example, if you had \$10,000 in revenue, that number does not mean anything in a business sense unless you know the context, such as which region, the designated time frame for the sales, and the labor involved in the transaction. Simply put, attributes provide categories for the summarization of data.

For background information on attributes, such as procedures to create them, see the *Project Design Guide*.

Attribute elements

Attribute elements are the data shown on the report. Think of them as a sub-level of the attribute. For example, if City is the attribute, attribute elements can include London, Milan, and New York.

In the data warehouse, attributes are usually represented by columns in a table, and attribute elements are represented by the rows.



Metrics

Metrics are analytical calculations performed against stored data (facts) to produce results that can then either be read as status material or analyzed for decision-making purposes. They are similar to formulas in spreadsheet software. Metrics represent business measures and key performance indicators. A metric can calculate revenue, inventory levels, employee counts, or visits to a Web page.

Advanced metrics are discussed in [Chapter 2, Advanced Metrics](#).

Prompts

A *prompt* is used to dynamically modify the contents of a report. With prompts, you can determine, during report execution, the objects to retrieve for the report and report filtering conditions. In addition, you can make different prompt selections each time you run the report.

For example, you can create a prompt for filtering criteria that qualifies on the Year attribute. When you run a report with this prompt in its report filter, you are prompted to select the year for which you want the report results. You can run the report the first time by selecting 2003 and then a second time by selecting 2004.

Advanced prompts are discussed in [Chapter 6, Advanced Prompts](#).

Reports

A *report* is a MicroStrategy object that represents a request for a specific set of formatted data from the data warehouse. Reports are the focus and goal of business intelligence. They allow users to gather business insight through data analysis.

The different parts of a report include:

- Facts and attributes from the warehouse
- Filters that determine how much data is used to generate the report
- Metrics to perform calculations on the facts

Advanced report objects such as level metrics, transformation metrics, consolidations, custom groups, and drill maps, among others, allow you to create more functional and informative reports. You can also use your own

customized SQL statements to generate reports from operational systems included in a MicroStrategy project. This is known as Freeform SQL. You can create predictive models based on existing and historical data to project potential outcomes for business activities and transactions. MicroStrategy Data Mining Services facilitates the development and deployment of these predictive models. All this advanced functionality is described in detail in this guide.

Moving to advanced reporting

Before you begin working with advanced reporting functionality, you must have a working project containing schema objects such as attributes and facts. For information on creating projects, facts, and attributes, refer to the *Project Design Guide*.

The *Project Design Guide* also contains a step-by-step example of designing a project for financial reporting and analysis. This includes standard reporting such as profit and loss reporting that provides analysis of a company's profits compared to its losses.

You can also use the MicroStrategy Tutorial that contains predesigned report objects and reports as a simulated project to familiarize yourself with MicroStrategy. Many of the facts, attributes, and other objects used in the examples in this guide are available in the MicroStrategy Tutorial project.

You can now create reports with more sophisticated analyses, using the concepts described in this guide. You will learn how to:

- Define level metrics, conditional metrics, transformation metrics, and compound metrics, and know when to use each type
- Create advanced filters such as attribute-to-attribute qualifications, relationship filters, joint element lists, and prompted filters, among others
- Set up custom groups to create relationships between attributes and to band, or slice, attribute elements using the values of a metric
- Create virtual attributes and perform row level math using consolidations
- Create prompts to save time by using one report to produce different results
- Define custom drill maps to set the drill paths for reports

- Customize SQL statements
- Create and use MDX Cube Reports to integrate with SAP BI, Essbase, and Microsoft Analysis Services
- Create and use data mart reports to establish relational tables that can be used like tables in a project schema

Once you have understood and practiced these concepts, you will be able to choose, manipulate, and format advanced reports that best answer your business questions.

Importing data from different data sources

You can use MicroStrategy Web to import data from data sources, such as an Excel file, a table in a database, a Freeform SQL query, or a Salesforce.com report, into MicroStrategy metadata with minimum project design requirements.

You can import the data directly into a dashboard or a document, or import the data into the project. You can link the data to define attributes and metrics, and save it as an Intelligent Cube. An Intelligent Cube is a multi-dimensional cube (set of data) that allows you to use OLAP Services features on reports, as well as share sets of data among multiple reports. For information on Intelligent Cubes and the OLAP Services features that they support, see the *In-memory Analytics Guide*.

If you import data into a dashboard or a document, the imported data can be used immediately in the dashboard or the document. You can use the Intelligent Cube from the data import process, whether imported into a dashboard, document, or project, to create:

- A report that runs against the Intelligent Cube.
- A document that uses the Intelligent Cube as a dataset. A document is used to format data from multiple datasets in a single display of presentation quality. The dataset is a set of data that can be displayed on the document. In this case, the dataset is a MicroStrategy Intelligent Cube. For general information on working with documents, see the *Document and Dashboard Analysis Guide*; for general information on creating documents, see the *Report Services Document Creation Guide*.
- A dashboard using the imported data. A dashboard is a visually-striking, interactive display that takes a minimal amount of time to set up and use.

You can add text, interactive data visualizations, data filtering, and multiple layers of organization to your dashboard, then take advantage of Visual Insight's formatting options to customize your display. For background information on dashboards, see the *MicroStrategy Web Help*.

Some common uses for importing data include:

- Combining and analyzing personalized data with your project data
- Quickly integrating data into MicroStrategy as part of a proof-of-concept
- Importing and reporting on personalized data from various data sources
- Immediately building reports, documents, and dashboards without having to "model" the data source
- Modifying the data in your data source, then republishing the Intelligent Cube to quickly update the data in your reports, documents, and dashboards

You can import data from the following data sources:

- A BI tool, including SAP Business Objects Universe (BO), Oracle Business Intelligence Enterprise Edition (OBIEE), and IBM Cognos Framework Manager
- Custom data, by typing values or copying them from a file
- A database by picking relational tables, building a SQL query, or typing a custom query
- Facebook
- A file in a folder, URL, or file URI schema
- A file stored on Dropbox
- Google Analytics
- Google BigQuery
- Google Drive
- Hadoop
- A MicroStrategy sample file
- An OLAP source, including Microsoft Analysis Services, Cognos TM1, SAP BW, and Essbase

- A Salesforce report
- A search
- Twitter
- Importing data by scraping a web page (public data)

For steps to import data, as well as steps to link the imported data to project attributes, see the *MicroStrategy Web Help*.

You can add multiple datasets to a single dashboard or document, including datasets from the Import Data process and project datasets. For more information on including supplemental data in a project using the Data Import feature, see the *Creating and Configuring a Project* chapter in the *Project Design Guide*. For more information on creating dashboards with multiple datasets, see the *MicroStrategy Web Help*. For more information on creating documents with multiple datasets, see the *Report Services Document Creation Guide*.

You can manage Intelligent Cubes using the Intelligent Cube Monitor, which provides usage statistics and other information about Intelligent Cubes. For an introduction to the Intelligent Cube Monitor, as well as steps to improve performance, see the *Managing Intelligent Cubes* chapter of the *System Administration Guide*.

Analyzing system information

MicroStrategy supports importing data from search indexes, also known as search as a source. For example, you can examine data patterns by correlating events recorded in multiple sources, such as web access logs, error logs, and so on. Using search to explore your data provides a quick and intuitive way to access and analyze subsets of your data.

To support this type of system information analysis, you need to configure the following system:

- **Log management:** Your log files for your system must be retrieved and combined. Various tools, such as syslog-ng, are available to provide this type of log file management. Common log management tasks include formatting, classifying data, correlating related messages, logging directly into a database, and so on.
- **Log collection and search creation:** Once the data from your logs has been retrieved, this data must be stored and put through ETL processing to

create a standard log format. This data can then be turned into an index and search system. Tools such as Apache Flume™ and Apache Solr™ can be used together to provide a system for log collection and search creation.

- **Log analysis:** With your log files combined into a search system, you can integrate the search system into MicroStrategy. Along with performing search analysis on text data, you can also take advantage of all the standard MicroStrategy reporting, dashboarding, and analysis capabilities. For example, you have error logs for server stored in your search system, and you have performance data for that same server stored in a database. You can include these two collections of data on a single dashboard for analysis in MicroStrategy. For steps to integrate your search system into MicroStrategy using Data Import, refer to the *MicroStrategy Web Help*.

ADVANCED METRICS

Report Data and Calculations

Introduction

This chapter assumes you are familiar with the concepts in the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*. You should already know how to create a simple metric, place it in a report, and set report subtotals. This chapter covers the following advanced metric functionality:

- *Level metrics: Modifying the context of data calculations, page 14*
- *Nested metrics, page 60*
- *Conditional metrics, page 63*
- *Transformation metrics: Time-based and other comparisons, page 90*
- *Compound metrics, page 95*
- *Creating metrics by combining metrics, filters, and transformations, page 99*
- *Metric subtotals, page 102*
- *Metric functions, page 112*
- *Join specifications, page 121*
- *Metric-specific VLDB properties: SQL and analytical customization, page 123*

- [Metric column aliases: SQL identifiers and data types, page 123](#)



This chapter does not include Data Mining functions or predictive metrics. Refer to [Chapter 13, Data Mining Services](#) for information on these topics.

Before you begin

This section reviews basic concepts covered in the *Basic Reporting Guide*. If you need to brush up on metrics basics, this section is designed to help you. If you need a broader refresher on metrics, facts, attributes, prompts, and filters, see the *Basic Reporting Guide*.

Metrics are MicroStrategy objects that represent business measures and key performance indicators. They are calculations to be performed on data stored in the database and are similar to formulas in spreadsheet software. Questions such as “What were the sales for the eastern region during the fourth quarter?”, “Are inventory levels being consistently replenished at the beginning of each week?”, or “How many employees are currently working for our company?” can easily be answered by creating metrics.

A metric definition must contain a *formula*, which determines the data to be used and the calculations to be performed on the data. A metric is categorized as one of the following types based on the functions used in its formula:

- **Simple metric:** The formula of a simple metric is a mathematical expression based on at least one group function, such as sum or average, which is applied to facts, attributes, or other metrics. It can also contain non-group functions or arithmetic operators, in addition to the required group function.

An example of the formula of a simple metric is

```
Sum(Profit)
```

where `Cost` and `Profit` are facts. The formula contains two group functions, `Avg` (which stands for average) and `Sum`.

Another example is

```
Avg((Cost + Profit))
```

where `Cost` and `Profit` are metrics instead of facts. The formula contains a group function, so it is still a simple metric.



The term simple only refers to a metric's structure; it does not restrict you to simple calculations.

- **Compound metric:** The formula of a compound metric is based on arithmetic operators and non-group functions. Arithmetic operators are `+`, `-`, `*`, and `/`; non-group functions are OLAP and scalar functions such as running sum or rank. The operators and functions can be applied to facts, attributes, or metrics.

An example of the formula of a compound metric is

```
RunningAvg (Cost)
```

where `Cost` is a metric. The formula contains a non-group function, `RunningAvg` (which stands for running average). Another example is

```
Sum (Cost) + Sum (Profit)
```

where `Cost` and `Profit` are metrics. The addition expression, denoted by the `+` operator, makes this a compound metric.



For more background on the difference between simple and compound metrics, refer to the *Building Query Objects and Queries* chapter of the *Basic Reporting Guide*. For syntax and examples of all the functions that MicroStrategy supplies, see the *Functions Reference*.

In addition to a formula, a simple metric can contain the following components:

- The **level**, or dimensionality, determines the attribute level of calculation for the metric. For example, you can choose to calculate profit at the month level or the region level. By default, a metric is calculated at the report level, which is at the level of the attributes of the report in which the metric is placed. For a more detailed description, including the components of level metrics, examples, and procedures see [Level metrics: Modifying the context of data calculations, page 14](#).
- **Conditionality** associates a filter to the metric calculation. This is an optional component. For a more detailed description, including examples and procedures, see [Conditional metrics, page 63](#).
- A **transformation** applies offset values, such as “one month ago” or “last year”, to the selected attributes. This allows you to perform time-series analysis, such as a comparison of revenue between this year and last year. A transformation is an optional component. For a more detailed

description, including examples, see [Transformation metrics: Time-based and other comparisons, page 90](#).

None of these components can be set on the compound metric as a whole, although they can be applied separately on the metric or metrics that make up a compound metric. This explains why metrics are divided into the different types—so you know whether you can adjust the components of a metric. You can also create a metric that contains levels, a condition, and transformations, or any combination of these components. For an example, see [Example of a conditional metric with a level, page 65](#).

This chapter focuses on compound metrics and on simple metrics that use level, condition, and transformation components. For examples and steps to create simple metrics that do not use any of these components, see the *Building Query Objects and Queries, for Designers* chapter in the *Basic Reporting Guide*.

In this chapter, the Metric Editor in MicroStrategy Developer is used to work with metrics. You can also use MicroStrategy Command Manager to work with metrics. Command Manager manipulates MicroStrategy objects by means of user-created scripts rather than a graphical user interface. This can be helpful if you have a large number of metrics to create or edit. For background information about Command Manager and how to use it, see the *Automating Administrative Tasks* chapter in the *System Administration Guide*. For specific Command Manager syntax for working with metrics, see the online help for Command Manager.

Level metrics: Modifying the context of data calculations

By default, metrics are evaluated at the level of the attributes on the report; this is called the *report level*. For example, a revenue metric is set to the report level. When the metric is placed on a report containing Region, the metric calculates regional revenue. When it is placed on a report containing Customer, the revenue is calculated for each customer.

You can set the level within the metric, allowing you to specify the attribute(s) to use in the metric calculation, regardless of what is contained on any report the metric is placed upon. A metric level is sometimes also referred to as *dimensionality*, but the term level is used throughout this manual.

Level metrics are useful for determining the contribution of one object to the whole. For example, you need to determine the contribution to revenue and profit of products bought by your top customers. Level metrics allow you to create contribution metrics, as shown in the following report, which can be helpful for marketing and customer service.

Product	Metrics				
		Revenue	% of all Revenue (by Product)	Profit	% of all Profit (by Product)
Ladies Home Journal		\$5.99	0.01%	\$0.30	0.00%
Seventeen		\$5.99	0.01%	\$0.60	0.01%
Vegetarian Times		\$5.99	0.01%	\$0.60	0.01%
W Magazine		\$5.99	0.01%	\$0.60	0.01%
Airport Business		\$5.99	0.01%	\$0.60	0.01%
Business Alabama Monthly		\$5.99	0.01%	\$0.60	0.01%
Customers.Com		\$21.99	0.03%	\$1.10	0.02%
Permission Marketing		\$21.99	0.03%	\$1.10	0.02%
Advertising on the Internet 2nd Edition		\$21.99	0.03%	\$1.54	0.02%
Competing on the Edge		\$21.99	0.03%	\$1.10	0.02%

The Revenue and Profit metrics are familiar; they are simply the sum of the Revenue fact or the Profit fact. On the surface, the contribution percentage metrics should be just as easy—product revenue divided by all revenue, or product profit divided by all profit. But how can you calculate at two different levels in the same metric? The answer is a level metric, which allows you to specify how the metric is calculated.

For example, the % of All Revenue (by Product) metric used in the example above divides the Revenue metric by a metric called Revenue (Dimensionality All Products). The Revenue metric is calculated at the report level, in this case, for each product. A different number is calculated for each row of the report. In contrast, the level of the Revenue (Dimensionality All Products) metric has been set to product, which means that it reflects the

revenue of all products. The same number is calculated for each row of the report, as shown below.

Product	Metrics	Revenue	% of all Revenue (by Product)	Revenue (Dimensionality All Products)
Ladies Home Journal		\$5.99	0.01%	\$85,934.00
Seventeen		\$5.99	0.01%	\$85,934.00
Vegetarian Times		\$5.99	0.01%	\$85,934.00
W Magazine		\$5.99	0.01%	\$85,934.00
Airport Business		\$5.99	0.01%	\$85,934.00
Business Alabama Monthly		\$5.99	0.01%	\$85,934.00
Customers.Com		\$21.99	0.03%	\$85,934.00
Permission Marketing		\$21.99	0.03%	\$85,934.00
Advertising on the Internet 2nd Edition		\$21.99	0.03%	\$85,934.00
Competing on the Edge		\$21.99	0.03%	\$85,934.00

The level is indicated between the curly braces ({ }) in the metric definition shown below:

Sum(Revenue) {~, Product}

The tilde (~) indicates that the report level is still part of the metric level. For information about how the report level provides flexibility and the effects of removing it, see [Report level: Interaction with the context of reports, page 34](#).

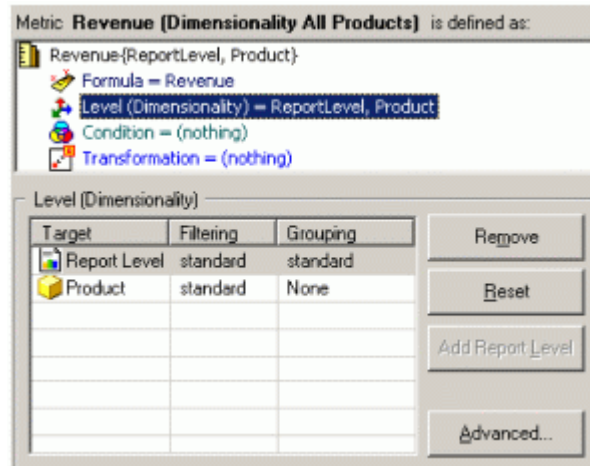
Elements of a metric level

The elements needed to specify a level for a metric are described below.

- **Target:** The target is the attribute level at which the metric calculation groups. For a more detailed description, see [Target: The context of a calculation, page 22](#).
- **Grouping:** Grouping determines how the metric aggregates. For a more detailed description, see [Grouping: How aggregation is performed, page 23](#).
- **Filtering:** Filtering governs how the report filter interacts with the metric calculation. For a more detailed description, see [Filtering: Interaction with report filters, page 30](#).

A target, grouping, and filtering combination composes one level unit.

These elements are set in the Metric Editor, using the metric definition window, which displays the complete metric definition, including its formula, level, condition, and transformation. When you select Level (Dimensionality) in the upper portion of this window, the Level (Dimensionality) component window appears below the metric definition area, as shown below.



Clicking **Reset** changes the level unit back to the default of report level for the target and standard for filtering and grouping. The **Add Report Level** button becomes available if the report level has been removed from the metric level.

 The **Advanced** button accesses the Level (Dimensionality) advanced options dialog box, which is discussed in [Level metrics: Applying the metric condition to the metric calculation, page 53](#) and [Level metrics accepting dimensionality units to emulate MicroStrategy 6.x behavior, page 59](#).

Metric level symbols: Defining expression syntax

The metric definition (displayed in the Metric Editor) indicates the target of the metric level. All information about the metric level is placed between curly braces ({ }).

You can define the grouping and filtering settings by displaying additional level settings. To show the additional settings, select **Show level (dimensionality) properties** on the **View** menu. This option is available when the metric definition is selected.

The following table describes the metric level symbols.

Symbol	Definition
~	Report level
An attribute name	Target
Symbols placed before the target	
No symbol	Grouping = Standard
!	Grouping = None
<	Grouping = Beginning fact
>	Grouping = Ending fact
<	Grouping = Beginning lookup
>	Grouping = Ending lookup
Symbols placed after the target	
+	Filtering = Standard
*	Filtering = Absolute
%	Filtering = Ignore
No symbol	Filtering = None

Overview of level metrics

This section helps you choose the right combination of level target, filtering, and grouping (referred to as elements) to achieve your desired results. This section shows the effects of different element settings on a single metric when used in the same report. The effects are summarized in [Level metrics summary table, page 19](#).

The elements of a metric level are described below:

- **Target:** The target is the attribute level at which the metric calculation groups. For a more detailed description, see [Target: The context of a calculation, page 22](#).
- **Grouping:** Grouping determines how the metric aggregates. For a more detailed description, see [Grouping: How aggregation is performed, page 23](#).

- **Filtering:** Filtering governs how the report filter interacts with the metric calculation. For a more detailed description, see [Filtering: Interaction with report filters, page 30](#).

The base metric and report

All the metrics in this section are based on a revenue metric that is the sum of the Revenue fact. The base report displayed below contains this Revenue metric, and the Category and Subcategory attributes. Each category is subtotaled, and a grand total is calculated. It has a report filter on the following subcategories:

- Art & Architecture
- Literature
- Drama
- Alternative

The base report is shown below:

Category	Subcategory	Metrics	Revenue
Books	Art & Architecture		\$480,173
	Literature		\$296,229
	Total		\$776,402
Movies	Drama		\$698,840
	Total		\$698,840
Music	Alternative		\$706,130
	Total		\$706,130
Total			\$2,181,372

Level metrics summary table

The Revenue metric calculates the revenue for each subcategory displayed on the report.

To Calculate	Set Filtering To	Set Grouping To
Revenue at the category level, including only the subcategories displayed on the report	Standard	Standard
Revenue at the category level, for all subcategories in the categories displayed on the report	Absolute	Standard
Total revenue for the subcategories displayed on the report	Standard	None

To Calculate	Set Filtering To	Set Grouping To
Total revenue for all subcategories in the categories displayed on the report	Absolute	None
Total revenue for all subcategories in the project	Ignore	None

 All the level metrics described in this section have a target of Category.

Category revenue examples

The following report shows the base report with the addition of two new metrics, both measuring category revenue in different ways.

		Metrics		Category Revenue for Displayed Subcategories	Category Revenue for All Subcategories in Displayed Categories
Category	Subcategory		Revenue		
Books	Art & Architecture		\$480,173	\$776,402	\$2,640,094
	Literature		\$296,229	\$776,402	\$2,640,094
	Total		\$776,402	\$776,402	\$2,640,094
Movies	Drama		\$698,840	\$698,840	\$4,098,943
	Total		\$698,840	\$698,840	\$4,098,943
Music	Alternative		\$706,130	\$706,130	\$3,893,367
	Total		\$706,130	\$706,130	\$3,893,367
Total			\$2,181,372	\$2,181,372	\$10,632,405

Notice that the Category Revenue for Displayed Subcategories metric returns the same number as the category subtotal. The Art & Architecture and Literature columns are the same as the Books Total. Why?

- **Standard filtering** allows the report filter to interact as usual in the metric calculation. Therefore, only the subcategories in the report filter are included in the metric calculation. This is affirmed by the grand total of this metric—it matches the total for the Revenue metric. This indicates that only the attributes displayed on the report are included in this Category Revenue metric.

The numbers returned by the Category Revenue for All Subcategories in Displayed Categories metric are higher than the numbers for the other metrics on the report. Why?

- **Absolute filtering** changes the filter on children of the target, by raising it to the level of the target, if possible. In this example, the report filter is

Subcategory, which is a child of Category, the level target. Since the report filter is on a lower level than the target, the filter is raised to the level of the target. All subcategories in the categories on the report are included in the metric calculation.

Why do the Category Revenue metrics calculate the same number for each row in a particular category?

- Both metrics have **standard grouping**, which means that the metric groups by the attribute level of the target. In this case, the target is category. The metric calculation is rolled up to Category, so the same number is repeated for each row in a particular category.

Total revenue examples

The following report shows the base report with the addition of three new metrics, all measuring total revenue.

Category	Subcategory	Metrics	Revenue	Total Revenue for Displayed Subcategories	Total Revenue for All Subcategories in Displayed Categories	Total Revenue for All Subcategories
Books	Art & Architecture		\$480,173	\$2,181,372	\$10,632,405	\$35,023,708
	Literature		\$296,229	\$2,181,372	\$10,632,405	\$35,023,708
	Total		\$776,402	\$2,181,372	\$10,632,405	\$35,023,708
Movies	Drama		\$698,840	\$2,181,372	\$10,632,405	\$35,023,708
	Total		\$698,840	\$2,181,372	\$10,632,405	\$35,023,708
Music	Alternative		\$706,130	\$2,181,372	\$10,632,405	\$35,023,708
	Total		\$706,130	\$2,181,372	\$10,632,405	\$35,023,708
Total			\$2,181,372	\$2,181,372	\$10,632,405	\$35,023,708

The most obvious difference between this report and the Category Revenue example above is that each column contains only one number; each metric is returning only one result, regardless of row. Why?

- All the metrics on this report, except for Revenue, have grouping set to none. **No grouping** means that the metric does not group on the target of category or the target's child attributes, such as subcategory. Therefore, separate results are not calculated for the different attributes on the report; only one result is calculated.

The Total Revenue for Displayed Subcategories metric returns the same number as the total for the Revenue metric. Why?

- **Standard filtering** allows the report filter to affect the metric. Therefore, only the subcategories in the report filter are included in the metric calculation. This is confirmed by the number matching the total for the Revenue metric. This indicates that only the attributes displayed on the report are included in this Total Revenue metric.

Refer to the report in [Category revenue examples, page 20](#). Notice that the total for the Category Revenue for All Subcategories in Displayed Categories metric is the same amount calculated for the Total Revenue for All Subcategories in Displayed Categories metric on the total revenue report. Why?

- Both metrics have filtering set to absolute. **Absolute filtering** raises the report filter to the level of the target, so all subcategories in the categories included on the report are added together.

The result for the Total Revenue for All Subcategories metric is huge. Why?

- It includes the total revenue for all subcategories in the entire project. **Ignore filtering** disregards filtering criteria based on the attribute in the target and its related attributes (both parents and children). In this case, subcategory in the report filter is ignored, so the report filter does not apply to this metric.

Target: The context of a calculation

The **target** is the attribute level at which the metric calculation groups. It determines the table to use to calculate the metric. Any set of attributes or a hierarchy can be the target. A special case is the default target, which is at the report level. For a more detailed description of the report level, including an example, see [Report level: Interaction with the context of reports, page 34](#). For specifics on using a hierarchy as a target, see [Using a hierarchy as the target of a metric level, page 47](#).



The following examples use attributes as the target; for an example using a hierarchy, see [Level metrics review: Standard grouping, page 36](#).

In the Revenue (Dimensionality All Products) metric discussed previously (see [Level metrics: Modifying the context of data calculations, page 14](#)), the target is Product. The target is indicated between braces in the metric definition, as shown below:

```
Sum(Revenue) {~, Product}
```

Grouping: How aggregation is performed

Grouping determines how the metric aggregates. The result of this setting is reflected in the GROUP BY clause of the SQL command. The grouping options for levels include:

- **Standard** groups by the attribute level of the target. The metric calculates at the level of the target, if possible.
- **None** excludes the attribute in the target from the GROUP BY clause. It also excludes any of the target attribute's children.



None is not an option if the target is set to the report level.

Grouping options for nonaggregatable metrics

The remaining grouping options are only used for nonaggregatable metrics. A nonaggregatable metric is one that should not be aggregated across an attribute. An example is an inventory metric. While the data warehouse records the inventory every month, these monthly numbers are not added together to calculate the yearly inventory. Instead, you may want to use the end-on-hand and beginning-on-hand inventory numbers to see how the total inventory changed over the year. These grouping options, described below, are used in such cases:

- **Beginning lookup** uses the first value of the lookup table.
- **Ending lookup** uses the last value of the lookup table.
- **Beginning fact** accesses the first value of the fact table.
- **Ending fact** accesses the last value contained in the fact table.

Setting a metric level to one of the options listed above defines the metric as nonaggregatable. Whether you select a fact or lookup table largely depends on how the necessary information is stored. For example, to find the beginning-on-hand inventory for a particular item, you need to know how the inventory information is stored. If the inventory count is not taken on the first day of the week, as the lookup table requires, the inventory count should be taken from the fact table for the first recorded entry.



There is another important difference between accessing a fact table and a lookup table. If a value, such as April sales, is missing from a fact table, the row still exists in the table and is reported as null or zero. If that same value is missing in a lookup table, the April row does

not exist. The previous or next value (March or May) is reported, depending on whether the level is set to beginning or ending value.

Grouping in the Products - By Top 10% Customers report

Review the Products - By Top 10% Customers report described at the beginning of [Level metrics: Modifying the context of data calculations, page 14](#). In the report, the % of All Revenue (by Product) metric divides the Revenue metric by a metric called Revenue (Dimensionality All Products). The Revenue metric is calculated at the report level, in this case, for each product. The level of the Revenue (Dimensionality All Products) metric has been set to product, with a grouping of none. This allows the calculation to total the revenue for all products, not just the single product on the particular row of the report. Therefore, the same number is calculated for each row of the report, producing the revenue percentage.

The grouping is indicated in the metric definition. If the grouping is standard, an additional indicator is not displayed. If the grouping is set to none, as in the Revenue (Dimensionality All Products) metric, an exclamation mark is placed before the target name, as shown below:

```
Revenue {~+, !Product+}
```



If you open this metric and do not see the exclamation mark, select **Show level (dimensionality) properties** from the **View** menu.

Level grouping examples

A revenue metric is defined as:

```
Sum(Revenue) {~, Subcategory}
```

The level target is set to Subcategory, with standard grouping. When this metric is placed on a report with the Subcategory attribute, the report results are shown below.

Subcategory	Metrics	Group By Subcategory
Art & Architecture		\$480,173
Business		\$400,871
Literature		\$296,229
Books - Miscellaneous		\$315,929
Science & Technology		\$811,787
Sports & Health		\$335,106
Audio Equipment		\$3,782,832
Cameras		\$5,061,148
Computers		\$1,928,998



This is only a subset of the report.

Notice that the sales are calculated for each subcategory, because the metric is grouping at the subcategory level, as shown in the SQL:

```
select a11.[SUBCAT_ID] AS SUBCAT_ID,
       max(a12.[SUBCAT_DESC]) AS SUBCAT_DESC,
       sum(a11.[TOT_DOLLAR_SALES]) as REVENUE
from [CITY_SUBCATEG__SLS] a11,
     [LU_SUBCATEG] a12
where a11.[SUBCAT_ID] = a12.[SUBCAT_ID]
group by a11.[SUBCAT_ID]
```

Using the same metric on a report with the Item attribute, however, yields the following results.

Item	Metrics	Group By Subcategory
100 Places to Go While Still Young at Heart		\$480,173
Art As Experience		\$480,173
The Painted Word		\$480,173
Hirschfeld on Line		\$480,173
Adirondack Style		\$480,173
Architecture : Form, Space, & Order		\$480,173
50 Favorite Rooms		\$480,173
500 Best Vacation Home Plans		\$480,173
Working With Emotional Intelligence		\$400,871
Attention to Detail		\$400,871
The 48 Laws of Power		\$400,871
Don't Step in the Leadership		\$400,871
The Prince		\$296,229
The Fountainhead		\$296,229
The Old Man and the Sea		\$296,229
Lord of the Flies		\$296,229
Atlas Shrugged		\$296,229
The Catcher in the Rye		\$296,229
Brave New World		\$296,229



The sample above is representative of the report, but rows were removed to make it easier to see the changing values.

Although each item is listed, the value for each item in a given subcategory is the same. The metric is calculating revenue by subcategory, based on the grouping level defined in the metric. The SQL for this report is, in essence, the same as the previous example:

```
insert into TEMP_TABLE

select a11.[SUBCAT_ID] AS SUBCAT_ID,
      sum(a11.[TOT_DOLLAR_SALES]) as REVENUE
from [YR_CATEGORY_SLS] a11
group by a11.[CATEGORY_ID]

select a11.[SUBCAT_ID] AS SUBCAT_ID,
      a11.[SUBCAT_DESC] AS SUBCAT_DESC,
      pa1.[REVENUE] as REVENUE
from [TEMP_TABLE] pa1,
```



```
[LU_SUBCAT] a11
where pa1.[CATEGORY_ID] = a11.[CATEGORY_ID]
```

Change the grouping to none on that same revenue metric and place it on a report with Category. Because Category is a parent of Subcategory, the metric can roll up to the Subcategory level. The report and its SQL are illustrated below.

	Metrics	Don't Group By Subcategory
Category		
Books		\$2,640,094
Electronics		\$24,391,303
Movies		\$4,098,943
Music		\$3,893,367

```
select a12.[CATEGORY_ID] AS CATEGORY_ID,
sum(a11.[TOT_DOLLAR_SALES]) as REVENUE
from [CITY_SUBCATEG_SLS] a11,
[LU_SUBCATEG] a12,
[LU_CATEGORY] a13
where a11.[SUBCAT_ID] = a12.[SUBCAT_ID] and
a12.[CATEGORY_ID] = a13.[CATEGORY_ID]
group by a12.[CATEGORY_ID]
```

A more efficient method is including a total category sales fact table in the project. Instead of adding all the subcategories together, the category total could have been pulled directly from that table. However, having Subcategory in the level of the metric forces the report to use the subcategory sales table.

If the same revenue metric, with the grouping set to none, is used on a report with Item, the report results are shown below.

	Metrics	Don't Group By Subcategory
Item		
100 Places to Go While Still Young at Heart		\$35,023,708
Art As Experience		\$35,023,708
The Painted Word		\$35,023,708
Hirschfeld on Line		\$35,023,708
Adirondack Style		\$35,023,708
Architecture : Form, Space, & Order		\$35,023,708
50 Favorite Rooms		\$35,023,708
500 Best Vacation Home Plans		\$35,023,708
Blue & White Living		\$35,023,708
Ways of Seeing		\$35,023,708

The metric calculates the same number for each item—the total for all the items included on the report. Because Item is a child of Subcategory, Item is excluded from the group by clause:

```
insert into TEMP_TABLE

select sum(a11.[TOT_DOLLAR_SALES]) as REVENUE

from [CITY_SUBCATEG_SLS] a11


select a12.[ITEM_ID] AS ITEM_ID,
       a12.[ITEM_DESC] AS ITEM_NAME,
       pa11.[REVENUE] as REVENUE
from [TEMP_TABLE] pa11,
     [LU_ITEM] a12


drop table TEMP_TABLE
```



Additional examples can be found in [Level metrics review: Standard grouping, page 36](#) and [Level metrics review: No grouping, page 42](#).

Level grouping examples with nonaggregatable metrics

Inventory is one example of a nonaggregatable metric. The following metric definition reports the inventory on hand at the end of the month. The level is set at the report level and at month, with a grouping of ending fact, so that the last entry in the fact table is used.

```
Sum([End on hand]) {~, Month}
```



If level (dimensionality) options are shown (click the **Show level (dimensionality) properties** icon on the toolbar), the metric definition looks like `Sum([End on hand]) {~+, >|Month+}`. If grouping was set to beginning fact, the indicator would be `<|`. Similarly, the indicators for beginning lookup and ending lookup are `<` and `>`, respectively. See [Metric level symbols: Defining expression syntax, page 17](#).

A report contains this metric and the month attribute. The last entry for each month in the fact table is placed on the report. No calculation is performed.

Month	Metrics	Month-End Inventory
Jan 02		25,145
Feb 02		50,055
Mar 02		33,740
Apr 02		22,830
May 02		32,139
Jun 02		30,320
Jan 03		43,233
Feb 03		39,779
Mar 03		38,626



This is a sample of different portions of the report, not the entire report.

When the same metric is used on a report with quarter, the value for each quarter is the value for the last month in the quarter. The monthly values for each quarter are not added together. For example, the on-hand inventory for

March 2002 is 33,740. Since that is the last month in Q1, that value is reported on the quarterly report.

Quarter	Metrics	Inventory
Q1 02		33,740
Q2 02		30,320
Q3 02		38,499
Q4 02		28,659
Q1 03		38,626
Q2 03		32,295
Q3 03		43,697
Q4 03		28,576

Filtering: Interaction with report filters



You can also place a filter, or condition, on a level metric. See [Example of a conditional metric with a level, page 65](#) for an example and detailed description.

Filtering in the Products - By Top 10% Customers report

Review the Products - By Top 10% Customers report described at the beginning of [Level metrics: Modifying the context of data calculations, page 14](#). In the report, the % of All Revenue (by Product) metric divides the Revenue metric by a metric called Revenue (Dimensionality All Products). The Revenue metric is calculated at the report level, in this case, for each product. The level of the Revenue (Dimensionality All Products) metric has been set to product, with filtering set to standard. The report filter contains a prompt for quarter and a filter for the top ten percent of customers based on revenue. Standard filtering means that revenue is calculated for only the quarter selected in the prompt.

The filtering is indicated in the metric definition. Standard filtering is indicated by a plus sign after the target, as shown in the definition of the Revenue (Dimensionality All Products) metric shown below:

Revenue {~+, !Product+}



If you open this metric and do not see the plus sign, select **Show level (dimensionality) properties** from the **View** menu.

Level filtering examples



All of the metrics in these examples, except the first report, have grouping set to none.

Consider the following report as a baseline to show the revenue for each Category and Subcategory.

Category	Subcategory	Metrics	Revenue
Books	Art & Architecture		\$480,173
	Business		\$400,871
	Literature		\$296,229
	Books - Miscellaneous		\$315,929
	Science & Technology		\$811,787
	Sports & Health		\$335,106
	Total		\$2,640,094
Electronics	Audio Equipment		\$3,782,832
	Cameras		\$5,061,148
	Computers		\$1,928,998
	Electronics - Miscellaneous		\$4,671,957
	TV's		\$3,837,906
	Video Equipment		\$5,108,464
	Total		\$24,391,303
Music	Alternative		\$706,130
	Country		\$729,299
	Music - Miscellaneous		\$597,508
	Pop		\$687,861
	Rock		\$700,756
	Soul / R&B		\$471,814
	Total		\$3,893,367
Total			\$35,023,708



None of the reports in these examples are presented in full; they are only subsets of the complete report.

A revenue metric is defined with Category as the target, no grouping, and standard filtering. A report is created with Category, Subcategory, this new revenue metric, and a filter for the Literature subcategory. When the report is executed, the revenue is the same for every row, as shown below. All categories are included on the report, even though the report filter is Literature. This is an effect of setting the grouping to none. Since Category in the target is a parent of Subcategory in the filter, all subcategories are

included on the report. The metric value is the grand total of the filter, in this case, only Literature.

		Metrics	Filter By Category
Category	Subcategory		
Books	Art & Architecture		\$296,229
	Business		\$296,229
	Literature		\$296,229
	Books - Miscellaneous		\$296,229
	Science & Technology		\$296,229
	Sports & Health		\$296,229
	Total		\$296,229
Electronics	Audio Equipment		\$296,229
	Cameras		\$296,229
	Computers		\$296,229
	Electronics - Miscellaneous		\$296,229
Music	Pop		\$296,229
	Rock		\$296,229
	Soul / R&B		\$296,229
	Total		\$296,229
Total			\$296,229

The same report is created with a metric set to absolute filtering and no grouping. When the report is executed, the revenue is the same for every row, as shown below. Because of the absolute setting, the report filter rolls up to the level of the metric, and thus Subcategory is elevated to Category. Because

the report is filtered for the Literature subcategory, the value is revenue for the Books category.

		Metrics	Absolute By Category
Category	Subcategory		
Books	Art & Architecture		\$2,640,094
	Business		\$2,640,094
	Literature		\$2,640,094
	Books - Miscellaneous		\$2,640,094
	Science & Technology		\$2,640,094
	Sports & Health		\$2,640,094
	Total		\$2,640,094
Electronics	Audio Equipment		\$2,640,094
	Cameras		\$2,640,094
	Computers		\$2,640,094
	Electronics - Miscellaneous		\$2,640,094
Music	Pop		\$2,640,094
	Rock		\$2,640,094
	Soul / R&B		\$2,640,094
	Total		\$2,640,094
Total			\$2,640,094



If level (dimensionality) options are shown, the metric definition looks like `Sum(Revenue) {~+, !Quarter*}`. The asterisk denotes absolute filtering, the exclamation mark that grouping is set to none. See [Metric level symbols: Defining expression syntax, page 17](#).

The same report is run, but this time with a metric that has level filtering set to ignore. Grouping remains set to none. Again the metric value is the same

for the whole report, but now it is the grand total of all sales in the project. Since Category is related to Subcategory, the filter is also ignored.

		Metrics	Ignore By Category
Category	Subcategory		
Books	Art & Architecture		\$35,023,708
	Business		\$35,023,708
	Literature		\$35,023,708
	Books - Miscellaneous		\$35,023,708
	Science & Technology		\$35,023,708
	Sports & Health		\$35,023,708
	Total		\$35,023,708
Electronics	Audio Equipment		\$35,023,708
	Cameras		\$35,023,708
	Computers		\$35,023,708
	Electronics - Miscellaneous		\$35,023,708
Music	Pop		\$35,023,708
	Rock		\$35,023,708
	Soul / R&B		\$35,023,708
	Total		\$35,023,708
Total			\$35,023,708



If level (dimensionality) options are shown, the metric definition looks like `Sum (Revenue) {~+, !Quarter%}`. The percent sign denotes ignore filtering.

Report level: Interaction with the context of reports

When you create a metric, the level is set by default as the report level. The metric calculates at the level of the attributes on the report on which the metric is placed. The level of a metric is indicated between curly braces ({}) in the metric definition, as shown in the following simple Revenue metric:

```
Sum (Revenue) {~+}
```

The tilde (~) represents the report level with standard filtering, denoted by the plus sign (+). If you add item as a level, the metric definition changes to reflect the addition:

```
Sum (Revenue) {~+, Item+}
```

Notice that the report level has not been deleted. Keeping the report level allows the metric calculation to adapt to the report. If this revenue metric is placed on a report containing region, that attribute affects the metric calculation, along with the level explicitly set on the metric (item). If you put

the same metric on a report with customer, that attribute, as well as the metric level, is used in the metric calculation. In other words, the report level in the metric level tells the engine to group by all the attribute IDs on the report. The content of the report is reflected in the metric calculation.

Removing the report level

If the report level is removed, only the level explicitly set on the metric affects the metric calculation, regardless of what the report contains. For example, remove the report level from the revenue metric described previously. The metric definition now looks like the following:

```
Sum (Revenue) { Item }
```

Place this metric on a report with the Region attribute and two other revenue metrics, one with report level only and the other with report level and item. The report results are shown below.

	Metrics	Revenue: Report Level	Revenue: Report & Item	Revenue: Item
Region				
Central		\$5,029,366	\$5,029,366	\$35,023,708
Mid-Atlantic		\$4,452,615	\$4,452,615	\$35,023,708
Northeast		\$8,554,415	\$8,554,415	\$35,023,708
Northwest		\$1,761,187	\$1,761,187	\$35,023,708
South		\$5,389,280	\$5,389,280	\$35,023,708
Southeast		\$2,239,951	\$2,239,951	\$35,023,708
Southwest		\$3,694,132	\$3,694,132	\$35,023,708
Web		\$3,902,762	\$3,902,762	\$35,023,708
Total		\$35,023,708	\$35,023,708	\$35,023,708

The Revenue: Report Level and Revenue: Report & Item metrics calculate the same result, providing the revenue for each region. The Revenue: Item metric calculates the revenue for all items, for all regions, in each row. The number is therefore the same for each row, since the metric does not differentiate between regions. The number is the same as the grand total of the report.

As shown above, removing the report level allows you to easily remove grouping and calculate the metric across all hierarchies. The Revenue: Item metric can be used to compare the sales performance of targeted regions to the total sales in the company everywhere and for all time. You could include multiple attribute levels in a revenue metric, to create a “total sales” metric. A quicker and easier solution is to remove the report level target that is present

by default and add any attribute as the target with no grouping, as in the Revenue: Item metric above.

By removing the report level from the target and with no grouping for any other available attribute, the SQL does not generate a GROUP BY clause. Any attribute can be used for this purpose. You do not need to add more than one attribute, unless a special filtering behavior is required for the metric. If a special filtering behavior is required, then other attributes are required but they should not be grouped.

This is a quick and easy way to do something that might otherwise involve multiple steps. It is especially helpful if you have many dimensions represented on a report that need to be included in the metric calculation to obtain the desired outcome.

Level metrics review: Standard grouping

This section describes how level metrics can help you answer your report requirements. In particular, it discusses how standard grouping interacts with different filtering options. Report samples, SQL code, and metric definitions are used to explain the results.

Report requirements

Your company has recently kicked off a new ad campaign targeted at certain areas that present high growth opportunities. In your regions, this consists of the Boston, New York, and Washington, DC call centers. You need to perform an analysis from different perspectives and are looking for answers to the following:

- 1 How do the sales of each call center compare to the total sales of the targeted call centers in a given region?
- 2 How do the sales of each call center compare to the total sales of all the call centers in a given region?
- 3 How do the sales of each call center compare to the total sales of all the call centers in a given region for a given category?

Report 1: Call center sales compared to total sales of targeted call centers in a given region

Level metric: Grouping = Standard, Filtering = Standard

In this case, the Regional Sales is equal to the sum of the revenues of the call centers in a given region. This sum takes into account only those call centers that are included in the report filter. For example, the Mid-Atlantic Regional Sales only includes the Washington, DC call center sales as this is the only call center from that region that has been included in the report filter. The metric groups at the target level of region because grouping is standard, as shown in the metric definition below:

```
Sum(Revenue) {~+, Region+}
```

With standard filtering, all of the report filter elements are included in the calculation of the metric. This occurs by placing the report filter in the WHERE clause of the SQL pass for this metric, as shown in the following example:

```
sum(a11.[ORDER_AMT]) as REGIONALSALES
  from [ORDER_FACT] a11,[LU_EMPLOYEE]a12,
      [LU_CALL_CTR] a13
 where a11.[EMP_ID] = a12.[EMP_ID]
 and a12.[CALL_CTR_ID] = a13.[CALL_CTR_ID]
 and a12.[CALL_CTR_ID] in (5, 11, 12)
 group by a13.[REGION_ID]
```

The report is displayed as follows:

Report details				
Report Filter: {Call Center} = Washington, DC, Boston, New York				
Region	Call Center	Metrics	Revenue	Regional Sales {Target=Region, Filtering=Standard, Grouping=Standard}
Mid-Atlantic	Washington, DC		\$3,135,283	\$3,135,283
	Total		\$3,135,283	\$3,135,283
Northeast	Boston		\$1,487,936	\$8,554,415
	New York		\$7,066,478	\$8,554,415
	Total		\$8,554,415	\$8,554,415
Total			\$11,689,697	\$11,689,697



The Revenue subtotals match up with the values of the total Regional Sales.

Report 2: Call center sales compared to total sales of all call centers in a given region

Level metric: Grouping = Standard, Filtering = Absolute

In this case, the Regional Sales is equal to the sum of revenues of all call centers included in a given region. Grouping continues to occur at the target attribute level of region. The metric definition is shown below:

```
Sum(Revenue) {~+, Region*}
```

With absolute filtering, the report filter is present in the subquery of the WHERE clause in the SQL pass for this metric as shown in the following example:

```
select a13.[REGION_ID]) as REGION_ID,
       sum(a11.[ORDER_AMT]) as REGIONALSALES
from [ORDER_FACT] a11, [LU_EMPLOYEE] a12,
     [LU_CALL_CTR] a13
where a11.[EMP_ID] = a12.[EMP_ID]
and a12.[CALL_CTR_ID] = a13.[CALL_CTR_ID]
and ((a13.[REGION_ID])
```

```

in (select s21.[REGION_ID]
    from [LU_CALL_CTR] s21
    where s21.[CALL_CTR_ID] in (5,11,12))
group by a13.[REGION_ID]

```

The report is shown in the following figure:

Report details				
Report Filter: {Call Center} = Washington, DC, Boston, New York				
Region	Call Center	Metrics	Revenue	Regional Sales {Target=Region, Filtering=Absolute, Grouping=Standard}
Mid-Atlantic	Washington, DC		\$3,135,283	\$4,452,615
	Total		\$3,135,283	\$4,452,615
Northeast	Boston		\$1,487,936	\$8,554,415
	New York		\$7,066,478	\$8,554,415
	Total		\$8,554,415	\$8,554,415
Total			\$11,689,697	\$13,007,030



Note the following:

- The Regional Sales values for the Mid-Atlantic region do not match the Revenue values because the Regional Sales metric includes all the call centers in the region, regardless of whether they are included in the report. The report filter is applied to the Revenue metric, so only Washington, DC is included in the subtotal. Likewise, the grand totals for the two metrics do not match.
- With absolute filtering, the report filter is placed in the subquery of the WHERE clause only if it is of a lower level than the target. If the report filter is of a higher level than the target, there is no need for a subquery and so the engine does not use one.
- The VLDB properties of the report can be changed to use two passes of SQL rather than a subquery. VLDB properties related to metrics are discussed in [Metric-specific VLDB properties: SQL and analytical customization, page 123](#).

Report 3: Call center sales compared to total sales of all call centers in a given region for a given category

Level metric: Grouping = Standard, Filtering = Ignore

In this case, the engine ignores the report filter and the report displays the Regional Sales as the sum of revenues of all the call centers in that region. The metric definition is shown below:

```
Sum(Revenue) {~+, Region%}
```

With no filtering, the report filter elements that are directly related to the target attributes are not placed in the WHERE clause of the SQL pass for the metric as shown in the following example:

```
select a13.[REGION_ID]) as REGION_ID,  
       sum(a11.[ORDER_AMT]) as REGIONALSALES  
from [ORDER_FACT] a11, [LU_EMPLOYEE] a12,  
     [LU_CALL_CTR] a13  
where a11.[EMP_ID] = a12.[EMP_ID]  
and a12.[CALL_CTR_ID] = a13.[CALL_CTR_ID]  
group by a13.[REGION_ID]
```



If the report filter contains attribute elements such as category, these attributes are not ignored because they are not directly related to the target attribute region.

In the following example, since call centers are directly related to the target attribute region, the entire report filter is ignored for the Regional Sales metric. Regional Sales calculates values for all the call centers in each region, regardless of whether the call centers are shown on the report. This explains

why Regional Sales values for the Mid-Atlantic region are higher than the Revenue values.

Report details				
Report Filter: {Call Center} = 'Washington, DC, Boston, New York'				
Region	Call Center	Metrics	Revenue	Regional Sales {Target=Region, Filtering=Ignore, Grouping=Standard}
Mid-Atlantic	Washington, DC		\$3,135,283	\$4,452,615
	Total		\$3,135,283	\$4,452,615
Northeast	Boston		\$1,487,936	\$8,554,415
	New York		\$7,066,478	\$8,554,415
	Total		\$8,554,415	\$8,554,415
Total			\$11,689,697	\$13,007,030

In the example that follows, the Electronics category is included in the report filter. For Regional Sales, the conditions in the report filter that are related to the target of Region (in this case, Call Center) are ignored. Since Category is not related, the Regional Sales metric is calculated for Electronics only. Revenue is also calculated for Electronics only.

Report details				
Report Filter: {Call Center} = 'Washington, DC, Boston, New York' And (Category = Electronics)				
Region	Call Center	Metrics	Revenue	Regional Sales {Target=Region, Filtering=Ignore, Grouping=Standard}
Mid-Atlantic	Washington, DC		\$2,193,709	\$3,106,940
	Total		\$2,193,709	\$3,106,940
Northeast	Boston		\$1,034,067	\$5,962,709
	New York		\$4,928,642	\$5,962,709
	Total		\$5,962,709	\$5,962,709
Total			\$8,156,418	\$9,069,649



Security filters are included in the WHERE clause of the level metric's SQL statement even with absolute or ignore filtering. The engine includes the security filter to ensure that there is no breach in security for any level metric. With filtering ignored, the security filter is unioned with the report filter and is applied to the metric also. With absolute filtering, the security filter is applied in the subquery with the report filter.

Level metrics review: No grouping

This section describes how level metrics can help you answer your report requirements. In particular, it discusses the interaction between grouping set to none and different filtering options. Report samples, SQL code, and metric definitions are used to explain the results.

Report requirements

Your company has recently kicked off a new ad campaign targeted at certain areas that present high growth opportunities. In your regions, this consists of the Boston, New York, and the Washington, DC call centers. You need to perform an analysis from different perspectives and are looking for answers to the following:

- 1 How did the sales of these three call centers compare to the total of all three?
- 2 How did the sales of these three call centers compare to the total sales of all call centers within the targeted regions?
- 3 How did the sales of each of the three call centers compare to the sales of the entire company?
- 4 What were the sales in each region, based on the items sold in each call center in that region?

The answers to these questions give you an insight into how the new campaign is being received in the targeted areas of your region.

Report 1: Call center sales compared to total sales of targeted call centers

Level metric: Grouping = None, Filtering = Standard

In this business scenario, the Regional Sales metric calculates the total sales for all call centers present in the report filter. By changing grouping to none, the metric does not group by anything directly related to the target attribute specified within the metric. The metric definition is shown below:

```
Sum (Revenue) {~+, !Region+}
```


Therefore, in this example, there is no GROUP BY statement in the SQL as the attributes call center and region are directly related to the metric target region. With standard filtering, the report filter elements are included in the WHERE clause of the SQL as shown in the following example.

```
select sum(a11.[ORDER_AMT]) as REGIONALSALES
from [ORDER_FACT] a11, [LU_EMPLOYEE] a12
where a11.[EMP_ID] = a12.[EMP_ID]
and a12.[CALL_CTR_ID] in (5,11,12)
```

The report is displayed in the following figure:

Report details			
Report Filter (Local Filter): {Call Center} = 'Washington, DC,Boston or New York'			
Region	Call Center	Revenue	Regional Sales { Target=Region, Filtering=Standard, Grouping=None }
Mid-Atlantic	Washington, DC	\$16,276,620	\$27,652,033
	Total	\$16,276,620	\$27,652,033
Northeast	Boston	\$2,027,109	\$27,652,033
	New York	\$9,348,304	\$27,652,033
	Total	\$11,375,413	\$27,652,033
Total		\$27,652,033	\$27,652,033

Report 2: Call center sales compared to total sales of call centers in targeted regions

Level metric: Grouping = None, Filtering = Absolute

In this scenario, the Regional Sales metric calculation includes the total for all the call centers present within all the regions listed in the report, and not just the call centers included in the report filter. The metric definition is shown below:

```
Sum(Revenue) {~+, !Region*}
```

With no grouping, the metric does not group by anything directly related to the target attribute specified within the metric. Since the attributes region and call center in this example are related to the target, there is no GROUP BY clause in the SQL as shown in the following example.

```

select sum(a11.[ORDER_AMT]) as REGIONALDOLL
from [ORDER_FACT] a11,[LU_EMPLOYEE]a12,
     [LU_CALL_CTR]a13
where a11.[EMP_ID] = a12.[EMP_ID]
and a12.[CALL_CTR_ID] = a13.[CALL_CTR_ID]
and ((a13.[REGION_ID])
in (select s21.[REGION_ID]
from [LU_CALL_CTR] s21
where s21.[CALL_CTR_ID] in (5,11,12)))

```

Also, with absolute filtering, the report filter is placed in the subquery only if it is of a lower level than the target. The report is shown in the following figure:

Report details				
Report Filter (Local Filter): {Call Center} = Washington, DC,Boston or New York				
Region	Call Center	Metrics	Revenue	Regional Sales { Target=Region, Filtering=Absolute, Grouping=None }
Mid-Atlantic	Washington, DC		\$16,276,620	\$34,554,237
	Total		\$16,276,620	\$34,554,237
Northeast	Boston		\$2,027,109	\$34,554,237
	New York		\$9,348,304	\$34,554,237
	Total		\$11,375,413	\$34,554,237
Total			\$27,652,033	\$34,554,237

Report 3: Call center sales compared to total sales of entire company

Level metric: Grouping = None, Filtering = Ignore

The Regional Sales metric calculates the total company sales for all call centers, ignoring the three call centers that are filtered out in the report filter. The metric definition is shown below:

```
Sum(Revenue) {~+, !Region%}
```

With no grouping, the metric does not group by anything directly related to the target attribute specified within the metric. Since the attributes region and call center are related to the target, there is no GROUP BY clause in the SQL as shown in the following example:

```
select sum(a11.[TOT_DOLLAR_SALES]) as REGIONALSALES
from [YR_CATEGORY_SLS] a11
```

The report is shown in the following image:

Report details			
Report Filter (Local Filter): {Call Center} = Washington, DC,Boston or New York			
Region	Call Center	Metrics	Regional Sales { Target= Region, Filtering= Ignore, Grouping= None }
Mid-Atlantic	Washington, DC		\$16,276,620
	Total		\$68,366,467
Northeast	Boston		\$2,027,109
	New York		\$9,348,304
	Total		\$11,375,413
Total			\$27,652,033

Report 4: Regional sales based on items sold in each call center in the region

Level metric: Grouping = None, Filtering = None

The Regional Sales metric calculates the total sales based on the number of items sold in each call center. The metric definition is shown below:

```
Sum(Revenue) {~+, !Region}
```

With no grouping, there is no GROUP BY clause for this metric calculation. With no filtering, you can define a fact of your choice in the calculation of a metric. This is accomplished by adding as many additional target attributes as necessary to the metric to force it to use the fact table that you want. Any target attribute that has no filtering borrows its filtering criteria from the other target attributes specified in the dimensionality of the metric. This allows you to choose the fact table but not alter the original intent of the report. The SQL statements for this example are as follows:

Regional Sales (Target=Region, Filtering=Standard, Grouping=Standard)

```
select a12.[REGION_ID] as REGION_ID,
       sum((a11.[QTY_SOLD]*
            a11.[UNIT_PRICE]-a11.[DISCOUNT])) as REGIONALDOLL
from [ORDER_FACT] a11, [LU_CALL_CTR] a12,
     [LU_EMPLOYEE] a13
where a11.[EMP_ID] = a12.[EMP_ID]
and a12.[CALL_CTR_ID] = a13.[CALL_CTR_ID]
and a11.[CALL_CTR_ID] in (5,11,12)
group by a12.[REGION_ID]
```

Regional Sales1 (Target1=Region, Filtering=Standard, Grouping=Standard, Target2=Item, Filtering=None, Grouping=Standard)

```
select a12.[REGION_ID] as REGION_ID,
       sum((a11.[QTY_SOLD]*
            (a11.[UNIT_PRICE]-a11.[DISCOUNT]))) as REGIONALDOLL
from [ORDER_DETAIL] a11, [LU_CALL_CTR] a12
where a11.[EMP_ID]=a12.[EMP_ID]
and a11.[CALL_CTR_ID]=a12.[CALL_CTR_ID]
and a11.[CALL_CTR_ID] in (5,11,12)
group by a12.[REGION_ID]
```

In this business scenario, if you want to use the Order_Detail fact table instead of the Order_Fact table, you include the Item attribute as the target. Since the Item attribute is found in the Order_Detail table and not in the

Order_Fact table, it forces the engine to use the Order_Detail fact table. The report is displayed in the following figure:

Report details				
Report Filter (Local Filter): {Call Center} = 'Washington, DC,Boston or New York				
Region	Call Center	Metrics	Revenue	Regional Sales { Target=Region, Filtering=Standard, Grouping=Standard }
Mid-Atlantic	Washington, DC		\$16,276,620	Regional Sales { Target1=Region, F=St, G=St, Target2=Item, F=None, G=None }
Northeast	Boston		\$2,027,109	
	New York		\$9,348,304	
Total			\$27,652,033	



In this example, Regional Sales is calculated using both the Order_Fact table and the Order_Detail fact table just to show that the data in the Order Detail fact table adds up correctly to match the data in the Order fact table.

Using a hierarchy as the target of a metric level

You can use a hierarchy as the target of a metric level to allow the level metric to dynamically adapt to the report. When placed on a report with attributes from that hierarchy, the metric calculates at the level of the lowest attribute of that hierarchy.

For example, a level metric with a target of Region calculates the regional revenue, regardless of what the report contains. If a level metric with the Geography hierarchy as the target is placed on a report with Region, the metric calculates the regional revenue. Placed on a report with Call Center, the metric calculates the revenue for Call Center. This example is explained in more detail in [Hierarchy level metric examples, page 49](#). The complete rules for replacing the target hierarchy are discussed in [Resolving a hierarchy in a level metric](#) below.



Attributes can be used as the target of a level metric to define the exact level of calculation for the metric. For information on using an attribute as the target of a level metric, see [Level metrics: Modifying the context of data calculations, page 14](#).

Resolving a hierarchy in a level metric

When a hierarchy is used as the target in a level metric, the hierarchy is automatically replaced with an attribute when the report is executed. The rules for replacing the hierarchy are listed below:



The examples described below use the Products hierarchy, which contains the following attributes, in order from highest to lowest: Category, Subcategory, and Item.

- If the report grid contains attributes from the target hierarchy, the lowest attribute of that hierarchy found on the report is used as the target.

A level metric with a target of Products, placed on a report with Category and Subcategory, calculates at the subcategory level. If the level metric uses standard grouping, the metric calculates a value for each subcategory. If the level metric is not grouped, the same number, the total for all the subcategories displayed on the report, is calculated for each row (Subcategory) of the report.

- If the report grid does not contain attributes from the target hierarchy, the report filter is checked. If the report filter contains attributes from the target hierarchy, the lowest attribute of that hierarchy found in the filter is used as the target.

A level metric with a target of Products, placed on a report with Region that is filtered for specific subcategories, calculates at the subcategory level. If the level metric uses standard filtering, the metric calculates only for the attributes in the filter. The same number, the total for all the subcategories on the report, is displayed for each row (Region) of the report. If the level metric uses absolute filtering, the target hierarchy is not included in the metric calculation, so the grand total across the project is displayed for each row.

- If neither the report grid nor the report filter contains attributes from the target hierarchy, then the hierarchy is ignored.

A level metric with a target of Products, placed on a report with Region that is filtered for Customer, calculates a grand total across the project, ignoring the target hierarchy.

If the level metric ignores filtering, the target hierarchy is ignored in all instances. The level metric calculates a grand total across the project for all rows of the report.

Hierarchy level metric examples

For example, the Geography Revenue metric is defined as:

- Formula: Sum(Revenue)
- Target: Geography hierarchy
- Filtering: Standard
- Grouping: Standard
- No report level

The Geography Revenue metric is placed on a report with attributes from the Geography hierarchy: Region and Call Center. Other metrics on the report include the Revenue metric and the Regional Revenue metric, which is a level metric with a target of Region. The report is filtered for Central and Mid-Atlantic. The report is shown below:

Report Filter: Region=Central, Mid-Atlantic					
Region	Call Center	Metrics	Revenue	Regional Revenue	Geography Revenue
Central	Milwaukee		\$4,182,139	\$5,029,366	\$4,182,139
	Fargo		\$847,227	\$5,029,366	\$847,227
	Total		\$5,029,366	\$5,029,366	\$5,029,366
Mid-Atlantic	Washington, DC		\$3,135,283	\$4,452,615	\$3,135,283
	Charleston		\$1,317,332	\$4,452,615	\$1,317,332
	Total		\$4,452,615	\$4,452,615	\$4,452,615
Total			\$9,481,981	\$9,481,981	\$9,481,981

- Revenue calculates at the level of the report, which is Call Center.
- Regional Revenue calculates at the level of Region. The Regional Revenue values for both call centers in each region is the same.
- Geography Revenue calculates at the level of the lowest attribute in the Geography hierarchy on the report, which is Call Center.

Remove Call Center from the report; the results are shown below:

Report Filter: Region=Central, Mid-Atlantic				
	Metrics	Revenue	Regional Revenue	Geography Revenue
Region				
Central		\$5,029,366	\$5,029,366	\$5,029,366
Mid-Atlantic		\$4,452,615	\$4,452,615	\$4,452,615
Total		\$9,481,981	\$9,481,981	\$9,481,981

All three metrics return the same numbers because:

- Revenue calculates at the level of the report, which is now Region.
- Regional Revenue continues to calculate at the region level.
- Geography Revenue calculates at the level of the lowest attribute in the Geography hierarchy on the report, which is now Region.

Add the Category attribute, which is not in the Geography hierarchy, to the report. The results are shown below:

Report Filter: Region=Central, Mid-Atlantic					
		Metrics	Revenue	Regional Revenue	Geography Revenue
Category	Region				
Books	Central		\$376,836	\$5,029,366	\$5,029,366
	Mid-Atlantic		\$337,656	\$4,452,615	\$4,452,615
	Total		\$714,492	\$9,481,981	\$9,481,981
Electronics	Central		\$3,506,062	\$5,029,366	\$5,029,366
	Mid-Atlantic		\$3,106,940	\$4,452,615	\$4,452,615
	Total		\$6,613,003	\$9,481,981	\$9,481,981
Movies	Central		\$589,357	\$5,029,366	\$5,029,366
	Mid-Atlantic		\$518,969	\$4,452,615	\$4,452,615
	Total		\$1,108,326	\$9,481,981	\$9,481,981
Music	Central		\$557,112	\$5,029,366	\$5,029,366
	Mid-Atlantic		\$489,049	\$4,452,615	\$4,452,615
	Total		\$1,046,161	\$9,481,981	\$9,481,981
Total			\$9,481,981	\$9,481,981	\$9,481,981

- Revenue calculates at the level of the report, which is now Region and Category.
- Regional Revenue continues to calculate at the region level.

- Geography Revenue calculates at the level of the lowest attribute in the Geography hierarchy on the report, which is still Region. The Geography Metric amount is the same for each region, regardless of category.

Replace Region with Call Center. The results are shown below:

		Metrics	Revenue	Regional Revenue	Geography Revenue
Category	Call Center				
Books	Washington, DC		\$235,588	\$4,452,615	\$3,135,283
	Milwaukee		\$313,177	\$5,029,366	\$4,182,139
	Fargo		\$63,659	\$5,029,366	\$847,227
	Charleston		\$102,068	\$4,452,615	\$1,317,332
Electronics	Washington, DC		\$2,193,709	\$4,452,615	\$3,135,283
	Milwaukee		\$2,916,573	\$5,029,366	\$4,182,139
	Fargo		\$589,489	\$5,029,366	\$847,227
	Charleston		\$913,231	\$4,452,615	\$1,317,332
Movies	Washington, DC		\$363,602	\$4,452,615	\$3,135,283
	Milwaukee		\$488,370	\$5,029,366	\$4,182,139
	Fargo		\$100,987	\$5,029,366	\$847,227
	Charleston		\$155,367	\$4,452,615	\$1,317,332
Music	Washington, DC		\$342,383	\$4,452,615	\$3,135,283
	Milwaukee		\$464,020	\$5,029,366	\$4,182,139
	Fargo		\$93,091	\$5,029,366	\$847,227
	Charleston		\$146,666	\$4,452,615	\$1,317,332

Now all three metrics return different results:

- Revenue calculates at the level of the report, which is now Category and Call Center.
- Regional Revenue continues to calculate at the region level.
- Geography Revenue calculates at the level of the lowest attribute in the Geography hierarchy on the report, which is now Call Center. The Geography Metric amount is the same for each call center, regardless of category.

Remove Call Center, so that no attributes in the Geography hierarchy remain on the report. The results are displayed below:

Report Filter: Region=Central, Mid-Atlantic				
	Metrics	Revenue	Regional Revenue	Geography Revenue
Category				
Books		\$714,492	\$9,481,981	\$9,481,981
Electronics		\$6,613,003	\$9,481,981	\$9,481,981
Movies		\$1,108,326	\$9,481,981	\$9,481,981
Music		\$1,046,161	\$9,481,981	\$9,481,981
Total		\$9,481,981	\$9,481,981	\$9,481,981

- Revenue calculates at the level of the report, which is now Category.
- Regional Revenue continues to calculate at the region level. Since no attributes from the Geography hierarchy are on the report, Regional Revenue calculates the revenue for all the regions on the report. Since the report is filtered for Mid-Atlantic and Central, only those regions are included in the calculation.
- Geography Revenue calculates at the level of the lowest attribute in the Geography hierarchy on the report. The report grid does not contain any Geography attributes, but the report filter does. Like Regional Revenue, Geography Revenue calculates the same number for each row, the sum of all regions in the report.

Remove the report filter, so that no Geography attributes remain in the report or report filter. The results are displayed below:

Report Filter: Empty Filter				
	Metrics	Revenue	Regional Revenue	Geography Revenue
Category				
Books		\$2,640,094	\$35,023,708	\$35,023,708
Electronics		\$24,391,303	\$35,023,708	\$35,023,708
Movies		\$4,098,943	\$35,023,708	\$35,023,708
Music		\$3,893,367	\$35,023,708	\$35,023,708
Total		\$35,023,708	\$35,023,708	\$35,023,708

- Revenue calculates at the level of the report, which is now Category. The number is higher than in the previous sample, since all regions are included in the calculation.
- Regional Revenue continues to calculate at the region level. Since no attributes from the Geography hierarchy are on the report or in the report

filter, Regional Revenue calculates the revenue for all the regions on the report. Without a report filter, all regions in the project are included, so this is a grand total of all revenue in the project.

- Geography Revenue calculates at the level of the lowest attribute in the Geography hierarchy on the report. Neither the report grid nor the report filter contains any Geography attributes. Like Regional Revenue, Geography Revenue calculates the same number for each row, the sum of all regions in the project.

Level metrics: Applying the metric condition to the metric calculation

The **Filter setting: uncheck to exclude attributes absent in report or level (dimensionality)** setting determines whether the metric filter is applied to the metric calculation. By default, the setting is selected. If it is cleared, filter attributes that are not on the report or in the level of the metric are not included in the metric calculation. For a detailed explanation of how the setting works, see [Example of Filter setting, page 53](#).

This setting can help you re-use the same metric in multiple reports. By clearing the Filter setting, the parts of the metric filter that are applied depend on what is included on the report. The example described in [Re-use metrics with the Filter setting, page 56](#), illustrates how the setting allows you to re-use metrics. For a definition and examples of metrics containing metric filters, see [Conditional metrics, page 63](#).



For brevity, this option is referred to as the Filter setting in the remainder of the chapter.

Example of Filter setting

The following reports all contain revenue for books sold in California stores and shipped via Pronto Packages, but the revenue amount changes depending on the Filter setting.

- 1 Create a filter with the following conditions and name it **CA Books Pronto**:
 - Call Center = San Diego and San Francisco
 - Category = Books

- Shipper = Pronto Packages
- 2 Create a revenue metric and use the CA Books Pronto filter as the condition. By default, the Filter setting is selected. Name it **Revenue (Attributes On)**.
- 3 Copy the Revenue (Attributes On) metric and rename it **Revenue (Attributes Off)**. Edit the metric to clear the Filter setting, by following the substeps outlined below:
 - Select **Level (Dimensionality)** in the breakdown window (under the heading Metric is defined as). The Definition window changes to display level options.
 - Click **Advanced** in the Definition window. The Level (Dimensionality) advanced options dialog box opens.
 - Clear the **Filter setting: uncheck to exclude attributes absent in report or level (dimensionality)** check box.
 - Click **OK** to return to the Metric Editor.
 - Click **Save and Close** to return to MicroStrategy Developer.
- 4 Create a report with the Region and Call Center attributes on the rows and the Revenue (Attributes On) metric on the columns. Execute the report. The results are displayed below:

		Metrics	Revenue (Attributes On)
Region	Call Center		
Northwest	San Francisco		\$4,615
Southwest	San Diego		\$16,226

- 5 Change to SQL view and notice the WHERE clause, as shown below:

```

where a11.[EMP_ID] = a12.[EMP_ID] and
      a..11[ITEM_ID] = a.13[ITEM_ID] and
      a13.[SUBCAT_ID] = a14.[SUBCAT_ID] and
      a11.[CUSTOMER_ID] = a15.[CUSTOMER_ID] and
      a11.[EMP_ID] = a15.[EMP.ID] and
      a11.[ORDER_DATE] = a15.[ORDER_DATE] and
      a11.[ORDER_ID] = a15.[ORDER_ID] and
      a12.[CALL_CTR_ID] = a15.[CALL_CTR_ID] and
      a16.[REGION_ID] = a17.[REGION_ID]
      and(a12.[CALL_CTR_ID] in (2, 4)
      and a14.[CATEGORY_ID] in (1)
      and a15.[SHIPPER_ID] in (1)

```

The complete metric filter (Call Center, Category, and Shipper) is included in the metric calculation.

- 6 Save the report as **CA Revenue (Attributes On)**.
- 7 Return to Design view. Delete the Revenue (Attributes On) metric and replace it with the Revenue (Attributes Off) metric. Execute the report. The results are displayed below:

		Metrics	Revenue (Attributes Off)
Region	Call Center		
Northwest	San Francisco		\$1,021,447
Southwest	San Diego		\$2,962,719

- 8 Why has the revenue increased substantially? Change to SQL view to check the WHERE clause:

```
where a11.[CALL_CTR_ID] = a12.[CALL_CTR_ID]
and a12.[REGION_ID] = a13.[REGION_ID]
and a11.[CALL_CTR_ID] in (2, 4)
```

With the Filter setting turned off, only those attributes in the metric filter which are on the report or in the metric level are included in the metric calculation. In this report, only Call Center meets those requirements, since it is on the template. Because the metric conditions of Category = Book and Shipper = Pronto Packages are excluded, the revenue is calculated for all categories and all shipping companies, increasing the revenue amount dramatically.



In the previous examples, the metric level has not changed from the default of report level, so the level does not affect the Filter setting. The next example in this procedure adds a metric level.

- 9 Save the report as **CA Revenue (Attributes Off)**.
- 10 Copy the Revenue (Attributes Off) metric, renaming it **Order Revenue (Attributes Off)**. Edit the metric to add Order to the metric level.
- 11 Copy the CA Revenue (Attributes Off) report, renaming it **Order CA Revenue (Attributes Off)**.

- 12** Edit the new report. Delete the Revenue (Attributes Off) metric and replace it with the Order Revenue (Attributes Off) metric. Execute the report. The results are displayed below:

		Metrics	Order Revenue (Attributes Off)
Region	Call Center		
Northwest	San Francisco		\$25,600
Southwest	San Diego		\$81,826

- 13** The revenue amount has changed again. Check the WHERE clause in the SQL view to discover why:

```
where a11.[EMP_ID] = a12.[EMP_ID] and
      a.11[CUSTOMER_ID] = a13.[CUSTOMER_ID] and
      a.11[EMP_ID] = a13.[EMP_ID] and
      a.11[ORDER_DATE] =a13.[ORDER_DATE] and
      a.11[ORDER_ID] = a13.[ORDER_ID] and
      a12.[CALL_CTR_ID] = a14.[CALL_CTR_ID] and
      a14.[REGION_ID] = a15.[REGION_ID]
and a12.[CALL_CTR_ID] in (2, 4)
and a13.[SHIPPER_ID] in (1)
```

Now the metric calculation includes Call Center because it is defined on the template. It also includes Shipper because it is in the same hierarchy as Order, which is the metric level. Category is not included, since it is neither on the report nor in the metric level. The metric calculates revenue in all categories for orders shipped by Pronto Packages for the California stores.

Re-use metrics with the Filter setting

The Filter setting can help you re-use the same metric in multiple reports. Clear the Filter setting so that the attributes on the report affect which parts of the metric filter conditions are applied. This eliminates the need to create and maintain multiple metrics, particularly if the metric and filter qualifications are complex.

For example, a revenue metric has a condition that filters for the Northeast region and the Electronics category. With the Filter setting cleared, attributes from the metric condition that are not on the report or in the level of the metric are not included in the metric calculation. Since the default of

report level has not been changed, only the attributes on the report will affect the metric calculation, as shown in the table below.

Metric Conditions	Attributes on the Report	Metric Results
Region = Northeast Category = Electronics	Category	Revenue for the Electronics category in all regions
Region = Northeast Category = Electronics	Call Center Category	Revenue for all Electronics in all Call Centers in the Northeast Region

Place the metric on a report that contains Category. Revenue is calculated for the Electronics category across all regions. The metric condition for Region is not included in the metric calculation because the Filter setting ignores any attributes in the metric condition that are not on the report.

Place the same metric on a report that contains Call Center and Category. Revenue is calculated for the Electronics category for all Call Centers in the Northeast Region. The metric condition for Region is now included in the metric calculation because the Filter setting only ignores any attributes in the metric condition that are not on the report or related to attributes on the report. Call Center and Region are in the same hierarchy and therefore related.

The same metric returns different results on different reports, because the report attributes determine which qualifications are used in the metric calculation. You could achieve the same results with two different metrics, each with its own filter. Re-using the metric and filter decreases the time spent creating and maintaining them, especially if the metric and filter are more complex than this simplified example.

The following procedure re-creates the example above, in particular showing how the report SQL is affected by the Filter setting.



For a more detailed description of metrics containing metric filters, including examples, see [Conditional metrics, page 63](#).

To re-use a metric with the Filter setting

- 1 Create a filter with the following conditions and name it **Northeast Electronics**:
 - Region = Northeast
 - Category = Electronics
- 2 Create a revenue metric and use the Northeast Electronics filter as the condition. Keep the metric level set at report level. By default, the Filter setting is selected. Save the metric as **Northeast Electronics Revenue -- selected**.
- 3 Copy the **Northeast Electronics Revenue -- selected** metric and rename it **Northeast Electronics Revenue -- cleared**. Edit the metric to clear the Filter setting, as described below:
 - Double-click the **Northeast Electronics Revenue -- cleared** metric. The Metric Editor opens.
 - Select **Level (Dimensionality)** in the breakdown window (under the heading Metric is defined as). The Definition window changes to display level options.
 - Click **Advanced** in the Definition window. The Level (Dimensionality) advanced options dialog box opens.
 - Clear the **Filter setting: uncheck to exclude attributes absent in report or level (dimensionality)** check box.
 - Click **OK** to return to the Metric Editor.
 - Save and close the metric.
- 4 Place the **Northeast Electronics Revenue -- cleared** metric on a report with **Category**. Switch to SQL View. As shown below, the WHERE clause does not filter for Region:

```
where a11.[CATEGORY_ID] = a12.[CATEGORY_ID]
and a11.[CATEGORY_ID] in (2)
```

This occurs because Region (or any related attributes) is not on the report or in the level of the metric.

- 5 Next, create a new report. Use the same metric, but this time add **Call Center** to the grid. The WHERE clause does not filter for Category:

```
where a11.[CALL_CTR_ID] = a12.[CALL_CTR_ID]
and a12.[REGION_ID] in (1)
```

Again, since Category is not on the report or in the level of the metric, the Category qualification does not affect the metric calculation. Call Center (from the report) is related to Region (from the metric condition) so only Northeast is included on the report. In both cases, the report attributes determine which qualifications are used in the metric calculation. The same metric returns different results.

- 6 To finish this example, place the **Northeast Electronics Revenue -- selected** metric on a new report with **Category**. Both the Region and Category qualifications are used because the metric's qualifications are not affected by the report's contents.

```
where a11.[SUBCAT_ID] = a12.[SUBCAT_ID]
and a.12[CATEGORY_ID] = A.13[CATEOGRY_ID]
and (a11.[REGION_ID] in (1)
and a12.[CATEGORY_ID] in (2))
```

Level metrics accepting dimensionality units to emulate MicroStrategy 6.x behavior

The **Allow other users to add extra units to this definition** setting is used to emulate MicroStrategy 6.x behavior and affects only those projects that have been upgraded from 6.x. The option indicates whether the metric accepts dimensionality units. It is applied to metrics used at the template level and metrics used in the filter for a metric qualification. This dimensionality is merged with the original units to complete the metric level.

This option is available on the Level (Dimensionality) advanced options dialog box, which is accessed from the **Advanced** button on the Level (Dimensionality) component window (shown in [Elements of a metric level, page 16](#)).

Nested metrics

Nested metrics, or nested aggregation metrics, are a type of simple metric, where one aggregation function is enclosed inside another. For example:

```
Avg (Sum (Revenue) )
```

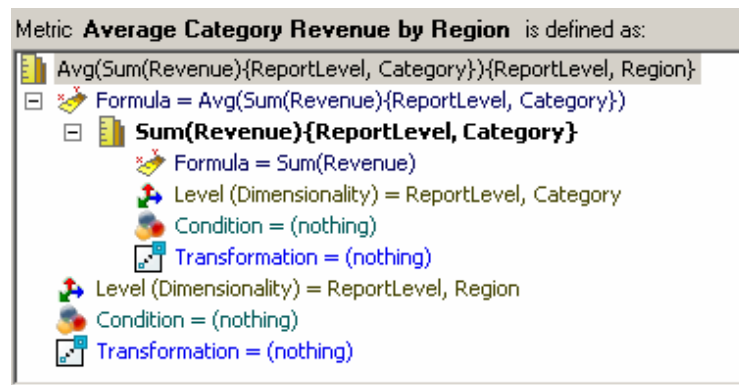
The inner function `Sum` calculates the total for the Revenue fact, and the outer function `Avg` calculates the average of that result.

Nested metrics are useful when the data warehouse does not store fact data at the level needed for a specific analysis, and are best used with each formula aggregated at a different attribute level. For information on levels, see [Level metrics: Modifying the context of data calculations, page 14](#).

For example, you want to know the average revenue per category for every region. You can create a nested metric that first calculates the revenue for all categories, and then calculates the average for each region. The metric is defined as:

```
Avg (Sum (Revenue) {~, Category}) {~, Region}
```

The inner formula, `Sum (Revenue)`, calculates the revenue at the Category level. The outer formula then calculates the average of the resulting values at the Region level. In the Metric Editor, the inner formula appears as a child of the outer formula, as shown below:



The following report uses the above metric to display the average category revenue for each region.

	Metrics	Average Category Revenue by Region
Region		
Central		\$1,257,342
Mid-Atlantic		\$1,113,154
Northeast		\$2,138,604
Northwest		\$440,297
South		\$1,347,320
Southeast		\$559,988
Southwest		\$923,533
Web		\$975,691

This is similar to creating a report that calculates the revenue by category for each region, and then enabling the Average subtotal, as shown below. The value of the Average subtotal is reflected in each row of the previous report:

Region	Category	Metrics	Revenue
Central	Books		\$376,836
	Electronics		\$3,506,062
	Movies		\$589,357
	Music		\$557,112
	Average		\$1,257,342
Mid-Atlantic	Books		\$337,656
	Electronics		\$3,106,940
	Movies		\$518,969
	Music		\$489,049
	Average		\$1,113,154
Northeast	Books		\$646,421
	Electronics		\$5,962,709
	Movies		\$1,001,561
	Music		\$943,724

Creating a nested metric

You can create a nested metric by directly typing the definition in the Metric Editor, or by adding an outer formula to a level metric that already exists in your project.



Both the inner and outer functions for the nested metric must be aggregation functions, such as Sum, Average, Maximum, and so on.

To type the definition for a nested metric

- 1 Under the **File** menu, select **New**, and then select **Metric**. The Metric Editor opens.

If the New Metric dialog box is displayed, click the **Empty Metric** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Choose **Empty Metric** and click **OK**.
- 3 In the Definition pane on the right, enter the formula for the metric. For example, `Avg(Sum(Revenue) {~, Category}) {~, Region}`, where `Revenue` is a fact.
- 4 Click **Validate**. You should receive a Valid Expression message in the status area. If not, check the formula for errors.
- 5 Click **Save and Close**.

To use an existing level metric

- 1 Under the **File** menu, select **New**, and then select **Metric**. The Metric Editor opens.

If the New Metric dialog box is displayed, click the **Empty Metric** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Choose **Empty Metric** and click **OK**.
- 3 In the Object Browser pane, browse to the level metric that you want to use.
- 4 Drag the metric into the Definition pane.
- 5 Enter the outer aggregation function, using the level metric as the argument. For example, `Avg([Category Revenue]) {~, Region}`,

where `Category Revenue` is a level metric, defined as `Sum (Revenue) {~, Category}`.

- 6 Click **Validate**. You should receive a Valid Expression message in the status area. If not, check the formula for errors.
- 7 Click **Save and Close**.

Conditional metrics

You can think of conditionality as a metric filter that is independent of the filters on any reports the metric is used in. A *conditional metric* allows you to apply a filter to only one metric on a report while not affecting the other metrics. The metric condition can be either a filter or a prompt that returns a list of filters. Only one filter or prompt can be associated with each metric, but that filter can contain multiple qualifications. See [Conditional metrics with multiple qualifications, page 65](#), for an example of the latter.



The terms metric filter and metric condition are used interchangeably to refer to the filter in a conditional metric.

The following report includes a simple example of a conditional metric, to quickly show you what a conditional metric is in its most basic form. In this example, you need to review regional revenue for different categories. You can create a report containing Category, the Revenue metric, and conditional metrics for the regions. In the following report sample, the Northeast Revenue and Mid-Atlantic Revenue metrics are conditional metrics. Northeast Revenue uses the same formula as the Revenue metric, with a filter for the Northeast region. The Mid-Atlantic Revenue metric was created in the same way.

Category	Metrics	Northeast Revenue	Mid-Atlantic Revenue
	Revenue		
Books	\$2,640,094	646,421	337,656
Electronics	\$24,391,303	5,962,709	3,106,940
Movies	\$4,098,943	1,001,561	518,969
Music	\$3,893,367	943,724	489,049

Conditional metrics can help you create more sophisticated analyses than the simple report above, as discussed in the following examples.

Conditional metric example

The following report compares the paid salaries and bonuses of different departments. It uses a prompted report filter on year; the report sample below contains 2003 data.

Department	Metric	Paid Salary	Paid Bonus
Sales		\$57,500	\$30,000
Marketing		\$61,833	\$37,000
Finance		\$46,250	\$20,000
HR		\$15,000	\$10,000
Technical Consulting		\$5,000	\$5,000
Program Management		\$52,500	\$30,000

The Paid Salary metric sums the Compensation Cost fact and has a condition, the Base Salary filter. This filter qualifies the Compensation Item attribute for Base Salary. The metric definition is shown below:

```
Sum([Compensation Cost]) {~+}
<[Base Salary]>
```

Notice that the filter name is placed between angle brackets (< >). (Additional symbols are described in [Metric level symbols: Defining expression syntax, page 17](#).) The Paid Bonus metric is similar to Paid Salary, except its condition is Regular Bonus, which filters the Compensation Item attribute for Bonus. The metric definition is shown below:

```
Sum([Compensation Cost]) {~+}
<[Regular Bonus]>
```

An additional requirement for this report is a comparison of the bonus amount to the salary amount. This is easy to do by combining the conditional metrics. Create a compound metric that divides the Paid Bonus metric by the Paid Salary. The metric definition is shown below:

```
([Paid Bonus] / [Paid Salary])
```

The new metric does not have its own condition; the condition on each of the constituent metrics is applied only to that metric. For a more detailed description of compound metrics, including examples and procedures, see [Compound metrics, page 95](#).

Conditional metrics with multiple qualifications

Only one filter can be associated with a metric, but that one filter can contain as many filter qualifications as needed. For example, you need a report showing a count of customers with the following qualifications:

- Female
- Living in the Central region
- Active customers

Create a filter with those three qualifications, apply the filter to the count metric, and add the metric to the report. The results are displayed below.

	Metrics		
Customer Lifetime Value Score		Active Customers - Current	Active Female Customers in Central Region
High		418	26
Medium		732	40
Low		213	17

The metric definition looks the same as any other conditional metric; the multiple filtering qualifications are contained within the filter placed on the metric. The metric definition is shown below:

```
[Count of Customers (Customer Lookup)] {~+} <[Active Female Customers in Central Region]>
```

Example of a conditional metric with a level

You can create a metric with both a condition and a level. For instance, you need a report that compares all regional sales to sales in the Central region.

Both conditions and levels are applied to metrics to achieve the desired result, which is shown below.

	Metrics			
Region		Revenue	Central Revenue	Variance from Central
Central		\$5,029,366	\$5,029,366	\$0
Mid-Atlantic		\$4,452,615	\$5,029,366	(\$576,751)
Northeast		\$8,554,415	\$5,029,366	\$3,525,048
Northwest		\$1,761,187	\$5,029,366	(\$3,268,179)
South		\$5,389,280	\$5,029,366	\$359,914
Southeast		\$2,239,951	\$5,029,366	(\$2,789,415)
Southwest		\$3,694,132	\$5,029,366	(\$1,335,235)
Web		\$3,902,762	\$5,029,366	(\$1,126,604)

The following descriptions of the metrics used in this report include metric definitions. In each metric definition, the information between the curly braces—{ }—indicates the metric level. Additional level options have been displayed in these definitions to indicate the grouping and filtering settings. For a table of symbols, see [Metric level symbols: Defining expression syntax, page 17](#). For more general information on level metrics, see [Level metrics: Modifying the context of data calculations, page 14](#).

- Revenue

`Sum(Revenue) {~+}`

In the metric definition above, `Revenue` is the Revenue fact. The `{~+}` indicates that this metric calculates as the report level. In other words, the metric calculates revenue for the attributes on the report, in this case, the regional revenue.

- Central Revenue

`Sum(Revenue) {~+, !Region+} <[Central]>`

where `Revenue` is the Revenue fact. This metric contains both a level (`Region`) and a condition (`Central`). It returns the revenue for the Central region, for each row on the report. How?

The condition allows data only from the Central region into the metric calculation, regardless of what the report contains. The metric now returns the correct data (\$5,029,366) for every row on the report. You might think that your Central Revenue metric is complete. However, if you place this on the report, only one row, Central, would be returned. Why?

Think of the metric results as tables of data. The table for the Revenue metric contains a row for each of the regions in the project, with its corresponding revenue amount. The table for the Central Revenue metric contains only one row, the revenue for the Central region. The tables are matched on the Region column to create the report. The only match is Central, so only that row is returned. How can more rows be included on the report?

Adding a level to the metric, with grouping set to none and a target of Region, returns eight rows of data in the report. The target is the attribute level at which the metric calculation groups. The target of the level is Region, but it can be any parent attribute of the attribute on the report (Region, in this case). So using Country would also work, since having the grouping set to none is more important than the actual attribute in the target.

No grouping means that the metric does not group on the target, so the single row of the Central region revenue is paired with each row of the Revenue metric table.

Standard filtering for the level, which is the default setting, allows the metric condition to apply to the metric calculation. Change the filtering to ignore, for example, and the metric condition is disregarded, defeating the purpose of the condition.

- Variance from Central

$(\text{Revenue} - [\text{Central Revenue}])$

where Revenue and Central Revenue are the metrics created above. This is a compound metric, since it uses an arithmetic operator (-). The levels and conditions set on the component metrics do not change when the metrics are combined in the compound metric, so the metric calculation is a straight-forward—Revenue - Central Revenue.

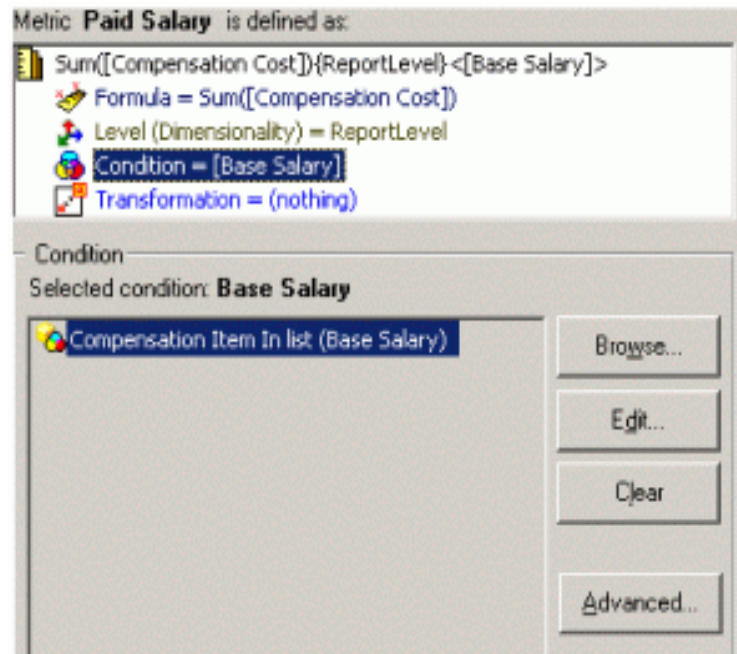


For a definition of compound metrics, including examples, see [Compound metrics, page 95](#).

Creating a conditional metric

Use the Metric definition area of the Metric Editor to set the metric condition. This area is split into two panes, the Metric component area on the top and the Definition pane on the bottom. The Metric component pane displays the complete metric definition, including its formula and calculation level, condition, and transformation components. When you select

Condition in the Metric component pane, the Definition pane displays the conditions for the metric, as shown below.



The steps below show you how to use the Metric Editor to create conditional metrics. The Metric Editor can be used to modify simple and compound metrics.

Rather than using the Metric Editor to apply a filter to a single metric, you can use the Advanced Metric Assistant to combine multiple, simple metrics with multiple filters. This can help reduce the time required to create the conditional metrics for a project, as described in [Creating metrics by combining metrics, filters, and transformations, page 99](#).

 You can also create, edit, or delete multiple metrics at the same time by using a Command Manager script. Command Manager is a MicroStrategy tool designed to automate certain tasks and processes. For more information about Command Manager, including steps to use Command Manager scripts, see the *Command Manager* chapter of the *System Administration Guide*.

To create a conditional metric

- 1 In MicroStrategy Developer, from the **File** menu, point to **New** and select **Metric**. The Metric Editor opens.

If the New Metric dialog box is displayed, click the **Empty Metric** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Create the metric formula by adding facts, attributes, metrics, operators, and/or functions. For example, to create the Paid Salary metric shown above, add the Compensation Cost fact from the Object Browser. The Sum function is automatically added. For detailed procedures to create basic metrics, see the *Building Query Objects and Queries, for Designers* chapter in the *Basic Reporting Guide*.



You can set conditionality for simple metrics but not for compound metrics. The formula of a simple metric is based on group functions (such as sum or average). For a quick review of simple metrics and formulas, see [Before you begin, page 12](#). For a more detailed description of simple metrics, including examples, see the *Basic Reporting Guide*.

- 3 Click **Condition** on the Metric component pane (located under the heading “Metric (metric name) is defined as”). The Condition definition pane displays below the Metric component pane.
- 4 Select the filter or prompt to place on the metric using one of the following methods:
 - Use the Object Browser to locate the object, then drag it into the Condition definition pane.
 - Click **Browse**. The Select a Filter dialog box opens, allowing you to select a filter from the displayed list, or navigate to locate the correct filter.



To see object prompts in the dialog box, select **Object Prompt** in the **Objects of type** list. Only an object prompt, which allows you to choose a filter, can be used as a metric condition.

- 5 To modify a filter or prompt, select it and click **Edit**. The Filter Editor or Prompt Generation Wizard opens. After you complete your changes, click

Save and Close (in the Filter Editor) or **Finish** (in the Prompt Generation Wizard) to return to the Metric Editor.

6 To remove a filter, select it and click **Clear**.

7 Save the metric.



The Advanced button accesses the Condition advanced options dialog box. The advanced options govern the interaction of conditional metrics and report filters, as described in [Embedding method, page 70](#), [Remove related report filter elements, page 75](#) and [Changing the condition advanced option defaults, page 87](#).

Conditional metric and report filter interactions

When a report contains a filter and a conditional metric, the report filter and the filter contained in the metric (the metric filter or metric condition) interact to produce the data for the conditional metric. The advanced options for the conditional metric establish this interaction. These options are described below.

- The [Embedding method, page 70](#) determines how the filters are merged by setting which filter is evaluated first. By default, the report filter is applied first.
- The [Remove related report filter elements, page 75](#) option is used when the report filter includes a qualification that uses an attribute related to an attribute in the metric condition. By default, the related attributes in the report filter are ignored.

These options are dependent on one another; although they each affect the filter interaction differently, they also work together. After you understand the options separately, refer to [Combining the embedding method and remove related elements settings, page 77](#) to learn how they come together to affect the conditional metric results.

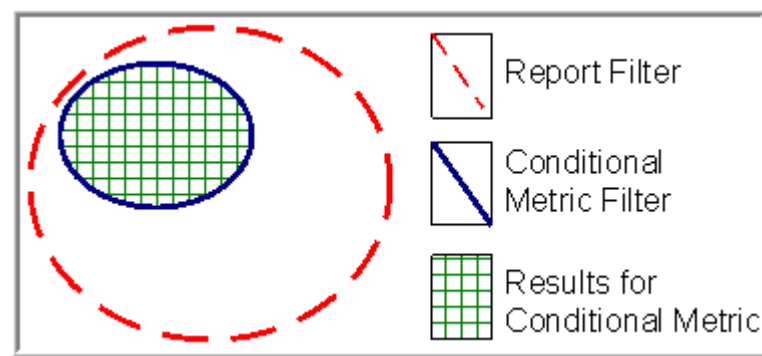
Embedding method

Merging the report filter and the metric filter (also known as the metric condition) is accomplished by embedding one filter in the other or by

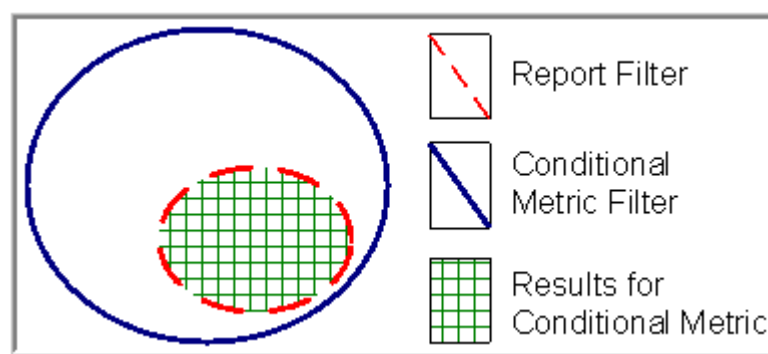
embedding both filters in a new, empty filter. The options are described below.

- The following diagrams represent the logic of the calculations, not the actual data values calculated by the conditional metric. The cross-hatched area represents the results of the conditional metric, indicating that results are calculated in each scenario. The cross-hatching is not meant to represent that those results will necessarily be identical sets of data.

- Merge report filter into metric** applies the report filter criteria to the data first. Then the metric filter is applied to the results of the first evaluation. This is the default setting. In the following diagram, the results for the conditional metric are represented by the cross-hatched circle.



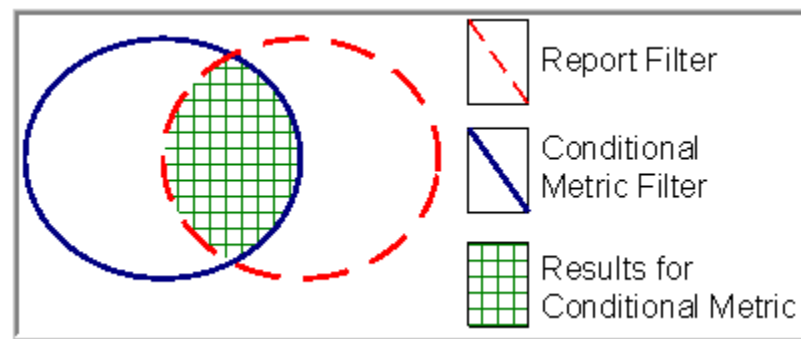
- Merge metric condition into report** evaluates the metric filter first, then applies the report filter to those results. While the following diagram may look similar to the previous one at first glance, notice that the outer circle is for the conditional metric filter.



- Selecting the Merge report filter into metric option includes the symbol @2 ; in the metric formula.

- Merge into new** intersects the metric filter and the report filter. Only those results that meet both the metric filter and the report filter are

returned. The intersection of the two filters, as shown by the cross-hatched area in the following diagram, is the data returned by the conditional metric on the report.



Depending on the embedding method you select, symbols are added to the metric formula as described in the table below:

Embedding Method	Symbol
Merge into new	@1;
Merge report filter into metric	@2;
Merge metric condition into report	@3;

A symbol is also added to determine if related report filter elements are removed, as described in [Remove related report filter elements, page 75](#).

The embedding method is relevant only if at least one filter contains a metric qualification, relationship qualification, or a shortcut-to-a-report qualification. The results from these types of qualifications can vary. For example, filtering on “country=US” always yields the same results. This is an example of an attribute qualification. However, filtering on “country where revenue is greater than \$1000” can return different results if the data is for 2002 only, 2003 only, or both years combined. This is a metric qualification.



For a list of descriptions of the different types of qualifications, see [Types of qualifications, page 127](#).

For example, you need to calculate revenue from high-volume customers, where high-volume is defined as a customer who has purchased more than 20 items. Create a conditional metric with a formula of `Sum (Revenue)`, setting the condition as the set of customers where `Units Sold > 20`.

You create a report containing the High-Volume Customer Revenue metric and a filter for the 75 most profitable items. How should the High-Volume

Customer Revenue metric be calculated? How should the report filter and the metric filter work together? Several possibilities exist, and the embedding method allows you to specify which possibility fits your business requirements. Note that both the metric filter and the report filter contain a metric qualification, so the embedding method is relevant to this example.

- Calculate the revenue for the 75 most profitable items, based on all customers, sold to the customers who bought more than 20 units of those top 75 items.

To do this, you must select the top 75 most profitable items, using all customers in the calculation. Next, find the customers that bought more than 20 units of those 75 items. Finally, the metric calculates the sales of those 75 items to the selected customers. Since the report filter contains the qualification for the profitable items, the report filter must be applied first.

Merge report filter into metric uses the results of the report filter to evaluate the metric filter. In this case, only one customer has bought more than 20 units of the selected items, as shown in the following report sample.

Metrics		High-Volume Customer Revenue (Merge report->metric)
Customer		
Klein	Norman	\$3,851

- Calculate the revenue for the top 75 most profitable items among customers who have purchased more than 20 units of any item.

To do this, you must first determine the customers who bought more than 20 units; these are the high-volume customers. Note that the units do not have to be of the same item. Next, find the 75 most profitable items that these big-volume customers bought. Finally, calculate the sales of those 75 items to the big-volume customers only. Since the metric filter contains the qualification for high-volume customers, the metric filter must be applied before the report filter.

Merge metric condition into report uses the results of the metric filter to evaluate the report filter. Norman Klein is included on the report, as

shown in the portion reproduced below, since we know from the previous example that he has bought more than 20 items.

		High-Volume Customer Revenue (Merge metric- >report)
Customer		
Aaronson	Maxwell	\$878
Abarca	Hugh	\$1,993
Abelson	Hazel	\$111
Abern	Brooks	\$1,439
Klein	Charles	\$1,501
Klein	Norman	\$3,851
Klipowski	Alexandria	\$1,657
		Rows: 9653

A total of 9653 customers are selected for this report, since the filter to find the high-volume customers is not restricted to specific items; any customer buying more than 20 units of any and all items is included.

- Calculate the revenue for the 75 most profitable items, based on all customers, sold to customers who have purchased more than 20 items.

To do this, you must determine the customers that bought more than 20 units of any items. Note that the units do not have to be of the same item. Select the top 75 most profitable items, using all customers in the calculation, not just the selected customers. Finally, calculate the sales of those 75 items to the selected customers only. You want to intersect the filters to obtain the results that meet both qualifications.

Merge into new allows you to calculate the report filter and metric filter independently, then intersect the results when calculating the metric. Norman Klein is included on the report, as shown in the portion

reproduced below, since we know from the first example that he has bought more than 20 items.

Customer		High-Volume Customer Revenue (Merge->new)
Aaronson	Maxwell	\$878
Abarca	Hugh	\$2,010
Abelson	Hazel	\$111
Abern	Brooks	\$1,439
Klein		\$1,501
Klein		\$3,851
Klipowski		\$1,674

Rows: 9655

Notice that the revenue calculated for Hugh Abarca in this report is \$2,010. In the second report, the value is shown as \$1,993. Why? The list of items included in the revenue calculation differs between the two reports. The second example includes only those items sold to the high-volume customers, while the third selects from all items in the database, sold to any customer.

A total of 9655 customers is selected for this report, slightly more than the previous example. Why? The second and third examples both start with the same list of customers, which is customers who have purchased more than 20 of any and all items. However, as explained above, the list of profitable products differs between the two reports. For the second report, if a customer's purchased units did not include any of the 75 most profitable items of the big-volume customers, he is not included on the report.

 For another description of this setting, see [Additional examples of the embedding method options, page 87](#).

Remove related report filter elements

The **Remove related report filter elements** setting influences the interaction between the metric filter and the report filter. When the check box is selected, if the report filter contains a qualification based on an attribute related to an attribute in the metric filter qualification, the metric filter qualification takes precedence. This is the default.

Depending on the embedding method you select, symbols are added to the metric formula as described in the table below:

Embedding Method	Symbol
Merge into new	@1;
Merge report filter into metric	@2;
Merge metric condition into report	@3;

A symbol is also added to determine if related report filter elements are removed, as described in [Remove related report filter elements, page 75](#).

For example, a report contains a Revenue metric and the Category attribute. The metric filter is set to the New York Call Center only and the report filter to all southern regions (South, Southeast, and Southwest).

Category	Metrics	New York Revenue
Books		\$533,459
Electronics		\$4,928,642
Movies		\$824,466
Music		\$779,912

Since Call Center and Region are related because they belong to the same hierarchy, the **Remove related report filter elements** setting is employed. The metric filter overwrites the report filter, so only New York revenue is included in the report. This setting allows you to ensure that the metric always calculates for the New York Call Center, regardless of whether other related elements are included in the report filter.

If the check box is cleared, the results are the intersection of the filters. In this case, New York and Southern regions exclude each other, so the combined filter is empty. The report returns no data. Clearing the option allows the metric filter to intersect with the report filter.



For another description of this setting, see [Additional examples of the embedding method options, page 87](#).

The report filter itself is not changed; the setting determines the filter interaction only for the conditional metric. For example, add the Revenue metric to the previous report and see the results below.

Category	Metrics	New York Revenue	Revenue
Books		\$533,459	\$857,351
Electronics		\$4,928,642	\$7,855,820
Movies		\$824,466	\$1,336,211
Music		\$779,912	\$1,273,981

Note that the Revenue metric does not have conditions or levels. It uses the report filter and therefore calculates revenue by category for all the southern regions.

If you select the **Remove related report filter elements** option, a plus (+) symbol is added after the symbol for the embedding method in the metric formula. If you clear the **Remove related report filter elements** option, a minus (–) symbol is added after the symbol for the embedding method in the metric formula. For information on the embedding method symbols, see [Embedding method, page 70](#).

Combining the embedding method and remove related elements settings

The embedding method and the Remove related report filter elements setting work together to affect the interaction of the report filter and the metric filter. The following list describes all the possible combinations of these advanced settings. A report sample is included for each combination to provide a concrete example.



If you create these reports, be aware that the years and data have been updated in MicroStrategy Tutorial; the reports will not display the same values as the samples.

The report samples build on the scenario described in [Embedding method, page 70](#). You need to calculate revenue from high-volume customers, where a high-volume customer is defined as having purchased more than 20 items. To calculate that revenue, use a conditional metric with a formula of `Sum (Revenue)`, setting the condition as the set of customers where `Units Sold > 20`. The report contains Year, the High-Volume Customer Revenue

metric, and a filter for the 75 most profitable items in the following Customer Regions: Central, Northwest, South, and Southwest.



These examples are also explained using diagrams, in [Combining the embedding method and remove related elements settings: A visual explanation, page 82](#).

- **Merge report filter into metric** with **Remove related report filter elements** selected. These are the default settings.

The report filter criteria is applied to the data first. Then the metric filter is applied to the results of the report filter evaluation. If the report filter contains a qualification based on an attribute related to an attribute in the metric filter, the attribute qualification in the metric filter overwrites the report filter qualification. This occurs only in the conditional metric and only for the related attributes.

In the sample report, the 75 most profitable items among customers in the specified Customer Regions are selected; these are the results from the evaluation of the report filter. The metric filter is applied to those results, so customers who bought more than 20 units of the selected items are chosen. The report filter condition on Customer Region is ignored because Customer Region is related to Customer in the metric filter.

In the report shown below, notice the revenue for the two years; as the options are changed in the following reports, the calculated revenue amount will change.

Metrics	High-Volume Customer
	Revenue (Merge report -> metric; Remove related)
Year	
2003	\$1,169
2004	\$2,657



This report is referred to as the **1-Merge report filter/Remove** report in the following sections.

- **Merge report filter into metric** with **Remove related report filter elements** cleared.

The report filter criteria is applied first, then the metric filter is applied to those results. All report filter elements, regardless of whether they are related to elements in the metric filter, are included in the criteria.

In the sample report, the 75 most profitable items among customers in the specified Customer Regions are selected; these are the results from the report filter. The metric filter is applied to those results, so customers in the specified Customer Regions who bought more than 20 units of the selected items are chosen. The results are shown below.

Metrics	High-Volume Customer Revenue (Merge report -> metric; Don't remove related)
Year	
2003	\$1,169
2004	\$2,657

 This report is referred to as the **2-Merge report filter/Keep** report in the following sections.

In this case, the calculated revenue is the same as the previous report (1-Merge report filter/Remove report), because the top 75 items among all customers are the same items as the top 75 items in the specified Customer Regions. Since the items are the same, whether Customer Region is removed from the filter does not change the final results.

 You can see samples of the SQL in [SQL samples of “Merge report filter into metric” reports, page 85](#) to see that the reports are different, even though the results are the same.

- **Merge metric condition into report** with **Remove related report filter elements** selected.

The metric filter criteria is applied to the data first. Then the report filter is applied to the results of the metric filter evaluation. Report filter elements that are related to the metric filter are ignored.

In the sample report, the customers who bought more than 20 units of any items are selected; these are the results from the evaluation of the metric filter. The report filter is applied to those results, so the 75 most profitable items among the selected customers are chosen. Notice that Customer Region does not figure in any of these evaluations. This is because the report filter condition on Customer Region is ignored since

Customer Region is related to Customer in the metric filter. The report results are displayed below.

Metrics	High-Volume Customer Revenue (Merge metric -> report; Remove related)
Year	
2003	\$5,986,768
2004	\$5,374,083

 This report is referred to as the **3-Merge metric filter/Remove** report in the following sections.

The calculated revenue amount is significantly larger than the previous two reports (the “Merge report filter” reports), since the number of customers included on the report is larger. This report includes all the high-volume customers, regardless of what items they bought. It then calculates the revenue of the most profitable items that those customers bought. The “Merge report filter” reports select the most profitable items first, then include only those high-volume customers who bought those particular items. The number of customers is therefore smaller in the “Merge report filter” reports.

- **Merge metric condition into report** with **Remove related report filter elements** cleared.

The report filter is applied to the results of the metric filter, which includes all its elements.

In the sample report, customers who bought more than 20 units of any item are selected; these are the results from the metric filter. The report filter is applied to those results, so the 75 most profitable items from the selected customers who are in the specified Customer Regions are chosen. The report results are shown below.

Metrics	High-Volume Customer Revenue (Merge metric -> report; Don't remove related)
Year	
2003	\$3,419,930
2004	\$3,083,509

 This report is referred to as the **4-Merge metric filter/Keep** report in the following sections.

As with the 3-Merge metric filter/Remove report, the metric values are significantly higher than the “Merge report filter” reports (\$3,419,930 vs. \$1,169). This is for the same reason—this report includes all the high-volume customers, regardless of what items they bought. However, these metric values are lower than the 3-Merge metric filter/Remove report because the 4-Merge metric filter/Keep report is filtered on Customer Region. The metric calculates revenue only for customers in the selected Customer Regions. The other report was calculated for all high-volume customers, regardless of the Customer Region.

- **Merge into new** with **Remove related report filter elements** selected.

The metric filter and report filter are calculated independently, and then the results are intersected during the metric calculation. If any report filter elements are related to any metric filter elements, the report filter elements are not included in the new filter.

In the sample report, the customers that bought more than 20 units of any items are selected. Note that the units do not have to be of the same item. This condition comes from the metric filter. The report filter determines the top 75 most profitable items, using all customers not just the selected high-volume customers. Finally, the sales of those 75 items only to the selected customers is calculated.

Notice that Customer Region does not figure in any of these evaluations. This is because the report filter condition on Customer Region is ignored since Customer Region is related to Customer in the metric filter.

In short, this report calculates the revenue for the 75 most profitable items, based on all customers, sold to high-volume customers. The results are shown below.

Metrics	High-Volume Customer Revenue (Merge -> new; Remove related)
	Year
2003	\$5,981,683
2004	\$5,372,001



This report is referred to as the **5-Merge into new/Remove** report in the following sections.

Notice that the revenue values are closer to those in the 3-Merge metric filter/Remove report than in the 1-Merge report filter/Remove report. The difference is that this report determines the revenue for the 75 most profitable items on all customers, while the 3-Merge metric filter/Remove report calculates the revenue for the 75 most profitable items

sold to high-volume customers only. The list of items included in the revenue calculation therefore differs. However, both reports start with the same list of customers, which is customers who have purchased more than 20 of any and all items.

- **Merge into new** with **Remove related report filter elements** cleared.

The results of the metric filter and report filter are intersected when the metric is calculated. All report filter elements, regardless of whether they are related to elements in the metric filter, are used to determine the results of the report filter.

In the sample report, the metric filter determines the high-volume customers. The report filter determines the top 75 most profitable items, using all the customers in the specified Customer Regions, not just the high-volume customers. When the revenue metric is calculated, it determines the sales of the selected items to the high-volume customers in the specified Customer Regions. The results are shown below.

Metrics	High-Volume Customer Revenue (Merge -> new; Don't remove related)
	Year
2003	\$3,420,330
2004	\$3,081,689



This report is referred to as the **6-Merge into new/Keep** report in the following sections.

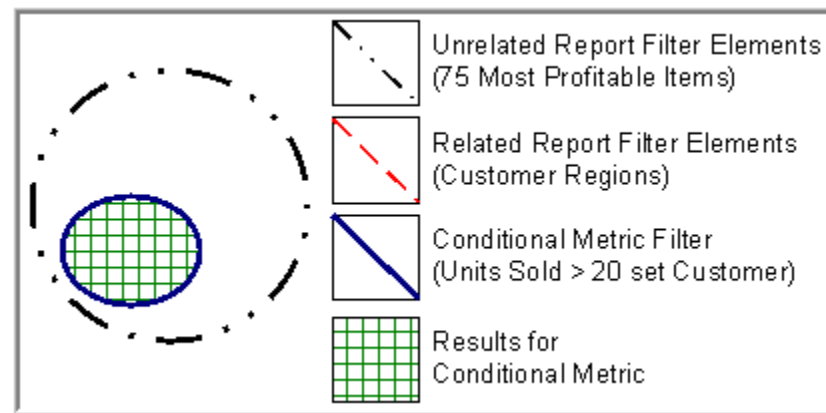
Notice that the revenue values are closer to those in the 4-Merge metric filter/Keep report than in the 2-Merge report filter/Keep report. The 4-Merge metric filter/Keep report and this report include all the high-volume customers, regardless of what items they bought. The metric values are lower than the 5-Merge into new/Keep report because this report filters for Customer Region.

Combining the embedding method and remove related elements settings: A visual explanation

The following diagrams provide a visual reference for the combinations discussed in the previous section.

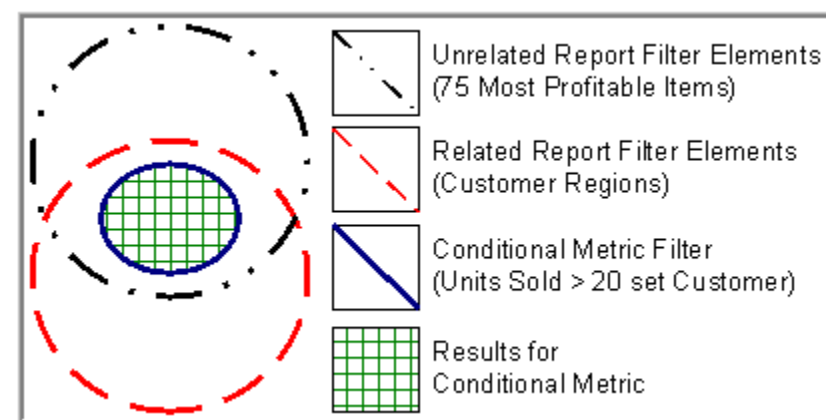
- **Merge report filter into metric** with **Remove related report filter elements** selected. These are the default settings.

Notice that the related report filter elements do not appear in the diagram, since Remove related report filter elements is selected. The results for the conditional metric are represented by the cross-hatched circle.



- **Merge report filter into metric** with **Remove related report filter elements** cleared.

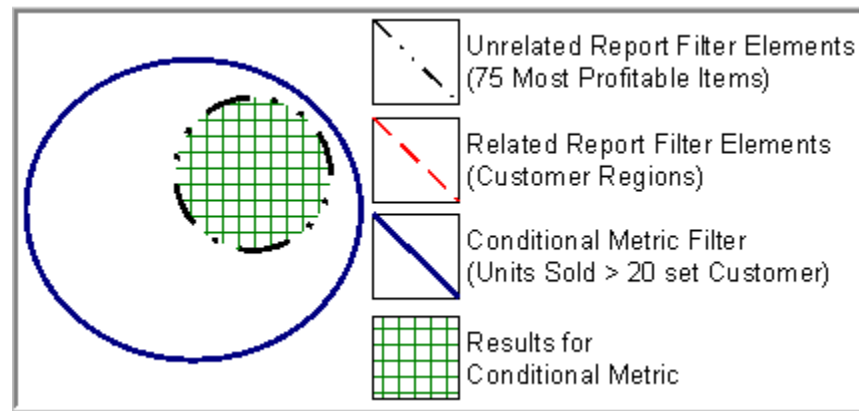
Notice that this diagram includes all report filter elements, even those related to the metric filter.



- **Merge metric condition into report** with **Remove related report filter elements** selected.

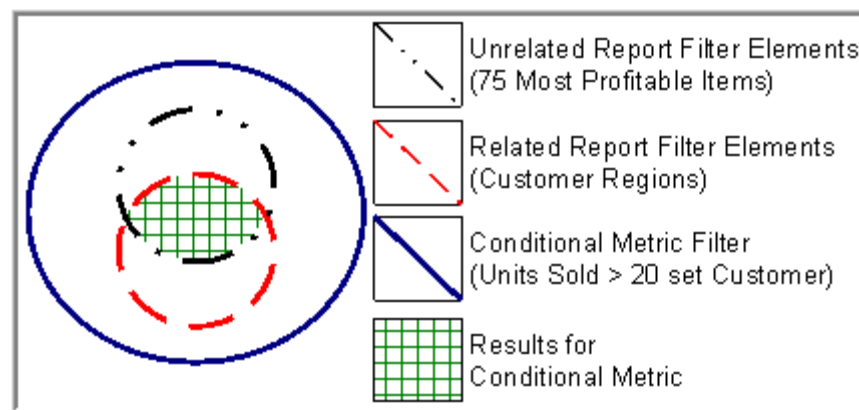
Again, this diagram does not include the related report filter elements since Remove related report filter elements is selected. The difference between this diagram and the first diagram is the outer circle—in the first

diagram it is the report filter, which is applied first. In this diagram, it is the metric filter that is applied first.



- **Merge metric condition into report** with **Remove related report filter elements** cleared.

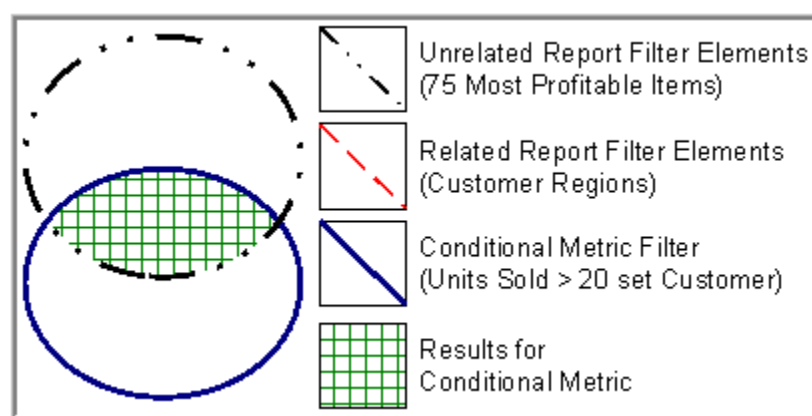
The report filter is applied to the results of the metric filter. This diagram includes all report filter elements, even those related to the metric filter, since Remove related report filter elements is cleared.



- **Merge into new** with **Remove related report filter elements** selected.

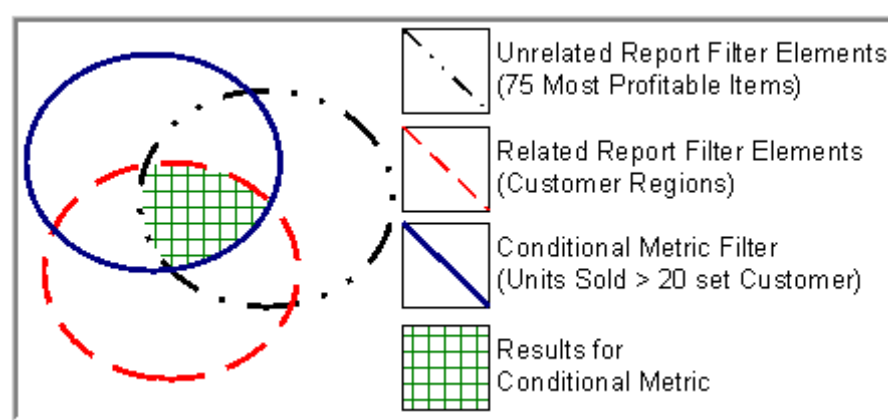
The metric filter and the report filter are intersected. Only those results that satisfy the qualifications of both the metric filter and the report filter are returned. The intersection of the two filters, as shown by the cross-hatched area in the following diagram, is the data returned by the conditional metric on the report.

Notice that the related report filter elements do not appear in the diagram, since Remove related report filter elements is selected.



- **Merge into new** with **Remove related report filter elements** cleared.

As with the previous diagram, the metric filter and the report filter are intersected. The difference is the third circle, which represents the related report filter elements. Since Remove related report filter elements is cleared, those filter qualifications are intersected with the other qualifications.



SQL samples of “Merge report filter into metric” reports

The two reports with the embedding method set to Merge report filter into metric returned the same revenue values. This is just an anomaly of the Tutorial data. The following SQL samples illustrate the difference between the reports. In the second sample, the bolded items are customer data and are not included in the first sample.



This is not the exact SQL from the reports; the samples have been edited for better readability.

Remove related report filter elements selected

```
select ORDER_DETAIL.CUSTOMER_ID AS CUSTOMER_ID
from ORDER_DETAIL, TempTable
where ORDER_DETAIL.ITEM_ID = TempTable.ITEM_ID
group by ORDER_DETAIL.CUSTOMER_ID
having sum(ORDER_DETAIL.QTY_SOLD) > 20.0
```

```
select LU_DAY.YEAR_ID AS YEAR_ID,
sum(REVENUE) from ORDER_DETAIL, TempTable, LU_DAY
where ORDER_DETAIL.CUSTOMER_ID = TempTable.CUSTOMER_ID
and
ORDER_DETAIL.ITEM_ID = TempTable.ITEM_ID and
ORDER_DETAIL.ORDER_DATE = LU_DAY.DAY_DATE
group by LU_DAY.YEAR_ID
```

Remove related report filter elements cleared

```
select ORDER_DETAIL.CUSTOMER_ID AS CUSTOMER_ID
from ORDER_DETAIL, TempTable, LU_CUSTOMER, LU_CUST_CITY,
LU_CUST_STATE
where ORDER_DETAIL.ITEM_ID = TempTable.ITEM_ID and
ORDER_DETAIL.CUSTOMER_ID =
    LU_CUSTOMER.CUSTOMER_ID and
LU_CUSTOMER.CUST_CITY_ID =
    LU_CUST_CITY.CUST_CITY_ID and
LU_CUST_CITY.CUST_STATE_ID =
    LU_CUST_STATE.CUST_STATE_ID
and LU_CUST_STATE.CUST_REGION_ID in (7, 6, 4, 5)
group by ORDER_DETAIL.CUSTOMER_ID
having sum(ORDER_DETAIL.QTY_SOLD) > 20.0
```

```
select LU_CUST_CITY.YEAR_ID AS YEAR_ID,
sum(REVENUE) from ORDER_DETAIL, TempTable, LU_DAY,
LU_CUSTOMER, LU_CUST_CITY, LU_CUST_STATE
where ORDER_DETAIL.ITEM_ID = TempTable.ITEM_ID and
ORDER_DETAIL.CUSTOMER_ID = TempTable.CUSTOMER_ID and
ORDER_DETAIL.ORDER_DATE = LU_DAY.DAY_DATE and
ORDER_DETAIL.CUSTOMER_ID = LU_CUSTOMER.CUSTOMER_ID and
LU_CUSTOMER.CUST_CITY_ID = LU_CUST_CITY.CUST_CITY_ID and
LU_CUST_CITY.CUST_STATE_ID = LU_CUST_STATE.CUST_STATE_ID
and
LU_CUST_STATE.CUST_REGION_ID in (7, 6, 4, 5)
group by LU_DAY.YEAR_ID
```

Changing the condition advanced option defaults

In most circumstances, you want the report filter to apply to the conditional metric, except for qualifications that contain attributes related to attributes in the metric filter. The conditional metric should be calculated based on the specified metric condition, regardless of whether any attributes from the same hierarchy appear in the report filter. Therefore, by default, the embedding method is set to Merge report filter into metric and the Remove related report filter elements check box is selected.

You can change these defaults, however. Choose the embedding method and whether to remove related elements, then select the **Remember option setting** check box. When you create another conditional metric, the new defaults are used.

Additional examples of the embedding method options

The following examples offer another scenario to explain how conditional metrics and report filters affect each other.

You want to identify the bottom 10 items in terms of revenue but you want to place an additional qualification to include only those items where sales are greater than \$100. You place the Bottom 10 qualification in the report filter and the Revenue greater than \$100 qualification in the metric condition. The results are shown below:

Item	Metrics	
	Revenue	
The Rules for Cats	\$112	
The Complete Prose of Woody Allen	\$120	
Guide to Life	\$120	

Only 3 rows are returned, instead of the 10 you expected based on your Bottom 10 qualification. Why?

By default, the report filter is applied first and then the metric filter is applied to that result set. This means that the bottom 10 sales are determined first, and then, of those bottom 10 sales, only those items with sales greater than \$100 are returned on the report. The number of rows on the report is only 3, since most of the 10 items returned by the report filter have sales less than \$100.

If you place the bottom 10 qualification in the metric and the revenue greater than \$100 qualification in the report filter, the report results change, as shown below:

	Metrics	Revenue
Item		
Cabin Fever : Rustic Style Comes Home		\$248
Voyaging Under Power		\$180
40 Most Wanted Cats		\$130
The Rules for Cats		\$112
Great Comedians Talk about Comedy		\$144
The Complete Prose of Woody Allen		\$120
Guide to Life		\$120
Engines of Creation		\$240
The Evolution of Useful Things		\$198
Rescuing Prometheus		\$200

The report contains 10 rows, as expected, and all items have a revenue higher than \$100. Since the report filter is applied first, all items with sales greater than \$100 are found first. Then the metric condition is applied to those results. The bottom 10 revenue items are then selected, so the report returns 10 rows.

To get the results you want, you can switch the qualifications, as shown in these examples above, or you can change the embedding method. The previous two examples used the default embedding option, which is **Merge report filter into metric**.

Change the embedding option to **Merge metric condition into report**. Now the metric filter is evaluated first, and then the report filter is applied to those results. Therefore, the bottom 10 sales are determined and then, of those bottom 10, only those items with sales greater than \$100 are returned on the report. The report, shown below, is identical to the first report example. Only 3 rows are returned, since 7 of the 10 items returned by the metric filter have sales less than \$100.

	Metrics	Revenue
Item		
The Rules for Cats		\$112
The Complete Prose of Woody Allen		\$120
Guide to Life		\$120

Change the embedding option to **Merge into new**, and the metric filter and the report filter are intersected. Only those items that are in the bottom 10 in

terms of sales and have sales greater than \$100 should be included. The results should be the same as the above report sample, with 3 rows of data.

Item	Revenue
Test Your Baseball IQ	\$75
Art of Getting Even	\$90
The Rules for Cats	\$112
Test Your History I.Q.	\$42
The Complete Prose of Woody Allen	\$120
Test Your Own IQ Again	\$80
Games for the Superintelligent	\$88
Guide to Life	\$120
Mensa Lateral Thinking	\$88
How to Live with a Neurotic Dog	\$48

However, the report contains 10 rows of data. The bottom 10 items are listed, but some of them have revenue of less than \$100. It does not seem that the report filter is being applied. Why? The embedding method is not the only option for conditional metrics that affects the interaction between filters. The **Remove related report filter elements** setting also influences the relationships between filters. By default, related report filter elements are removed. In this example, the filters use the same element (Item), so the report filter is ignored. Only the metric filter, for the bottom 10 sales, is applied. For a detailed description of how this setting works, see [Remove related report filter elements, page 75](#). For a description of how the two settings interact, including a list of all the possible combinations, see [Combining the embedding method and remove related elements settings, page 77](#).

Advanced options for metric filter and report filter interaction

Both conditional metrics and level metrics have advanced options that affect filters. The difference is that the advanced option for level metrics applies the metric filter to the metric calculation, while the advanced option for

conditional metrics sets the interaction between the metric filter and the report filter.

Metric Type	Option Name	Description
Level	Filter setting: clear selection to exclude attributes absent in report or level (dimensionality)	Determines whether the metric filter is applied to the metric calculation. By default, the setting is selected. If it is cleared, filter attributes that are not on the report or in the level of the metric are not included in the metric calculation. For a description of level metrics, including examples, see Level metrics: Applying the metric condition to the metric calculation, page 53.
Conditional	Remove related report filter elements	Influences the interaction between the metric filter and the report filter. By default, the setting is selected. If the report filter contains a qualification based on an attribute related to an attribute in the metric filter qualification, the metric filter qualification takes precedence. For a more detailed description of how this setting works with the report filter, including examples, see Conditional metric and report filter interactions, page 70.

Transformation metrics: Time-based and other comparisons

You need to compare business unit costs on a quarterly basis. For example, the business unit costs are represented by the Account Amount fact, which is the actual amount of each recorded transaction. To calculate the Account Amount, create the following metric, called Actual Amount, to calculate the dollar value of the accounts:

```
Sum([Account Amount]) {~+}
```

To count last quarter's costs, you apply a transformation to your Account Amount metric. A transformation maps a specified time period to another time period. In other words, it applies an offset value, such as current quarter minus one quarter.

Transformation-style analysis can also be supported using the `Lag` and `Lead` functions provided with MicroStrategy. These functions can be used to define metrics that compare values from different time periods without the

use of transformation objects. For information on using these functions to support transformation-style analysis, see the *Functions Reference*.

Time-based transformations

Time transformations are used in metrics to compare values at different times, such as this year versus last year or current date versus month-to-date. The last year transformation maps each time period to its corresponding time period last year, while the month-to-date transformation maps each time period to a set of time periods that comprise the entire month to date.

MicroStrategy provides numerous prebuilt transformations, although you can create your own as needed. One of the MicroStrategy-provided transformations answers our needs for the following example.



Transformations are schema objects and therefore only a project designer with the Create schema objects privilege can create them. Accordingly, this section focuses on how to use transformations in metrics and reports, and provides only an overview of transformations in general. For information about transformations and how to create them, refer to the *Transformations* chapter of the *Project Design Guide*.

The definition of the new transformation metric is:

```
Sum([Account Amount]) {~+} | [Last Quarter] |
```

In the metric definition, a transformation is placed between pipe symbols (|). (See the table of symbols in [Metric level symbols: Defining expression syntax, page 17](#).) The two metrics are placed on a report with the Account

Type attribute, as well as other metrics. The results are shown in the following figure.

Account Type	Metrics				
		Actual Amount	Actual Amount - Last Quarter	Actual Amount - Last Quarter Difference	% Variance
Salaries		\$328,633	\$302,762	\$25,871	8.55%
High-Tech and Communications Expense		\$37,312	\$39,709	(\$2,397)	(6.04%)
Recruiting Expense		\$1,163	\$2,326	(\$1,163)	(50.00%)
Travel and Entertainment (T&E)		\$411	\$1,284	(\$873)	(68.02%)
Casual Labor		\$3,022	\$6,923	(\$3,901)	(56.35%)
Communications		\$9,806	\$9,675	\$130	1.34%
Shipping, Printing, Supplies		\$13,109	\$12,497	\$612	4.90%
Consulting and Advisory		\$31,320	\$12,908	\$18,412	142.64%
Depreciation		\$66,033	\$73,998	(\$7,964)	(10.76%)
Other General and Administrative		\$878	\$10,395	(\$9,516)	(91.55%)
Insurance		\$311,031	\$326,598	(\$15,567)	(4.77%)
Cost of Goods Sold		\$112	\$586	(\$473)	(80.85%)
Total		\$802,830	\$799,659	\$3,170	0.40%

Transformations are useful for such time-series analyses, which are relevant to many industries, including retail, banking, and telecommunications. Another typical example of this type of analysis is a TY/LY comparison (This Year versus Last Year). To calculate a variance or a growth percentage such as last year's revenue versus this year's revenue, a transformation is convenient and flexible, although there are alternatives.

For example, you can use filters to create the TY/LY comparison. To calculate this year's revenue, add a filter for this year to the Revenue metric. Similarly, to calculate last year's revenue, use the Revenue metric in conjunction with a filter for last year. However, a more flexible alternative is to use a previously created Last Year transformation in the definition of a new metric, called Last Year Revenue. With a single filter, on 2003 for example, the two metrics Revenue and Last Year Revenue give you results for 2003 and 2002, respectively. In this example, two filters were created for the report, while the transformation needs only one. The same transformation metric can be applied to a report with a different filter to achieve different results, while, without a transformation, new filters would have to be created to build a different report. Transformations are usually the most generic approach and can be re-used and applied to other time-series analyses.

Since a transformation represents a rule, it can describe the effect of that rule for different levels. For instance, the Last Year transformation intuitively describes how a specific year relates to the year before. It can also express how each month of a year corresponds to a month of the prior year. In the

same way, the transformation can describe how each day of a year maps to a day of the year before. This information defines the transformation and abstracts all cases into a generic concept. Therefore, you can use a single metric with a last year transformation regardless of the time attribute contained on the report.

Non-time-based transformations

While transformations are most often used for discovering and analyzing time-based trends in your data, not all transformations have to be time-based. For example, a transformation can map defunct product codes to new ones. An example of a non-time-based transformation is This Catalog/Last Catalog, which might use catalog_ID-1 to perform the transformation.

Year-to-date transformation example

You need to compare the number of units sold today to the number of units sold from January 1 through today. Create a metric that sums the fact Units Sold. Create a second metric identical to the first, but add the predefined Year to Date transformation. This metric definition is shown below:

```
Sum([Units Sold]) {~+} | [Year to Date] |
```

Place the two metrics on a report with the Item attribute. A sample of the results is shown in the following image, sorted by Item.

Item	Metrics	Units Sold	YTD Units Sold
100 Places to Go While Still Young at Heart		74	74
1984		108	108
3Com 10/100 CardBus		213	213
3Com 56K Cellular Modem PC Card		226	226
3Com Networking Kit		246	246
3Com OfficeConnect 56K LAN Modem		233	233
3Com Palm IIIx Connected Organizer		186	186
3Com Palm V Connected Organizer		246	246

The Units Sold and YTD Units Sold metrics calculate the same value. Why? The transformation has not been given a date to transform, so both metrics calculate the number of units sold for all time. In the transformation example at the beginning of this section, with the number of Web visitors,

the report contained the Date attribute and a filter on time. To correct the problem using this current example, add a report filter that prompts the user for a specific date. If a user answers 9/21/2004 to the prompt, the report returns the following:

Item	Metrics	Units Sold	YTD Units Sold
3Com OfficeConnect 56K LAN Modem		1	73
9 Steps to Financial Freedom		2	253
98 Degrees & Rising		4	559
A Boy Named Goo		2	552
A Bug's Life		6	723
Al Green's Greatest Hits		2	511
Alice In Chains		3	538
Alice in Wonderland		1	634

The Units Sold metric returns the number of units sold on 9/21/2004, while the YTD Units Sold metric calculates the units sold from 1/1/2004 until 9/21/2004.

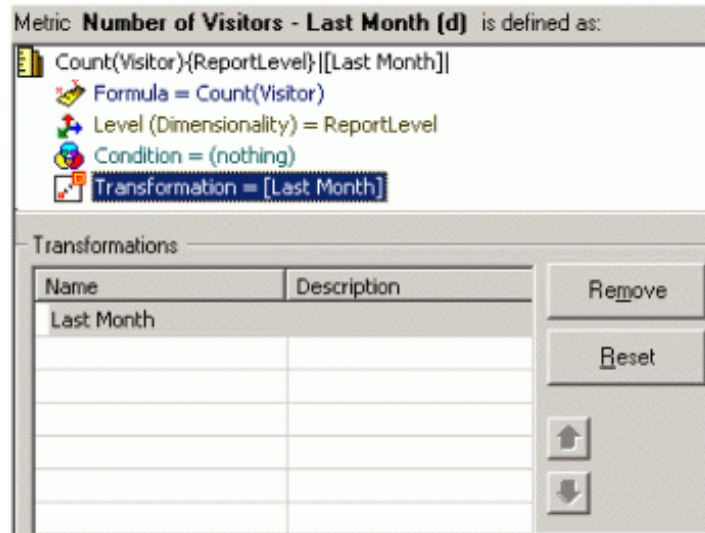


If you re-create these reports, be aware that the years and data have been updated in MicroStrategy Tutorial; the reports will not display the same values as the samples.

Transformation metrics in the Metric Editor

A metric transformation is set in the Metric Editor, using the metric definition window, which displays the complete metric definition, including its formula, level, condition, and transformation. When you select Transformation in the upper portion of this window, the Transformation

component window appears below the metric definition area, as shown below.



Use the Object Browser on the left side of the Metric Editor to select the transformation for the metric. To remove a transformation, select it in the Transformations window and click **Remove**.

You can add multiple transformations to the same metric. The arrows allow you to reposition the transformations and thereby change the order they are evaluated. To delete all the transformations, click **Reset**.

Rather than using the Metric Editor to apply a transformation to a single metric, you can use the Advanced Metric Assistant to combine multiple, simple metrics with multiple transformations. This can help reduce the time required to create the transformation metrics for a project, as described in [Creating metrics by combining metrics, filters, and transformations, page 99](#).

Compound metrics

The formula of a *compound metric* is based on arithmetic operators and non-group functions. Arithmetic operators are +, -, *, and /; non-group functions are OLAP and scalar functions such as running sum or rank. The operators and functions can be applied to facts, attributes, or metrics.



Several non-group functions that are frequently employed are described in [Metric functions, page 112](#).

The following are examples of compound metric formulas:

- `Sum(Cost) + Sum(Profit)`
where `Cost` and `Profit` are facts joined by an arithmetic operator
- `Cost + Profit`
where `Cost` and `Profit` are metrics joined by an arithmetic operator
- `RunningAvg(Cost)`
where `Cost` is a metric and `RunningAvg` is a non-group function

You cannot set the level, conditionality, or transformation for a compound metric as a whole. A compound metric draws this information from each of its component metrics; each component metric can have its own levels, condition, and transformations. For example, levels can be set on the `Cost` and `Profit` metrics in the examples above, but not on the formula of the compound metric that contains them.

However, compound metrics, unlike simple metrics, can have smart subtotals, which allow a different evaluation order for the calculation. For more information on smart subtotals (also called smart metrics), see [Smart subtotals, page 98](#). You can also control how the expression or metrics within the compound metric are joined, as described in [Formula join type for compound metrics, page 99](#).

Compound metric example

The following report compares actual amounts to forecasted amounts. It is used to assess differences between projected amounts, referred to as

forecasts, and the actual amounts for the selected time period. This report sample presents only a subset of the entire report, as it is a long report.

		Global Enterprises USA			
		Actual Amount	Forecast Amount	Actual - Forecast Amount Difference	Actual - Forecast Amount Variance
Account Class	Account Type				
Revenue					
	Product Income	\$91,236,247	\$80,262,162	\$10,974,084	13.67%
	Subscription Income	\$84,160,073	\$60,743,281	\$23,416,792	38.55%
	Advertising Income	\$5,866,470	\$5,700,440	\$166,030	2.91%
	Affiliate Fee Income	\$12,443,817	\$8,611,179	\$3,832,638	44.51%
	Software Royalties	\$326,375	\$251,013	\$75,362	30.02%
	Other Revenue	(\$398)	(\$593)	\$196	(32.98%)
Expense					
	Salaries	\$8,308,705	\$4,551,278	\$3,757,428	82.56%
	High-Tech and Communications	\$36,099	\$23,395	\$12,705	54.31%
	Recruiting Expense	\$156,026	\$102,854	\$53,172	51.70%

The first two metrics, Actual Amount and Forecast Amount, are simple metrics. The Actual Amount is the sum of the fact Account Amount, which is the dollar value of all the accounts. The Forecast Amount is the sum of the fact Account Forecast Amount, which is a calculated value generally assigned to the accounts quarterly and usually adjusted monthly. These definitions are shown below:

```
Sum([Account Amount]) {~+}
```

```
Sum([Account Forecast Amount]) {~+}
```

The {~+} indicates that the metrics are calculated at the report level; for an explanation of how the report level affects metric calculation, see [Report level: Interaction with the context of reports, page 34](#). See the table of symbols in [Metric level symbols: Defining expression syntax, page 17](#).

The last two metrics on the report are compound metrics based on these simple metrics. The metric definition for Actual - Forecast Amount Difference simply subtracts Forecast Amount from Actual Amount, as shown below:

```
([Actual Amount] - [Forecast Amount])
```

The Actual - Forecast Amount Variance metric determines the ratio of the difference to the forecast amount, as shown below:

```
((1.00001 * ([Actual Amount] - [Forecast Amount])) /  
(1.00001 * [Forecast Amount]))
```

Notice the lack of symbols for levels, conditions, or transformations in the formulas of the compound metrics. If this information were present, it would come from the simple metrics. In this case, the compound metrics are calculated at the report level.

Smart subtotals



Smart subtotals are also referred to as *smart metrics*.

A compound metric, at a high level, can be composed of multiple objects joined by arithmetic operators, such as `Metric1/Metric2` or `(Fact1 + Fact2) / Fact3`. The subtotal of a compound metric can be calculated in different ways:

- Calculate the sum of all parts of the compound metric, then calculate the compound metric. This formula is represented by `Sum(Metric1) / Sum(Metric2)`.
- Calculate the compound metric for each row of the report, and then roll up the data to the correct level. This formula is represented by `Sum(Metric1/Metric2)`.

The first case uses **smart subtotals**, which calculate subtotals on the individual elements of a metric. For example, the Profit Margin metric is calculated as the Profit metric divided by the Revenue metric. The Profit Margin metric can be totaled as follows:

- Add all the profit values together. Add all the revenue values together. Divide the two sums. This is a smart metric.
- Divide each profit value by each revenue value. Sum up these ratios.

These different totals are both shown in the following report.

Metrics	Profit	Revenue	Profit Margin	Smart Profit Margin
Year				
2003	\$2,153,701	\$8,876,630	24.26%	24.26%
2004	\$1,945,407	\$8,059,804	24.14%	24.14%
Total	\$4,099,108	\$16,936,434	48.40%	24.20%

Notice that the totals for the two Profit Margin metrics are different. Profit Margin simply sums the Profit Margin column, adding 24.26% and 24.14%

to arrive at 48.40%. Smart Profit Margin divides the Profit total by the Revenue total, calculating the correct percentage of 24.20%.



The smart metric setting is applied in the Metric Editor. To toggle smart metrics on and off, use the **Allow Smart Metric** check box at the bottom of the Subtotals/Aggregation tab.

Formula join type for compound metrics

A compound metric contains multiple expressions or metrics. You can define how these elements are joined using the Metric Formula Join Type dialog box. This dialog box is accessed from the **Advanced Settings** option in the **Tools** menu of the Metric Editor. The join types for a metric base formula are as follows:

- A **default join** is a join that is defined in each element.
- An **inner join** includes only data that is common across all elements.
- An **outer join** includes data that apply to every metric in a report.

Creating metrics by combining metrics, filters, and transformations

Rather than using the Metric Editor to apply a filter or transformation to a single metric, you can use the Advanced Metric Assistant to combine multiple metrics with multiple filters or transformations. This can help reduce the time required to create the metrics for a project.



You can also create multiple metrics at the same time by using a Command Manager script. Command Manager is a MicroStrategy tool designed to automate certain tasks and processes. For more information about Command Manager, including steps to use scripts, see the *Command Manager* chapter of the *System Administration Guide*.

The Advanced Metric Assistant allows you to combine simple metrics with filters or transformations to create metrics with additional analytical capabilities. Combining metrics with filters allows you to apply conditionality to your metrics (see [Conditional metrics, page 63](#)), while combining metrics with transformations allows you to compare values at

different times (see [Transformation metrics: Time-based and other comparisons, page 90](#)).

Using the Advanced Metric Assistant, you can combine simple metrics with filters, transformations, or both to create metrics with additional analytical capabilities. The steps below show you how to use the Advanced Metric Assistant to combine simple metrics with filters, transformations, or both.

Prerequisites

- You have created simple metrics to combine with filters, transformations, or both. Compound metrics, level metrics, and metrics that already use conditions or transformations cannot be used with the Advanced Metric Assistant. For information on creating simple metrics, see the *Basic Reporting Guide*.
- If you plan to create conditional metrics, you must have created filters. For information on creating filters, see [Advanced Filters, page 125](#).
- If you plan to create transformation metrics, you must have created transformations. For information on creating transformations, see the *Project Design Guide*.

To combine metrics with filters, transformations, or both

- 1 In MicroStrategy Developer, log in to a project.
- 2 From the **Tools** menu, select **Advanced Metric Assistant**. The Advanced Metric Assistant opens.



You can also access the Advanced Metric Assistant by first selecting metrics. Select a metric or metrics to combine with filters to create conditional metrics. Right-click your selection, point to **Create Advanced Metrics**, and select **Add Filter** or **Add Transformation**. The Advanced Metric Assistant opens.

To select metrics

- 3 From the **Metrics** tab, you can include a metric or metrics to combine with filters or transformations. Use the arrows to add or remove metrics to combine with filters or transformations to create new metrics.

To select filters

- 4 If the Filter tab is not shown, from the **Options** menu, select **Show Filters**.

You can also choose to hide this tab if you do not plan to combine metrics with filters. To hide the Filter tab, from the **Options** menu, select **Hide Filters**.

- 5 From the **Filters** tab, browse to and select the filters to combine with the metrics you selected. Use the arrows to add or remove filters to use to create conditional metrics.

To select transformations

- 6 If the Transformation tab is not shown, from the **Options** menu, select **Show Transformations**.

You can also choose to hide this tab if you do not plan to combine metrics with transformations. To hide the Transformation tab, from the **Options** menu, select **Hide Transformations**.

- 7 From the **Transformations** tab, browse to and select the transformations to combine with the metrics you selected. Use the arrows to add or remove transformations to use to create transformation metrics.

To select the metric combinations to create

- 8 If you include both filters and transformations to combine with metrics, you can specify the types of metric combinations created, as described below:
 - Metrics are always created to combine each metric with all filter and transformation combinations. These metrics include both a filter and a transformation.
 - To create metrics that combine metrics with transformations only, select the **Include no filter** check box. These metrics do not include the selected filters.
 - To create metrics that combine metrics with filters only, select the **Include no transformation** check box. These metrics do not include the selected transformations.



To review the metrics you are about to create by selecting or clearing the options above, click **Next** to open the Run page. You can then click **Back** to make any required changes.

To create metrics

- 9 Once you are finished selecting metrics, filters, and transformations, click **Next**. The Run page opens.

- 10 The metrics that will be created are displayed. The names and locations for potential metrics are listed, along with the status of creating the metrics. Click **Run** to create the metrics listed.
- 11 The metrics are created in the location specified. If a metric cannot be created, you can double-click the metric to view a message describing why the metric creation failed.
- 12 Once all required metrics have been created, click **Exit** to close the Advanced Metric Assistant.

Once the metrics are created, you can then begin to use these metrics on reports and documents. You can also modify the metrics using the Metric Editor. For Help documentation on using the Metric Editor, with the Metric Editor open, press **F1**.

Metric subtotals

In the context of metrics, subtotals allow you to compute and display quantified data along attribute groupings that you can specify dynamically for a report. The behavior of subtotal aggregations is based on the types of data included in the metric to which the subtotal is applied.

Displaying subtotals on reports

In a report, you can display grand totals by pressing **F11** or selecting **Grand Totals** from the **Data** menu. By default, the metric totals are calculated using the sum function. You can put other subtotals on the report by selecting **Subtotals** from the **Data** menu, then choosing from the list of available subtotals. By default, this list contains the standard subtotal functions such as total, count, minimum, maximum, standard deviation, and others. For a list of the standard subtotal functions, see [Standard subtotal functions, page 103](#). You can set the display location and how the subtotal is calculated on the report, as described in [Totaling and subtotaling data on reports, page 288](#).



You can select the value that is displayed in place of a null value in a subtotal. For steps, see the *Report Formatting* chapter in the *Basic Reporting Guide*.

Defining subtotal defaults

For each metric, you can change the default grand total function and add functions to the list of available subtotals. This allows you to use different grand total functions for different metrics on the same report, for instance. These options are found on the **Subtotals/Aggregation** tab of the Metric Editor. The default grand total is called the **Total subtotal function**. You can add functions to the list of available subtotals or remove functions. The functions listed here are available as subtotals in reports.



You can disable totaling on metrics that should not be subtotaled, such as inventory metrics or other non-aggregatable metrics. For steps to define the total subtotal function and available subtotals to disable subtotals, see [Disabling subtotals, page 111](#).

Your choices for both the total subtotal and available functions are standard predefined subtotal functions, which are simple aggregate functions such as Total and Count that satisfy many subtotalling requirements. If they do not answer your specific needs, you can create a user-defined subtotal and use it just like any of the built-in subtotal functions. For a list of the predefined subtotal functions, see [Standard subtotal functions, page 103](#). For an explanation of when to use user-defined subtotals, a list of objects that can be used in them, and examples, see [Creating your own subtotals, page 104](#).

Standard subtotal functions

The predefined subtotal functions in the following table are automatically available for use with every metric and report. They are simple aggregate functions designed to satisfy many subtotalling requirements.

Subtotal function	Description
Count	Count [count] number of input values
Average	Avg [average] sum of input values divided by number of input values
Minimum	Min [minimum] smallest input value
Maximum	Max [maximum] largest input value
Product	Product [product] all input values multiplied together
Median	Median [median] middle value when all values are sorted
Mode	Mode [mode] most frequently found input value
Standard deviation	Stdev [standard deviation] distribution of input values

Subtotal function	Description
Variance	Var [variance] square of the distribution of input values
Geometric mean	Geomean [geometric mean] square root of the product of input values

These functions reflect only those most frequently used for evaluating metric subtotals. The metric subtotal calculations and consolidations needed to create a report are performed by the Analytical Engine, which is a component of MicroStrategy Intelligence Server. In addition to the above standard functions, the Analytical Engine also handles a large number of statistical, mathematical, financial, date-and-time, string, and OLAP functions, ranging from simple to highly complex.

See the *Functions Reference* for examples and details on every built-in function available in MicroStrategy.

Creating your own subtotals

The standard predefined subtotal functions, which are automatically available for use with every metric and report, are simple aggregate functions that satisfy many subtotalling requirements. If they do not answer your specific needs, you can create your own *user-defined subtotal* using the Subtotal Editor. You can then use it in subtotal definitions just like any of the built-in subtotal functions such as Total and Count.

You can create your own subtotal using any combination of the following:

- An aggregation function, such as avgdev, IRR, MIRR, and NPV, that is not one of the standard predefined subtotal functions
- Multiple functions
- Constants in conjunction with aggregation functions
- Nested functions
- Dimensional subtotals
- Other metrics in the subtotal formula

For example, you need a subtotal that always calculates at the regional level, regardless of the level of the report. You can create a user-defined subtotal, setting it to the Region level (or dimension). Another example is a weighted subtotal, where a subtotal is weighted with another metric, such as an

average profit weighted by units sold. This example is included in the [User-defined subtotal example \(weighted subtotal\), page 108](#).

After a user-defined subtotal is created, it is indistinguishable from the standard predefined subtotal functions such as total or count. For a particular metric, you can choose a user-defined subtotal as the total subtotal function or add it to the list of available subtotals. When you run a report containing this metric, the total subtotal function calculates the grand totals for the metric. Select from the available subtotals to display additional subtotals for the metric. For more information, see [Displaying subtotals on reports, page 102](#).



Do not confuse user-defined subtotals with custom report subtotals, which are defined for a particular report for display purposes. User-defined subtotals are made available to metrics and can then be used on any report that uses the metric. For more information on custom report subtotals, see [Custom report subtotals, page 294](#).

You can create a user-defined subtotal with a level, and select the subtotal as the dynamic aggregation function for a metric, to allow you to specify the level of dynamic aggregation for that metric. (Dynamic aggregation is the roll-up of metric values that occurs when an attribute is moved from the report grid to the Report Objects.) You must also change the Subtotal Dimensionality Use VLDB property because the default behavior is to use the level of the metric that is being aggregated, not that metric's dynamic aggregation function. For a more detailed description of this VLDB property, along with an example of its use, see the *Supplemental Reference for System Administration*.

User-defined subtotal example (First function)

You need to produce an inventory report showing the number of electronics products received in 2003 by month and by product. The report must also provide the first shipment quantity. To do this, create a user-defined subtotal using the First function. Create a metric with the Units Received fact and set the **Total subtotal function**, which calculates the metric's grand totals, to the new subtotal. Next, create the report. When you apply subtotals, the user-defined subtotal displays the number of items received in the first shipment, regardless of the month in which it arrived.

Since this example focuses on user-defined subtotals and using the Subtotal Editor, steps are provided to create the subtotal. For details to create metrics and reports, refer to the online help.

To create and use a user-defined subtotal (using the First function)

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then select **Subtotal**. The Subtotal Editor opens. Notice its resemblance to the Metric Editor—their similarity in function and appearance helps to ease the process of creating subtotals.
- 2 In the Object Browser, navigate to the **Basic Functions** folder, located under **Functions and Operators**. Double-click the **First** function.
- 3 In the definition, right-click **First**, and select **First parameters**. The First Parameters dialog box opens.
- 4 The First function must be sorted to achieve correct results. Click the **Sort By** tab.
- 5 Select **Sort by objects**, then click **Add**. The Select Objects dialog box opens.
- 6 Since the report will be sorted by time, in the Object Browser select **Attributes**, then **Time**. Double-click the following, in the order they appear below, to add them to the sort:
 - Year
 - Quarter
 - Month
 - Day
- 7 Click **OK**, then **OK** again to return to the Subtotal Editor.



For more information on the First function, including details on sorting, see the *Functions Reference*.
- 8 Click **Validate**. You should receive a “Valid expression” message in the status area. If you do not, check the formula for any errors.
- 9 Click **Save and Close**. Name the new subtotal **First (Date Sort)**. You are returned to MicroStrategy Developer.
- 10 Create a new metric:
 - In the Metric Editor, select the **Units Received** fact.

- On the **Subtotals/Aggregation** tab, select **First (Date Sort)** as the Total subtotal function.
- Save the metric as **Units Received**.

11 Create a new report:

- In the Report Editor, place **Item** on the rows.
- Place **Month** and the **Units Received** metric on the columns.
- Filter on Year = 2003 and Category = Electronics.
- Add Grand Totals.

12 Execute the report. The results are displayed in the following report sample:

Item	Metrics	Units Received			
	Month	Jan 03	Feb 03	Mar 03	Total
Harman Kardon Digital Surround Sound Receiver			20	20	20
Harman Kardon AM/FM Stereo Receiver		20	20	20	20
Harman Kardon Dolby Digital Receiver		20	20		20
GPX Portable CD Player with Bass Boost		20	20		20
GPX Portable CD Player with Car Kit		20	20		20
GPX CD AM/FM Cassette Recorder		20	20		20
Karaoke Machine		20	20		20
Panasonic Portable CD Player		20	20		20
Pioneer 25-Disc Changer with Remote Control		20	20		20
Pioneer CD Recordable, Digital Synchro Recording		20		20	20
Pioneer Dolby Pro Logic Receiver		20		20	20
RCA CD Changer		20	20	20	20
Sony Discman		20		20	20
Sony Boombox with Digital Tuner		20	20		20
Sony Digital MiniDisc		20	20		20
Hitachi Hi8 Camcorder				40	40



This sample presents only a subset of the entire report, showing the first three months of the year and the total.

Notice that the Total does not add all units received throughout the year, but rather displays the amount of the first shipment, regardless of what month the shipment arrived. For example, you received 20 AM/FM Stereo Receivers each month; the total is 20, the amount received in January alone. No Digital Surround Sound Receivers were received in January, but 20 were

received in February, which the subtotal reflects. Hitachi Hi8 Camcorders, in the last line of the sample, were not received until March, so the total is derived from the March shipment.



Remember to save the report if you want to keep it.

User-defined subtotal example (weighted subtotal)

You need to create a report containing units sold and profit by call center. For each region, the report must calculate the average units sold and the average profit, weighted by the number of units sold in the region. Since these two averages must be displayed on the same line, a custom subtotal is needed. The formula for the weighted regional average profit is:

```
Sum(x*[Units Sold]){Region}/Sum([Units
Sold]){Region}
```

where:

- `x` is the placeholder for the metric to be subtotaled. In this case, it is the Profit metric.
- `[Units Sold]` is a metric that sums the fact Units Sold.
- `{Region}` is the level of the subtotal calculation.

For information on symbols, see [Metric level symbols: Defining expression syntax, page 17](#).

Finally, you need to sum the weighted regional average profit over all the regions. The formula for this subtotal is:

```
Sum(Sum(x*[Units Sold]){Region}/Sum([Units
Sold]){Region}){ }
```

which calculates the sum on an empty level, represented by the `{ }`. An empty level calculates across the entire report. This subtotal will be used as a grand total on the report. To use these formulas in any of the Analytics Modules, you need to create user-defined subtotals because they are not standard subtotal functions in the Analytics Modules. The MicroStrategy Tutorial, however, offers them as standard subtotals.

The focus of this example is user-defined subtotals and using the Subtotal Editor, so steps are provided to create the subtotal. For details to create metrics and reports, see the *MicroStrategy Developer help*.

To create and use a user-defined subtotal (using a weighted subtotal)

- 1 In MicroStrategy Developer, select **File**, point to **New**, and then **Subtotal**. The Subtotal Editor opens.
- 2 Type the formula for the weighted regional average subtotal, as described above.
- 3 Click **Validate**.
- 4 In the Ambiguity: 'Units Sold' dialog box, select the metric and click **OK**.
- 5 In the Ambiguity: 'Region' dialog box, select the attribute and click **OK**. You are returned to the Subtotal Editor, and you should see a "Valid expression" message in the status area. If you do not, review the formula and correct any errors.
- 6 Click **Save and Close**. Name the new subtotal **Weighted Regional Average**. You are returned to MicroStrategy Developer.
- 7 Open the Subtotal Editor again.
- 8 Type the formula for the sum of the weighted regional average subtotal, as described above.
- 9 Click **Validate**.
- 10 In the Ambiguity: 'Units Sold' dialog box, select the metric and click **OK**.
- 11 In the Ambiguity: 'Region' dialog box, select the attribute and click **OK**. You are returned to the Subtotal Editor, and you should see a "Valid expression" message in the status area. If you do not, review the formula and correct any errors.
- 12 Click **Save and Close**. Name the new subtotal **Sum of WRA** (which stands for Weighted Regional Average). You are returned to MicroStrategy Developer.
- 13 Open the Profit metric in the Metric Editor.
- 14 Click the **Subtotals/Aggregation** tab.
- 15 Select **Weighted Regional Average** and **Sum of WRA** in the **Project subtotals** list. Click **>** to add them to the metric.

16 Save the metric and close the Metric Editor.

17 Create the new report:

- a In the Report Editor, place **Region** and **Call Center** on the rows, and then the **Units Sold** and **Profit** metrics on the columns.
- b Next, create the subtotals for the report. Select **Subtotals** from the **Data** menu.
- c Click **Advanced**.
- d Click **New** to create a new custom subtotal, which will contain a standard average for Units Sold and the weighted regional average subtotal for Profit.
- e Type **Average, Weighted Regional Average** as the name of the custom subtotal.
- f For **Units Sold**, select **Average** from the pull-down list.
- g For **Profit**, select **Weighted Regional Average**.
- h Click **OK**.
- i Select **Across level** and then **Call Center**, so the subtotals are calculated for each region.
- j Next, create another custom subtotal to display in the grand total position. Click **New**.
- k Type **Overall Average, Sum of WRA** as the name.
- l For **Units Sold**, select **Average** from the pull-down list.
- m For **Profit**, select **Sum of WRA**.
- n Click **OK**.
- o Select **By position**. For **Rows**, select **Grand Total** from the pull-down list. For both **Columns** and **Pages**, select **None**.
- p Click **OK**. Notice your two custom subtotals are selected.

Notice that custom subtotals are distinguished by a different icon. An icon with an exclamation mark (!) means that the subtotals are not available for all metrics on the report. Recall that you added the user-defined subtotals to the Profit metric only, not the Units Sold metric.

- q Click **OK** to return to the Report Editor.

- r Execute the report. The results are displayed in the following image:

		Metrics	Units Sold	Profit
Region	Call Center			
Central	Milwaukee		100,720	\$637,545
	Fargo		20,646	\$126,778
	Average, Weighted Regional Average		60,683	\$550,657
Mid-Atlantic	Washington, DC		75,187	\$473,200
	Charleston		32,039	\$199,884
	Average, Weighted Regional Average		53,613	\$391,533
Northeast	Boston		35,860	\$224,495
	New York		170,458	\$1,076,237
	Average, Weighted Regional Average		103,159	\$928,196
Northwest	San Francisco		24,461	\$156,330
	Seattle		17,622	\$110,655
	Average, Weighted Regional Average		21,042	\$137,204
South	New Orleans		80,327	\$504,990
	Memphis		51,041	\$301,966
	Average, Weighted Regional Average		65,684	\$426,108
Southeast	Atlanta		25,654	\$157,963
	Miami		29,281	\$178,713
	Average, Weighted Regional Average		27,468	\$169,023
Southwest	San Diego		72,440	\$449,553
	Salt Lake City		17,828	\$111,779
	Average, Weighted Regional Average		45,134	\$382,842
Web	Web		93,909	\$583,538
	Average, Weighted Regional Average		93,909	\$583,538
Overall Average, Sum of WRA			56,498	\$3,569,101

Disabling subtotals

You can also choose to disable totaling on a metric such as an inventory metric or another non-aggregatable metric. To do this, set the total subtotal function to None and clear all the available subtotals for the metric. If you set the total subtotal function to None but do not clear the available subtotals, grand totals are unavailable in reports; in this scenario, the only way you can show grand totals is by defining subtotals. If you clear the Available subtotals for metrics list and set a total subtotal function, only that function is available for subtotal calculation.

Metric functions

Some useful functions to use in metrics include:

- [Rank, page 112](#)
- [Count, page 114](#)
- [Running and moving sums and averages, page 115](#) (also called OLAP functions)
- [N-tile, page 117](#)
- [First and Last, page 119](#)



All of the above except count and first/last are non-group functions. Non-group functions are used in compound metrics.

- [Creating your own plug-in functions, page 119](#)
- [Apply functions: Using database-specific functions in metrics, page 120](#)

The following sections contain a brief overview of the functions in each group listed above; for complete details about these and other functions see the *Functions Reference*.



Other useful functions include those used in subtotals and data mining. For descriptions of subtotal functions, see [Standard subtotal functions, page 103](#). Refer to [Chapter 13, Data Mining Services](#) for information on data mining and predictive metrics.

Rank

In **rank** functions, you specify the metric to be ranked and whether to rank in ascending or descending order. You can also specify whether to break by an attribute.

The level of the rank depends on the level in the report. For example, the following metric, named Revenue Rank, combined with customer on a report shows the highest revenue customer as number one.

Rank (Revenue)

Adding a parameter to break by year displays the customer generating the highest revenue in each year as number one. The metric definition for this Revenue Rank by Year metric is shown below:

Rank<BreakBy={Year}>(Revenue)

The document shown below uses two different reports:

- The “Top 10” reports are created from the same report, which uses the Revenue Rank by Year metric. Each of the reports on the document is filtered to display a particular year and the top ten Revenue Rank by Year values for that year.
- The “Overall Ranking” report is created from a report that uses the Revenue Rank metric.

Top 10 Customers of 2007					Overall Ranking		
Customer		Year	Metrics	Revenue Rank by Year	Customer		Metrics Revenue Rank
Lysne	Kasey	2007		1	Yadon	Kate	1
Quillman	Julie	2007		2	Talman	Holly	2
Hand	Earnest	2007		3	Shaffer	Carman	3
Smothers	Gabriel	2007		4	Watson	Nick	4
Isaac	Frank	2007		5	Boudreau	Ricky	5
Moran	Horace	2007		5	Combs	Lorri	5
Tysor	Carlos	2007		5	Penrose	Micah	5
Umbras	Theda	2007		5	Seay	Penny	8
Hider	Kyong-Ok	2007		9	Olds	Carolyn	9
Milsted	Lella	2007		10	Carpenter	Dona	10
Olderman	Leslie	2007		10	Epperson	Tracey	10
Roland	Kleth	2007		10	Pascal	Garland	10
Snee	Lynda	2007		10	Braley	Marjoria	13
					Porges	Wendell	14
Top 10 Customers of 2008					Hayworth	Rona	15
Customer		Year	Metrics	Revenue Rank by Year	Badour	Kamil	16
Pietrusiewicz	Odell	2008		1	Self	Arden	17
Corsi	Vivkie	2008		2	Downer	Oriana	18
Arruda	Girard	2008		3	Austin	Melissa	19
Rolfson	Terence	2008		3	Laurie	Chad	20
Starkel	Moise	2008		3	Leahy	Clovel	21
Daly	Patrice	2008		6	Pulido	Bertrand	22
Crow	Alexander	2008		7	Wright	Arnie	22
Utep	Daphne	2008		7	Victor	Fran	24
Gronfein	Dome	2008		9	Richards	Ula	25
Nelson	Dorward	2008		10	Martens	Bret	26
					Fukushima	Kein	27
					Madany	Douglas	28
					Alken	Tarek	29
					Durkin	Garth	29
					Judkins	Janie	31



The image above shows a MicroStrategy document. Documents can display different reports within the same display, as shown in this example. For more information on using documents in general, see the *Document and Dashboard Analysis Guide*; for steps to create documents, see the *Report Services Document Creation Guide*.

Count

The **count** function is usually used with an attribute, although a fact can also be counted. Metrics that use the Count function in their definition are commonly referred to as *count metrics*.

By default, the Count function counts the total number of attribute elements available in the data warehouse. However, this may not always be useful for analysis, because the function will return the same value regardless of the objects you place on a report's template.

For example, you wish to find out how many customers made purchases in each category. You can use the Count of Customers metric, which uses the formula `Count (Customer)`. The formula counts the number of elements in the Customer lookup table, and will always have the value 10,000, as in the report shown below:

	Metrics	Count of Customers
Category		
Books		10,000
Electronics		10,000
Movies		10,000
Music		10,000

To use the Count function for such an analysis, you must change the function parameters to count the attribute elements based on a fact, rather than the lookup table.

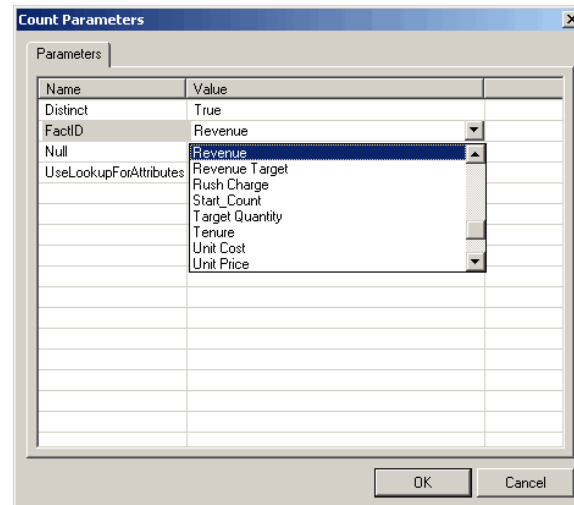
Change the function parameters for the Count function

- 1 Open the metric in the Metric Editor.
- 2 In the Metric Definition pane, click the Count function once to highlight it.
- 3 Right-click the highlighted Count function and choose **Count parameters**. The Count Parameters dialog opens.
- 4 Use the **Distinct** drop-down list to select whether to count distinct occurrences of the target.

For example, a single customer may have made several orders. If you select **False**, the customer is counted once for each order. If you choose

True, each customer is counted once only, irrespective of how many orders they have made.

- 5 Under the **FactID** drop-down, select a fact you wish to use for the analysis. The fact's source table should contain data for the attribute that is being counted.



- 6 Click **OK**.

- 7 In the Metric Editor, click **Save and Close**.

The metric can now be used for various analyses, such as in the previous example, where you need a report that shows the number of customers that bought products in each category. The image below shows the same report, using the newly defined metric:

Category	Metrics	Count of Customers (by Revenue)
Books		9,940
Electronics		9,804
Movies		9,968
Music		9,966

Running and moving sums and averages

These functions include:

- Moving average

- Moving sum
- Running average
- Running sum

All these functions are similar and are called OLAP functions in the Metric Editor. The running sum function uses a metric as input and calculates a cumulative total of values based on a sort order specified in the metric definition. The sort order is not based on the report level. For example, a report with dates, revenue, and month-to-date revenue is needed. The month-to-date revenue is defined as `RunningSum<BreakBy={ [Month of Year] }, SortBy= (Day) >(Revenue)`. The Break-by value is Month of Year, so that the running sum is restarted for each month. The running sum is sorted by Day, so that the first day of the month is displayed first, then the sum of the first and second day, then the sum of the first three days, and so on. Finally, the definition includes the metric that the running sum uses.

The moving sum/average is basically a running sum/average with specific start and end times.

The following document contains both a running sum and a moving sum, both calculated on the Revenue metric. The running sum begins at the first quarter, and adds every quarter after that. The moving sum calculates a year-to-date revenue amount, resetting at each new year. Therefore, the moving sum restarts for each year. The running sum's definition is `RunningSum<SortBy= (Year, Quarter) >(Revenue)`, while the moving sum is defined as `MovingSum<BreakBy={ Year } >(Revenue, 4.0)`, where 4.0 is the number of quarters to calculate.

Year	Quarter	Revenue	Running Sum of Revenue	Year-To-Date Revenue (Moving Sum)
2007	2007 Q1	\$1,682,656	\$1,682,656	\$1,682,656
	2007 Q2	\$1,985,788	\$3,668,444	\$3,668,444
	2007 Q3	\$2,314,295	\$5,982,738	\$5,982,738
	2007 Q4	\$2,664,500	\$8,647,238	\$8,647,238
2008	2008 Q1	\$2,498,756	\$11,145,994	\$2,498,756
	2008 Q2	\$2,684,764	\$13,830,758	\$5,183,520
	2008 Q3	\$3,067,019	\$16,897,777	\$8,250,539
	2008 Q4	\$3,267,067	\$20,164,844	\$11,517,606
2009	2009 Q1	\$3,111,989	\$23,276,833	\$3,111,989
	2009 Q2	\$3,504,479	\$26,781,312	\$6,616,468
	2009 Q3	\$3,729,456	\$30,510,769	\$10,345,924
	2009 Q4	\$4,512,940	\$35,023,708	\$14,858,864



The image above shows a MicroStrategy document. For more information on working with documents, see the *Document and*

Dashboard Analysis Guide; for steps to create documents, see the *Report Services Document Creation Guide*.

N-tile

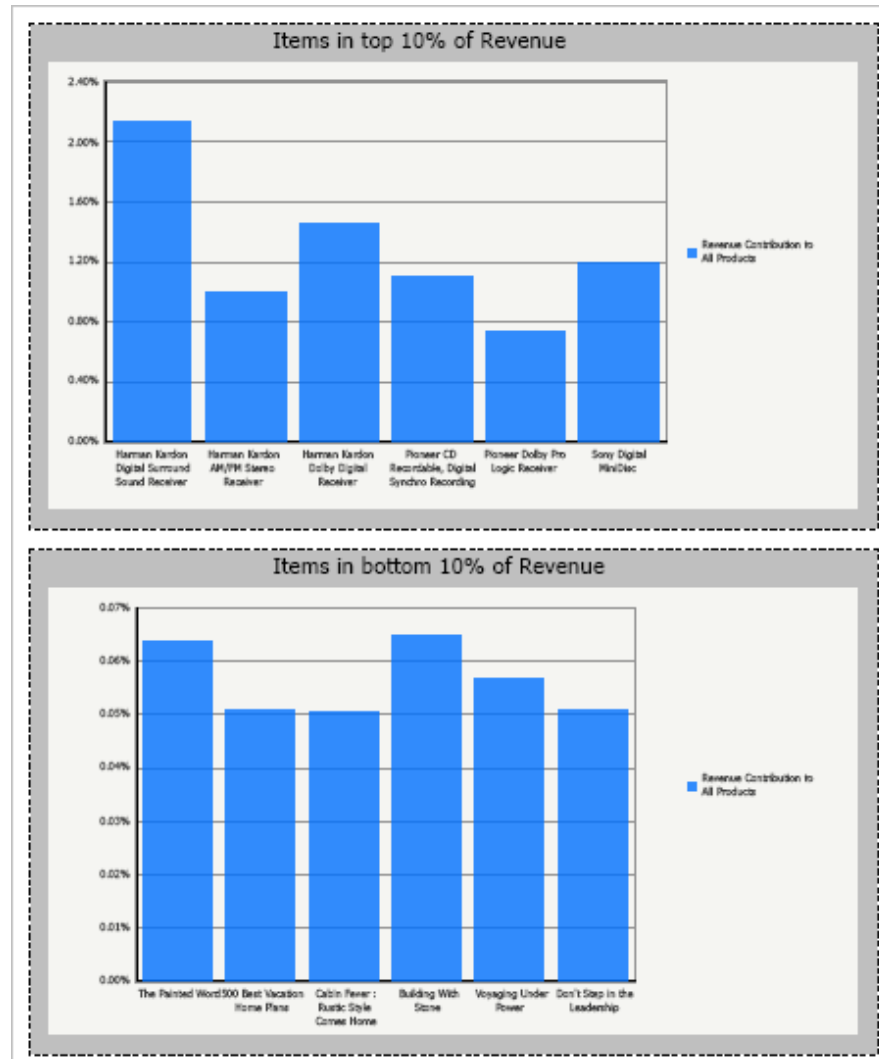
The **N-tile** function, which is also referred to as segmentation, sets up numbers of groups, or tiles, for a metric. Examples of required parameters are the number of groups and whether they should be sorted in ascending or descending order.

An example of an N-tile function in use is displaying what items are in the top 10% of sales, the next 10%, and so on. Use the N-tile function with the Revenue metric. Because the results are in tenths (ten groups of ten each), the number of tiles is ten. To assign the number one to the top 10%, sort the values in descending order. The metric's definition is `NTile<Tiles=10, Ascending=False>(Revenue)`. This decile revenue metric is used on the a report with a contribution and revenue metric. Only eight rows of the report are displayed below:

Item	Metrics	Revenue	Revenue	Revenue
			Contribution to All Products	in Deciles
100 Places to Go While Still Young at Heart		\$67,993	0.19%	3
Art As Experience		\$23,733	0.07%	9
The Painted Word		\$22,323	0.06%	10
Hirschfeld on Line		\$50,442	0.14%	4
Adirondack Style		\$39,101	0.11%	7
Architecture : Form, Space, & Order		\$41,378	0.12%	6
50 Favorite Rooms		\$26,502	0.08%	9
500 Best Vacation Home Plans		\$17,729	0.05%	10

The report is used on the document shown below. Although the Revenue in Deciles metric is not displayed on the document, it is used in the view filters of the two graphs. The first graph displays only those items in the top 10% of revenue (the Revenue in Deciles metric is equal to one), while the second graph displays only those items in the bottom 10% of revenue (the Revenue in Deciles metric is equal to ten). You can see in the bottom graph that two of the items displayed in the report are shown on the graph (The Painted Word

and 500 Best Vacation Home Plans). Only a portion of the document is displayed.



The image above shows a MicroStrategy document. Documents can display a report as a graph, a grid, or as both a grid and a graph in the same display. Documents can also display different parts of the same report in the same display, as shown in this example. For more information on using documents, see the *Report Services Document and Dashboard Analysis Guide*; for steps to create documents, see the *Report Services Document Creation Guide*.

First and Last

The First and Last functions provide the ability to use sort-by inside aggregation functions, which are functions where the value depends on the sort order. First returns the First value in a sorted set of values, while Last returns the last value. You can define the sort attributes in the function parameters.

For example, an inventory report lists the on-hand supply for each day. The report subtotals are the last day's inventory. Creating a user-defined subtotal that uses the Last function provides the needed last day inventory subtotal. If the sort parameters of the function are not set to sort by Day, the function may not provide the correct answer.

For a sample scenario using the First function, see [User-defined subtotal example \(First function\), page 105](#).

Creating your own plug-in functions

The MicroStrategy Function Plug-In Wizard can be used for defining custom functions relevant to your business case scenarios. Intelligence Server makes no distinction between these custom functions and the functions provided by default. These custom plug-in functions are indistinguishable from all other functions or operators, such as Sum, Average, Min, Max, Count, -, +, /, or *. Once a custom plug-in function is created, you can use it in a metric, in a subtotal, as the grand total function, or as a report subtotal.



For more information about subtotals and grand totals, see [Metric subtotals, page 102](#).

Defining custom plug-in functions involves the following steps:

- In the **design** stage, you determine how to implement the analytical procedures into a computer system.
- **Creation** builds the Microsoft Visual C++ project, which is used to produce a library containing your algorithms.
- **Implementation** involves creating the code for the algorithms and compiling this code into a library that will be used by MicroStrategy.
- **Importing** adds the library to a MicroStrategy project so its algorithms are available for use in the project.

- **Deployment** distributes your library to Intelligence Server, which will execute it.
- The final step is **execution**, which is creating new metrics that use the algorithms and using those metric in a MicroStrategy report.

The Function Plug-In Wizard guides you through the creation and implementation steps. It helps you create a Microsoft Visual C++ project with placeholders where you can add custom analytic code. After adding your function-specific C++ code and building your project, you can launch MicroStrategy Developer to import your new function plug-in to be used for all the reports. Deployment occurs on each Intelligence Server system that will use it. The execution step is also performed in MicroStrategy Developer, when you create metrics and reports using the new function. For detailed information on each step, see the Function Plug-In Wizard online help. The *MicroStrategy Developer help* also provides steps to import functions.



The Function Plug-In Wizard must be installed before you can create and implement custom plug-in functions. This installation is incorporated into the standard MicroStrategy installation. The option to install this component is enabled only if Microsoft Visual C++ 6.0 is present on the system where the installation is being performed. During installation, the Function Plug-In Wizard option is located in MicroStrategy Architect under MicroStrategy Developer Products. For installation procedures, see the *Functions Reference*.

Integrating R analytics into MicroStrategy

Customers interested in deploying analytics from the R programming language into MicroStrategy can do so using the R Integration Pack, available separately from the CodePlex open source community web site. For more information, go to <http://www.codeplex.com> and search for the **RIntegrationPack** project (current as of March 1, 2013).

Apply functions: Using database-specific functions in metrics

Apply functions, also called Pass-through functions, provide access to functionality that is not standard in MicroStrategy products but can be obtained through the relational database. When you include an Apply function in an attribute, fact, or transformation expression, the SQL Engine recognizes it as custom SQL and treats it as such. The expression is then sent to the relational database as written. For a full description of apply functions,

including its types, syntax, and examples, see [Apply functions in metrics: Using database-specific functions, page 164](#).

Join specifications

Setting a join specification allows you to place conditions on the data selected for display in a report. You can apply an inner or outer join, which are described in more detail below. In short, an inner join includes only the data common to all the elements in the join, whether that data is tables or metrics. An outer join includes all of the data in all of the elements. You can set joins at the metric and report levels:

- **Join at the metric level:** how the metric is joined to other metrics
- **Join at the report level:** how metrics are joined together in the report; overrides any metric join settings for that report only

Setting the metric join type at the report level (using the **Report Data Options** menu option in the Report Editor) affects only the results for the report being modified. For steps, see the *Analyzing Data* chapter of the *Basic Reporting Guide*.

Setting the metric join type at the metric level affects the results for all reports using that metric. To set the metric join type at the metric level, use the **Metric Join Type** option on the **Tools** menu of the Metric Editor.

By default, a report uses the metric's join type setting. The default metric join type at the metric level uses the default inherited value, which is an inner join.

For compound metrics, you can also set the join type at the **formula** level. This controls how the expressions or metrics within the compound metric are joined. For more information about join types at the formula level, see [Formula join type for compound metrics, page 99](#). For more information about compound metrics in general, see [Compound metrics, page 95](#).

Inner joins versus outer joins

In short, an **inner join** includes only data that is common to all components of the join. An **outer join** includes data that applies to all components. The following descriptions use an example of a report containing the Region attribute, the Sales metric, and the Budget metric:

- By default, **inner joins** are generated for all metrics in a report. The resulting report contains only those rows that have data returned for all the metrics. If a particular region does not have both sales and budget data, the region is not included on the report.
- If you switch to **outer joins** for all the metrics on the report, all rows with data for any of the metrics are displayed. All regions are included in the report, even if they have no data for one of the metrics.

You can specify different joins for each of the metrics on a report. For example, use an outer join for the Sales metric and an inner join for the Budget metric. The resulting report displays all regions with sales data. If a particular region has budget data but not sales data, the region is not included in the report.

For a more detailed explanation of this example, with report samples, see the *Analyzing Data* chapter of the *Basic Reporting Guide*.

Joins between metrics

Setting the metric join type allows you to define the default action for joining the metric to other metrics. This setting is used for the metric on any report the metric is placed on. However, you can override this metric join setting for a particular report. For information on setting metric join types at a report-specific level, see the *Analyzing Data* chapter of the *Basic Reporting Guide*.

The **Metric Join Type** option is accessed from the **Tools** menu of the Metric Editor. The metric join types are as follows:

- **Default** uses the default value.
- **Inner** includes only information contained by all the elements.
- **Outer** keeps all the information in all the elements.



The **Metric Join Type** option is a shortcut to the **Metric Join Type** **VLDB** property located under **Advanced Options** in the same menu.

Metric-specific VLDB properties: SQL and analytical customization

VLDB (Very Large DataBase) properties allow you to customize the SQL that MicroStrategy generates. The metric join type, described in the previous section, is not the only VLDB property that affects metrics. The following list summarizes the metric-specific VLDB properties that can be set at the metric level. For details on each of these properties, see [VLDB properties at the metric level, page 861](#). For details on all VLDB properties available in MicroStrategy, see the *Supplemental Reference for System Administration*.

- **Integer Constant in Metric** determines whether to add a “.0” after an integer.
- **Null Check** indicates how to handle arithmetic operations with null values.
- **Zero Check** indicates how to handle division by zero or when to check for zeros in the denominator during division calculations.
- **Null Checking for Analytical Engine** determines whether a null value is interpreted as zero when the Analytical Engine performs calculations.
- **Subtotal Dimensionality Aware** enables subtotaling based on the dimensionality of a metric.



You can set additional metric VLDB properties at other levels, such as the report and project levels. For more information on how the levels work together, see [Levels of VLDB properties, page 860](#).

Metric column aliases: SQL identifiers and data types

A **column alias** changes the name of a metric in temporary tables, which are used in data marts and SQL generation. This renaming does not affect the actual name of the metric in the metadata but allows you to identify the metric easily, such as in a long SQL report. The column alias is set in the Metric Column Alias dialog box in the Metric Editor. To access it, select **Tools**, then **Advanced Settings**, and then **Metric Column Options**.



For more information on data marts, see [Chapter 12, Accessing Subsets of Data: Data Marts](#).

You can also set the data type and byte length for the metric. The MicroStrategy data types are Big Decimal, Binary, Char, Date, Decimal, Double, Float, Integer, LongVarBin, Long.VarChar, Numeric, Real, Time, Timestamp, Unsigned, VarBin, and VarChar.



There are several drawbacks to using the Big Decimal data type for metric values.

For more information on all data types, see the *Data Types* appendix in the *Project Design Guide*.

ADVANCED FILTERS

Filtering Data on Reports

Introduction

This chapter builds on knowledge provided in the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*. You should already know how to create a simple filter and place it in a report. You should also know the difference between different types of qualifications, although [Types of qualifications, page 127](#), provides a brief description of each type. This chapter discusses the concepts necessary to create and use the following advanced filters and qualifications:

- [Attribute-to-attribute qualification, page 130](#)
- [Dynamic dates: Filtering by offsetting the current date, page 135](#)
- [Importing and exporting elements for attribute qualifications, page 143](#)
- [The attribute level of set qualifications: Output level, page 146](#)
- [Resetting rank and percent metric qualifications: Break by, page 149](#)
- [Metric-to-metric comparisons, page 150](#)
- [Merging attribute qualifications, page 151](#)
- [Applying qualifications independently of relationship filters, page 153](#)
- [Custom expressions and database-specific functions, page 154](#)
- [Joint element lists: Grouping elements from multiple attributes, page 172](#)
- [Prompted filters: Asking for user input for filter conditions, page 174](#)

- [Shortcut vs. embedded filters, page 177](#)



This chapter does not include information about security filters, which are discussed in the *System Administration Guide*, or creating multiple filters at the same time by using a Command Manager script. Command Manager is a MicroStrategy tool designed to automate certain tasks and processes, such as creating, editing, or deleting multiple filters. For more information about Command Manager, see the *Command Manager* chapter of the *System Administration Guide*.

Before you begin

This section reviews basic concepts covered in the *Basic Reporting Guide*. If you need a brief overview of filter basics, this section is designed to help you. If you need a broad refresher on metrics, facts, attributes, prompts, and filters, see the *Basic Reporting Guide*.

In this chapter, the Filter Editor in MicroStrategy Developer is used to work with filters. You can also use MicroStrategy Command Manager to work with filters. Command Manager manipulates MicroStrategy objects by means of user-created scripts rather than a graphical user interface. This can be helpful if you have a large number of filters to create or edit. For more information about Command Manager, see the *Automating Administrative Tasks* chapter in the *System Administration Guide*. For specific Command Manager syntax for working with filters, see the online help for Command Manager.

A filter specifies the conditions that the data must meet to be included in the report results. Using a filter on a report narrows the data to consider only the information that is relevant to answer your business question, since a report queries the database against all the data stored in the data warehouse. Filters are helpful in clarifying large quantities of data and only displaying subsets of that data, so reports show users what they really need to see.

For example, you want to determine the sales for all items in the Book category for the Washington, DC region. Creating a report with the Call Center and Category attributes, as well as the Revenue metric, returns all the revenue for all items in all categories for all call centers. Without a filter, you would have to sift through a lot of report data on your own—the report contains about 4500 rows. By creating a filter that includes the Book category and Washington, DC, and using the filter on the report, the data

displayed when the report is executed is limited—only about 90 rows are returned in the report. A sample follows.

Call Center	Category	Item	Metrics	Revenue
Washington, DC				
	Books			
		100 Places to Go While Still Young at Heart		\$5,874
		Art As Experience		\$2,315
		The Painted Word		\$2,180
		Hirschfeld on Line		\$4,338
		Adirondack Style		\$2,491
		Architecture : Form, Space, & Order		\$3,795
		50 Favorite Rooms		\$2,741

Types of qualifications

A filter is composed of at least one qualification. Qualifications are the actual conditions that must be met, for example, "Region = Northeast" or "Revenue > \$1 million". You can create multiple qualifications for a single filter, and then combine the qualifications using the logical operators AND, AND NOT, OR, and OR NOT.

You can create the following types of filter qualifications:

- **Attribute qualification** allows you to filter on an attribute's form (ID, description, and so on) or on elements of the attribute.
 - An example of an **attribute form qualification** is a condition on the attribute form Customer Last Name for last names that start with the letter H.
 - An **attribute element list qualification** on the Customer attribute consists of a list of customers.
- **Set qualification** allows you to create a set of attributes based on either the metrics associated with those attributes (a **metric qualification**) or the relationships between them (a **relationship qualification**). For example:
 - A metric qualification allows a report to display the sales data for only those products with inventory level below a specified value. It is also referred to as a **metric set qualification**.

- A relationship qualification can be used to create a report that displays all the stores selling Nike shoes in the Washington, DC area. It is also referred to as a **relationship set qualification**.
- **Shortcut-to-a-report qualification**, also known as a report qualification or a report as filter, uses the data from one report as a filter inside another report.
- **Shortcut-to-a-filter qualification**, also known as a filter qualification or embedded filter, uses an existing filter, either as is or with additional conditions, on a report.
- **Advanced qualifications** allow you to create the following:
 - **Custom expressions**, which are customized filter expressions. You can use Apply functions and you can create relationship filters.
 - **Joint element lists**, which join attribute elements and then filter the result sets.



By default, the Advanced Qualifications option is not displayed in the Filter Editor. You must activate it first, using the My Preferences dialog box. For steps, see the online help.

The *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide* provides the fundamentals of creating all these types of filters, except for advanced qualifications.

Advanced filter topics

This chapter builds on the fundamentals provided in the *Basic Reporting Guide* to help you create filters to answer more complex business questions. These advanced topics are listed below, organized by the qualification type used to create them:

- Attribute qualifications:
 - [Attribute-to-attribute qualification, page 130](#), compare two attributes through attribute forms.
 - [Dynamic dates: Filtering by offsetting the current date, page 135](#), are fixed offsets of the current date.
 - [Importing and exporting elements for attribute qualifications, page 143](#), allow lists of data from existing files to be imported into the filter definition.

- Set qualifications:
 - *The attribute level of set qualifications: Output level, page 146*, specifies the attribute level at which the metric or set is calculated for the qualification.
 - *Relating the output level and filter of relationship qualifications, page 148*, specifies the relation between the attributes in the output level and the filter qualification.
 - *Resetting rank and percent metric qualifications: Break by, page 149*, is the attribute level at which to restart counting rank or percent values for a metric.
 - *Metric-to-metric comparisons, page 150*, dynamically compare the values of two metrics.
 - *Merging attribute qualifications, page 151*, specifies whether existing attribute qualifications should be merged into the calculation of the metric qualification.
 - *Applying qualifications independently of relationship filters, page 153*, sets whether the filter criteria are applied to the entire report or to the relationship qualification itself.
- Advanced qualifications using custom expressions and Apply functions:
 - *Custom expression qualification filters, page 156*, allow you to create customized filter expressions. In situations where your report requirements cannot be satisfied via MicroStrategy's built-in functions, you will use Apply functions, as well.
 - *Custom expressions in attribute-to-attribute qualification filters, page 160* allow you to create filters that are based on a relationship between two attributes or on a relationship between an attribute and a custom expression.
 - *Custom expressions in relationship qualification filters, page 161*, allow you to generate a list of attribute elements as a filter based on a specified fact or table in your warehouse (which defines a relationship), a metric qualification or other filter, and a specified list of attributes.
- *Prompted filters: Asking for user input for filter conditions, page 174*, provide dynamic report definitions which can change with every report execution. Most qualification types allow prompts.

Attribute-to-attribute qualification

For some purposes, you may need to filter on an attribute based on criteria related to another attribute or related to a custom expression. For example, perhaps you need to generate a report that lists customers who live in the same city as call centers in your company. In this case, you qualify on the Customer and Call Center attributes using an attribute-to-attribute qualification filter. (See [Example: Customers who live in the same city as call centers, page 131](#) for steps to complete this example.)

In another scenario, your next marketing campaign is targeted towards customers with income in a certain bracket. This requires an attribute-to-attribute qualification filter to qualify on the Income Bracket attribute, but in this case, the second item qualified on is a custom expression instead of an attribute. (See [Example: Filtering on an income bracket, page 133](#) for steps to complete this example.)



Attribute-to-attribute qualification filters always require the use of custom expressions, which are touched upon briefly in this section. For a more comprehensive discussion of custom expressions, Apply functions, and examples, please refer to [Custom expression qualification filters, page 156](#).

Creating attribute-to-attribute qualification filters

Basic steps for creating an attribute-to-attribute qualification filter in MicroStrategy Web are below, followed by specific examples of its use.

To create an attribute-to-attribute qualification

- 1 In MicroStrategy Web, log in to a project.
- 2 Open any folder page (for example, click Shared Folders on the home page).
- 3 Click the **Create Filter** icon . The New Filter page opens.
- 4 From the Object browser on the left, select an attribute and drag it on the filter pane on the right.
- 5 Click **Qualify**.

- 6 From the first drop-down list, select the attribute form to filter report data.
- 7 From the next drop-down list, select an operator such as Equals or Less Than.
- 8 Click the **Select Attribute** icon  to select another attribute for comparison. The Select an Object dialog box opens.

You can either select an attribute or type a custom expression for comparison. For an example of selecting another attribute for comparison, see [Example: Customers who live in the same city as call centers, page 131](#). For an example of typing a custom expression for comparison, see [Example: Filtering on an income bracket, page 133](#).

- 9 Select an attribute qualification and click **OK**.
- 10 Click the **Apply** icon  to apply your filter.
- 11 Save the filter.

Example: Customers who live in the same city as call centers

Your new utility company has call centers located throughout the country, and your recent surveys indicate that customers who live in the same city as a call center are particularly satisfied with service due to extremely rapid repairs during power outages. To begin your new advertising campaign, you want to generate a list of Call Centers that coincide with Customer Cities.

The following steps create an attribute-to-attribute qualification filter that generates the list of desired cities.

To create an attribute-to-attribute qualification that compares the Call Center and Customer City attributes

- 1 In MicroStrategy Web, log in to a project.
- 2 Open any folder page (for example, click Shared Folders on the home page).
- 3 Click the **Create Filter** icon . The New Filter page opens.

- 4 From the Object browser on the left, select the **Customer City** attribute from the Customers hierarchy and drag it to the filter pane on the right.
- 5 Change **Qualify** to **ID**.
- 6 Change the operator to **Equals**.
- 7 Click the **Select Attribute**  icon. The Select an Object dialog box opens.
- 8 Select the **Call Center** attribute from the Geography hierarchy.
- 9 Click **OK**.
- 10 Click the **Apply** icon  to apply the filter.
- 11 Save and close the filter, naming it **Customer City = Call Center**.

Now that you have created the attribute-to-attribute qualification filter, create a report and add the filter.

To create the report

- 1 In MicroStrategy Web, log in to a project.
- 2 Click the MicroStrategy icon  at the upper left of any page and select **Create Report**.
- 3 Add the Customer City attribute to the report.
- 4 Add the **Customer City = Call Center** filter to the report.
- 5 Save the report, and then execute it.

When the report is executed, only a small subset of the total number of Call Center attribute elements are displayed. Each of the elements (cities) present on the report is also a Customer City.

Customer City
Addison
Akron
Albany
Albert City
Alexandria
Allentown
Anderson
Annapolis
Arden
Arlington Heights
Arlington
Artesia
Ashby
Asheville
Ashland
Atlanta
Atlantic City
Auburn

Example: Filtering on an income bracket

Your marketing manager wishes to see profit data on customers who earn less than \$100,000 per year, and your Income Bracket attribute contains the following elements (shown in their DESC form):

DESC
20K and Under
20K-30K
30K-40K
40K-50K
50K-60K
60K-70K
70K-80K
80K-90K
90K-100K
Over 100K

Use the steps below to create a filter for a report that lists all income brackets—except for the highest one, Over 100k—and their associated profits.

To create an attribute-to-attribute qualification filter that uses a custom expression

- 1 In MicroStrategy Web, log in to a project.
- 2 Click the MicroStrategy icon  at the upper left of any page and select **Create Report**.
- 3 From the Object browser on the left, right-click the **Income Bracket** attribute from the Customers hierarchy and click **Add to Grid**. The Income Bracket attribute is added to the report grid.
- 4 Right-click the **Income Bracket** attribute again and click **Add to Filter**. The Income Bracket attribute is added to the Report Filter pane.
- 5 Change **Qualify** to **DESC**.
- 6 Change the operator to **Does not begin with**.
- 7 In the text box, type O.

Note that in the image above, all income brackets except for the Over 100k bracket begin with a character other than “O”, so by excluding attribute elements that begin with that character, you filter out of your report only the highest Income Bracket, Over 100k. All other brackets appear on the report.

- 8 Click the **Apply** icon  to apply the filter.
- 9 Save the report and then execute it.

Executing a report with this filter and placing the Income Bracket attribute and Profit metric on the report displays the following data:

Income Bracket	Profit
20K and Under	\$388,021
20K-30K	\$360,628
30K-40K	\$309,851
40K-50K	\$983,601
50K-60K	\$743,493
60K-70K	\$771,286
70K-80K	\$599,035
80K-90K	\$470,937
90K-100K	\$375,436

Note that the highest income bracket, Over 100k, is excluded, due to the custom expression used in the attribute-to-attribute qualification filter. While this example used an attribute-to-attribute qualification filter, the filter actually only used one attribute, Income Bracket, in its definition. The attribute was coupled with a custom expression to achieve the desired result.

Dynamic dates: Filtering by offsetting the current date

When you qualify on a date attribute form with the date data type, you can select dynamic dates, which are fixed offsets of the current date. They can be either a fixed set of dates or different date ranges that change through time. For example, a dynamic date can be used in a report that examines revenue amounts in the previous two months. This is represented as “today” with an offset of two months.

You can express dynamic date qualifications in several ways, as shown in the following examples:

- An offset of four years, three months, two weeks, and one day from today
- Monday of this week
- Monday of this week with an offset of two days
- The fourth of this month
- The fourth Wednesday of this month
- May fourth of next year
- The third Wednesday in May of this year

While evaluating a dynamic date such as “first of this month minus seven days,” the order in which these two parts are calculated is significant. The addition or subtraction of days, weeks, months, or years (the offset) is always done first, before “first of this month,” “this week,” “this year,” and so on is calculated. For example:

- If today is February 13th, then “today minus seven days” is February sixth, and “the first of the month of today minus seven days” is February first.

- However, if today is February second, then “today minus seven days” is January 26th, and “the first of the month of today minus seven days” is January first.

Dynamic dates are available for any attribute form that has a Date or Datetime format type. For an attribute form with a Datetime format type, dynamic times are also available. Hours, minutes, or both can be calculated dynamically. Hours, minutes, or both can be defined as static. For example, if the current time is 12:40 PM:

- Then “this hour plus 1 hour” and “this minute minus 0 minutes” (dynamic hour and dynamic minute) is calculated as 1:40 PM.
- Then “this hour plus 1 hour” and “Static Minute = 24” (dynamic hour and static minute) is calculated as 1:24 PM.
- Then “Static Hour = 11 AM” and “this minute minus 10” (static hour and dynamic minute) is calculated as 11:30 AM.
- Then “Static Hour = 11 AM” and “Static Minute = 24” (static hour and static minute) is calculated as 11:24 AM.

When you create a dynamic date for an attribute form with a Datetime format, you must also create a dynamic time, as described in the *MicroStrategy Developer help*.

You use the Date and Time Editor to create a dynamic date in an attribute qualification. When you qualify on a date attribute, a calendar icon is displayed; click it to open the Date and Time Editor.

Dynamic date example

The following report uses two filters, each containing a dynamic date qualification. The first filter, applied as the report filter, is called Today and qualifies on the current date. The second filter, applied as a condition on a metric condition, is called Today - 2 months. It qualifies on the current date minus an offset of two months. The Revenue metric calculates revenue for the current date because the report filter (the Today filter) is applied. The Revenue (2 months ago) metric calculates revenue for today’s date minus two months because the metric contains a condition (the Today - 2 months filter).

The following report sample was executed on 9/26/09. The Revenue metric returns revenue for 9/26/09, and the Revenue (2 months ago) metric

calculates revenue for 7/26/09. This report allows you to compare the revenue values quickly and easily.

Region	Metrics	Revenue	
		Revenue	Revenue (2 months ago)
Central		\$4,596	\$5,099
Mid-Atlantic		\$10,828	\$6,738
Northeast		\$12,307	\$5,226
Northwest		\$631	\$1,704
South		\$8,430	\$9,455
Southeast		\$3,738	\$710
Southwest		\$5,637	\$1,615
Web		\$6,138	\$6,744

To create this report, follow the procedure below.



Note the following:

- This procedure assumes that you are familiar with creating a conditional metric. For information, see [Conditional metrics, page 63](#).
- MicroStrategy Tutorial may not contain information for the current date; if the report does not return any data, adjust the dates accordingly. For example, you could create qualifications for six months ago and for 8 months ago. If you are working with your own data, rather than Tutorial data, the current date calculation works if the project contains data for the current date.

To create an attribute qualification using a dynamic date

Create the Today filter

This filter, which will be used on the report, qualifies on the current date.

- 1 In MicroStrategy Developer, on the **File** menu, point to **New**, and choose **Filter**. The Filter Editor opens.

If the New Filter dialog box is displayed, click the **Empty Filter** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Drag the **Day** attribute from the Object Browser to the Filter definition pane. The Attribute Qualification pane opens.
- 3 Select **ID** from the **Qualify On** drop-down list.
- 4 Click the **Date Editor** icon (the calendar). The Date Editor opens.
- 5 Select **Dynamic date**.
- 6 By default, Today is selected. Note that the current date appears in the Preview area.
- 7 Click **OK**, and then **OK** again to return to the Filter Definition pane. Notice that the qualification is displayed as Day(ID) Exactly Today.
- 8 Save and close the filter, naming it **Today**.

Create the Today - 2 months filter

This filter subtracts two months from the current date. It will be used on the conditional metric.

- 9 Begin creating a new filter, as described in the steps above, through selecting Dynamic date.
- 10 By default, Today is selected. In the boxes next to months, select **Minus** and **2**. This indicates an offset of two months ago. Note that the current date and the dynamic date appear in the Preview area.
- 11 Click **OK**, and then **OK** again. Notice that the qualification is displayed as Day (ID) Exactly Today minus 2 months.
- 12 Save and close the filter, naming it **Today - 2 months**.

Apply the Today - 2 months to a metric

Create a conditional metric to calculate the revenue on the dynamic date.

- 13 On the **File** menu, point to **New**, and then select **Metric**. The Metric Editor opens.

If the New Metric dialog box is displayed, click the **Empty Metric** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 14 Drag the **Revenue fact** from the Object Browser to the Metric Definition pane.
- 15 Select **Condition** in the Metric component pane. The Condition definition pane opens.
- 16 Drag the **Today - 2 months** filter from the Object Browser to the Condition definition pane.
- 17 Save and close the metric, naming it **Revenue (2 months ago)**.

Create the report

- 18 On the **File** menu, point to **New**, and then select **Report**. The Report Editor opens.

If the New Grid dialog box is displayed, click the **Empty Report** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 19 Add the **Region** attribute, the **Revenue** metric, and the **Revenue (2 months ago)** metric.
- 20 Drag the **Today** filter from the Object Browser to the Report Filter pane.
- 21 Save the report, then execute it.

Dynamic date based on a prompt answer

You can base the dynamic date on a prompt answer, rather than today's date. For example, you want to see daily revenue for the last month, starting at a

particular date. If the specific date is 11/8/09, you want to see the daily revenue from 10/8/09 until 11/8/09, as shown in the report sample below:

Day	Metrics	Revenue
10/8/2009		\$40,434
10/9/2009		\$48,852
10/10/2009		\$52,986
10/11/2009		\$56,835
10/12/2009		\$52,632
10/13/2009		\$31,302
10/14/2009		\$56,277
10/15/2009		\$37,353
10/16/2009		\$45,574
10/17/2009		\$43,407
10/18/2009		\$58,024
10/19/2009		\$51,206
10/20/2009		\$55,250
10/21/2009		\$43,064
10/22/2009		\$40,994
10/23/2009		\$43,385
10/24/2009		\$39,919
10/25/2009		\$60,749
10/26/2009		\$40,673
10/27/2009		\$38,559
10/28/2009		\$41,140
10/29/2009		\$53,214
10/30/2009		\$29,912
10/31/2009		\$62,959
11/1/2009		\$67,248
11/2/2009		\$60,833
11/3/2009		\$33,976
11/4/2009		\$43,429
11/5/2009		\$30,605
11/6/2009		\$48,763
11/7/2009		\$37,927
11/8/2009		\$64,542

To do this, create a prompt for a date. Use that date prompt in a filter that returns the dates between the date selected in the prompt and one month before the prompt answer.



This procedure assumes that you are familiar with creating a prompt. For steps, see the *Basic Reporting Guide* or the *MicroStrategy Developer help*.

To create a dynamic date based on a prompt answer

Create the date prompt

This prompt, which will be used on the filter, asks the user for one date.

- 1 In MicroStrategy Developer, on the **File** menu, point to **New**, and choose **Prompt**. The Prompt Generation Wizard opens.
- 2 On the Welcome page, select **Value prompt**, then click **Next**. The Value Prompt page opens.
- 3 Select **Date and Time**, then click **Next**. The General Information page opens.
- 4 Type a **Title**, such as Target Date.
- 5 Type text in the **Instructions** field, which is displayed when the prompt is run during report execution.
- 6 Select **Prompt answer required**.
- 7 Click **Finish**. The Save As dialog box opens.
- 8 Select the folder in which to save the new prompt, type **Target Date** as the **Object Name**, and click **Save** to return to MicroStrategy Developer.

Create the filter

This filter, which will be used on the report, returns the dates between the date selected in the prompt and one month before the prompt answer.

- 9 On the **File** menu, point to **New**, and then select **Filter**. The Filter Editor opens.

If the New Filter dialog box is displayed, click the **Empty Filter** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 10 Drag the **Day** attribute from the Object Browser to the Filter definition pane. The Attribute Qualification pane opens.
- 11 Select **ID** from the **Qualify On** drop-down list.

- 12 Select **Between** from the **Operator** drop-down list.
- 13 Drag the **Target Date** prompt from the Object Browser to the text box for the first value, as shown below. The cursor will change to a + when it is in the correct position.

The screenshot shows the 'Attribute Qualification' dialog box. It has several sections: 'Choose an attribute' with 'Attribute:' set to 'Day'; 'Qualify On:' set to 'ID'; 'Operator:' set to 'Between'; and two 'Value' fields, both containing '11/15/2010'. Between the two value fields is an 'AND' button. A mouse cursor is hovering over the 'AND' button, and a small '+' icon is visible next to it. On the right side of the dialog are buttons for 'OK', 'Cancel', 'Help', and 'Prompt...'.

When you drop the prompt, the text box changes to Shortcut to: Target Date, and the drop-down list changes to Simple Prompt.

- 14 Select **Custom** from the second drop-down list.
- 15 Type the following in the second text box:
AddMonths(
- 16 Drag the Target Date prompt from the Object Browser to the second text box.
- 17 Type the following at the end of the second text box:
, -1)

The interface should now look like the following:

This screenshot shows the 'Attribute Qualification' dialog box after several modifications. The 'Operator' remains 'Between'. The first 'Value' field now has a dropdown menu set to 'Simple Prom' and the text 'Shortcut to: Target Date'. The second 'Value' field has a dropdown menu set to 'Custom' and the text 'AddMonths(? [Target Date],-1)'. The 'AND' button is still between the two value fields. The 'Prompt...' button on the right is now disabled.

- 18 Click **OK** to return to the Filter Definition pane.

19 Save and close the filter, naming it **1 Month Ago from Prompted Date**.

Create the report

20 On the **File** menu, point to **New**, and choose **Report**. The Report Editor opens.

If the New Grid dialog box is displayed, click the **Empty Report** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

21 Add the **Day** attribute and the **Revenue** metric.

22 Drag the **1 Month Ago from Prompted Date** filter from the Object Browser to the Report Filter pane.

23 Save the report, then execute it.

Importing and exporting elements for attribute qualifications

You can import filter elements into the Filter Editor from sources other than MicroStrategy. The following requirements must be met:

- The filter is an attribute qualification.
- The filter is qualified on an attribute form, not elements.
- The operator must be **In list** or **Not in list**.
- The elements to be imported are stored in an Excel file or a text file.

The import filter elements option adds more flexibility to the Filter Editor by allowing lists of data from pre-existing files to be imported into the filter definition. Existing filter definitions can also be exported to a file.



You can use a prompt to allow you to select the file to import when you run the report.

Importing elements from a text file or a Microsoft Excel file can be quicker and more efficient than selecting each individual element to be included in the filter. For example, you have an Excel spreadsheet that lists the products

on sale this month. You need to review last week's revenue for just these items. Rather than selecting them in the Filter Editor, you can simply import the file. Likewise, you can export existing filter definitions to a file.

Requirements for formatting the import file

The following rules apply to the formatting of files:

- **Excel**—Data can be stored in rows, columns, or both, as follows:
 - If the data in a cell has double quotes in the first and last position, it is imported as it is, with the quotes.
 - If the data in a cell has single quotes in the first and last position, it is imported as is, with the quotes.
 - If the data in a cell does not satisfy conditions 1 or 2, it is checked to see if it is a number. If the data is a number, it is imported as it is.
 - If the data in a cell does not satisfy conditions 1 or 2, it is checked to see if it is a date. If it is a date, it is imported by adding single quotes at the beginning and at the end to comply with the date format.
 - If the data in a cell does not satisfy any of the above conditions, it is considered as text data and is imported by adding double quotes at the beginning and end to comply with the text format.
- **Text**—Data in a text file must be one of the following:
 - Tab-delimited
 - List-delimited as specified in the regional settings
 - Return-delimited

To import filter elements

- 1 In MicroStrategy Developer, select **File**, point to **New**, and then choose **Filter**. The Filter Editor opens.

If the New Filter dialog box is displayed, click the **Empty Filter** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Double-click in the Filter definition pane. The Filtering Options dialog box opens.
- 3 Select **Add an Attribute qualification** and click **OK**. The Attribute Qualification pane opens.
- 4 Select an attribute by dragging the attribute from the Object Browser into the **Attribute** box.
- 5 Select any form except Elements from the **Qualify On** drop-down list.
- 6 Select **In list** or **Not in list** from the **Operator** drop-down list.
- 7 Click **Import** to import the filter elements. The Import Elements from File dialog box opens.
- 8 Select the Excel file or text file that contains the attribute elements.
- 9 Click **OK** to close the Attribute qualification pane.
- 10 Save and close the filter.

To export filter elements

- 1 In MicroStrategy Developer, select **File**, point to **New**, and then choose **Filter**. The Filter Editor opens.

If the New Filter dialog box is displayed, click the **Empty Filter** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Double-click in the Filter definition pane. The Filtering Options dialog box opens.
- 3 Select **Add an Attribute qualification** and click **OK**. The Attribute Qualification pane opens.
- 4 Select an attribute by dragging the attribute from the Object Browser into the **Attribute** box.
- 5 Select any form except Elements from the **Qualify On** drop-down list.
- 6 Select **In list** or **Not in list** from the **Operator** drop-down list.

- 7 In the **List** field, type the element IDs, descriptions, or other items, separated by a comma, as shown in the example below:

Central, Mid-Atlantic, Northeast



You can drag and drop elements from the Object Browser, but you must type a comma between each item. If you drag and drop elements, they are enclosed in double quotes. If you export to an Excel spreadsheet, the quotes are not exported, but if you export to a text file instead, the quotes are exported.

- 8 Click **Export** to export the filter elements. The Export Elements to File dialog box opens.
- 9 Navigate to the directory in which to save the file.
- 10 Type a **File Name**.
- 11 By default, the file is saved as an Excel spreadsheet. To change it to a text file, select **Text Files** from the **Save as type** drop-down list.
- 12 Click **Save** to save the export file and return to the Filter Editor.
- 13 Click **OK** to close the Attribute qualification pane.
- 14 Save and close the filter.

The attribute level of set qualifications: Output level

The output level of a set qualification specifies the attribute level at which the metric or set is calculated for the qualification. For example, if a metric set qualification is Sales > 1000, Sales could mean sales per day, month, category, or region.

You can select the attributes to use as the output level for either a metric qualification or a relationship qualification. For a metric qualification, you can set the output level to the level of the report or the metric instead. For details and examples, see the following sections:

- [The output level of relationship qualifications, page 147](#)
- [The output level of metric qualifications, page 148](#)

The output level of relationship qualifications

The output level of a relationship qualification specifies the attribute level at which the set is calculated for the qualification. It controls the contents of the relationship qualification output. You select the attribute or attributes to use as the output level.

For example, you need a report that lists customers from the same region as a specific customer. This example uses Hugh Abarca as the specific customer. Create a relationship qualification to link the Customer Region and Customer attributes, and then filter that relationship on Hugh Abarca. Since the report needs to filter on Customer Region, the relationship qualification must return that attribute. This is the purpose of the output level, which specifies the attribute level of calculation for the set.

To build this sample report, first create a filter with an attribute qualification for Customer = Hugh Abarca. Then create a second filter with the following qualification:

- **Type:** Relationship
- **Output Level:** Customer Region
- **Filter Qualification:** the filter for Customer = Hugh Abarca
- **Relate Output level and Filter qualification:** Use System Default
- **Advanced Options:** Clear the **Also apply this qualification independently of the relationship filter** check box



Clearing this option applies the filter criteria to the relationship qualification, not to the entire report. This ensures that all customers in the selected Customer Region are listed on the report, not just Abarca. For more information on this option, see [Applying qualifications independently of relationship filters, page 153](#).

Create a report containing Customer Region, Customer, and the Revenue metric. Use the filter containing the relationship qualification as the report filter. A sample of the results is displayed below.

Customer Region Customer		Metrics	Revenue
Central	Abarca	Hugh	\$2,755
	Abern	Brooks	\$1,928
	Abram	Ross	\$1,133
	Adler	Keith	\$2,172
	Alyea-Burkell	Xavier	\$2,062
	Anthony	Nicholas	\$1,541

Relating the output level and filter of relationship qualifications

The **Relate output level and filter qualification by** option sets how the attributes in the output level relate to the existing filter. The relation can be a logical table or a fact. If you do not select a fact or table, and leave the option set to Use System Default, the schema is used to select the appropriate table.

For example, to create a report that shows all the stores selling Nike shoes in the Washington, DC area, you need to set the output level to Stores, the filter qualification to Nike shoes and Region, and the relation to the Sales fact.

The output level of metric qualifications

The output level of a metric qualification specifies the level at which the metric is calculated for the qualification. For example, if a metric qualification is Sales > 1000, Sales could mean sales per day, month, year, store, region, and so on. Creating a metric qualification with an output level of store is equivalent to having a fixed list of stores, if you knew which ones met the metric qualification, in a simple attribute qualification. However, the list of stores in the qualification is generated dynamically.

The output level of a metric qualification can be specified in any of the following ways:

- An **attribute list** allows you to specify the exact set of attributes (such as day, category, or region) to use as the output level.
- **Report level** means that the output level is defined by the level of the report that contains the metric qualification. For example, if the lowest

level of the report is category and the output level is set to report level, the metric is calculated for the category.

- **Metric level** means that the output level is defined by the level, or dimensionality, of the metric itself, regardless of the level of the report.
- The **None selected** option calculates the results at the report level if any of the following is true:
 - The metric is a compound metric.
 - The metric's dimensionality is set to report level.
 - The metric's dimensionality is set to nothing.

Otherwise, the metric's dimensionality is used.

If you do not select an output level for a metric qualification, the **None selected** option is used by default.

Resetting rank and percent metric qualifications: Break by

The break by setting of a metric qualification allows you to choose the attribute level at which to restart counting rank or percent values for a metric. This level must be higher than or equal to the level of aggregation for the metric itself.

For example, given the following data:

Region	Market	Store	Actual Sales \$ (in thousands)
Northeast	Mid-Atlantic	Baltimore	40
Northeast	Mid-Atlantic	Philadelphia	30
Northeast	New England	Boston	20
Northeast	New England	Greenwich	10

The metric is aggregated at the lowest level on the report, which is Store. Therefore, the break by value can be Market or Region, but not Store.

If you break by Market, the ranking counter is reset for each market (in descending order).

Region	Market	Store	Actual (\$K) Sales	Rank
Northeast	Mid-Atlantic	Baltimore	40	1
Northeast	Mid-Atlantic	Philadelphia	30	2
Northeast	New England	Boston	20	1
Northeast	New England	Greenwich	10	2

If you break by Region, the ranking counter is reset for each region. In this example, there is only one region, so the counter is not reset.

Region	Market	Store	Actual (\$K) Sales	Rank
Northeast	Mid-Atlantic	Baltimore	40	1
Northeast	Mid-Atlantic	Philadelphia	30	2
Northeast	New England	Boston	20	3
Northeast	New England	Greenwich	10	4

Metric-to-metric comparisons

Metric-to-metric comparisons allow you to create reports that dynamically compare the values of two metrics. A metric-to-metric comparison is a type of metric qualification. For example, you can create a report that restricts the data to revenue greater than last quarter's revenue.

Create a report that displays the revenue for Call Centers Atlanta, San Diego, and Miami for each quarter during the year 2002. To do this, create one filter that includes the Call Centers Atlanta, San Diego, and Miami and another filter that includes the year 2002. For steps to create a filter, see the *MicroStrategy Developer help*.

When you execute the report, it looks like the following:

Call Center	Quarter	Metrics	Revenue
Atlanta	Q1 02		\$131,604
	Q2 02		\$121,566
	Q3 02		\$120,400
	Q4 02		\$195,356
San Diego	Q1 02		\$288,073
	Q2 02		\$341,784
	Q3 02		\$229,975
	Q4 02		\$395,128
Miami	Q1 02		\$91,743
	Q2 02		\$117,008
	Q3 02		\$86,216
	Q4 02		\$181,105

Now, create a revenue metric that calculates the revenue for the previous quarter and save it as **RevenueLastQuarter** metric. Create a metric-to-metric comparison filter that uses the Revenue metric. Choose the Function as **Metric Value**, and Operator as **Greater than**. Choose the Value as **Metric** and browse to select the newly created metric **RevenueLastQuarter**. Save the filter as **LastQuarter**. The report, when re-executed with the **LastQuarter** filter, now looks like the following:

Call Center	Quarter	Metrics	Revenue
Atlanta	Q4 02		\$195,356
San Diego	Q2 02		\$341,784
	Q4 02		\$395,128
Miami	Q2 02		\$117,008
	Q4 02		\$181,105

Note that only those revenues whose values are greater than the revenues of the previous quarter are displayed on the report.

Merging attribute qualifications

You can specify whether existing attribute qualifications are merged into the calculation of the metric qualification by setting the advanced options for a metric qualification or a shortcut-to-a-filter qualification. The full name of this option is **When resolving this qualification in a report, other qualifications will be taken into consideration and merged**, and it is located on the Advanced Options dialog box. By default, this option is selected, which combines the qualifications.

A metric qualification is contained in a separate pass of SQL, creating a temporary table. If the qualifications are merged, attribute qualifications are added to that pass of SQL. If they are not merged, the attribute qualifications are not included in the metric qualification. They instead appear in the main SQL pass.



For more information on how metric qualifications work, see [About metric qualifications, page 280](#).

For example, a report shows revenue by region. The report filter contains the attribute qualification of Category equal to Electronics and the metric qualification of revenue over \$2 million. If the default is kept, the qualifications are merged. Only revenue from Electronics is considered when the metric checks for revenue over \$3 million. The report results include four regions, as shown below:

Region	Metrics	Revenue
Central		\$3,506,062
Mid-Atlantic		\$3,106,940
Northeast		\$5,962,709
South		\$3,741,753

In contrast, if the qualifications are not merged, revenue is calculated for all categories before the metric qualification is evaluated. However, only revenue from Electronics is displayed on the report. As shown in the following sample, regions are included that do not have at least \$3 million of revenue in the Electronics category, but do have \$3 million of revenue across all categories. The report, as shown below, now includes the Southwest and Web regions.

Region	Metrics	Revenue
Central		\$3,506,062
Mid-Atlantic		\$3,106,940
Northeast		\$5,962,709
South		\$3,741,753
Southwest		\$2,562,060
Web		\$2,724,922

Besides affecting the report results, merging the qualifications reduces the amount of data a calculation must process.

Applying qualifications independently of relationship filters

A relationship qualification creates a link between two attributes and then filters that relationship. The filter used in the relationship qualification is called the filter qualification. You can specify whether this filter qualification is applied to the report or only to the relationship qualification. By default, the **Also apply this qualification independently of the relationship filter** check box is selected, which applies the filter qualification to the whole report. This check box is located on the Advanced Options dialog box.

If the option is cleared, the filter qualification is applied to the relationship qualification itself and not to the entire report. In other words, the filter qualification is only included in the SQL statement that resolves the relationship qualification (subquery) and not the outer query that resolves the report as a whole.

For example, a report must list customers from the same region as a specific customer, Hugh Abarca. Begin by creating a filter with an attribute qualification for Customer = Hugh Abarca.

Next, create a second filter with a relationship qualification that links the Customer Region and Customer attributes. Set the filter qualification, which filters the relationship, to the Customer = Hugh Abarca filter you just created. Set the output level to Customer Region.



For steps to create this relationship qualification, including explanations of the various options, see [The output level of relationship qualifications, page 147](#).

By default, the **Also apply this qualification independently of the relationship filter** check box is selected, which applies the filter qualification to the whole report. Name this filter **Customer Region of Hugh Abarca**.

Create a report containing Customer Region, Customer, and the Revenue metric. Use the Customer Region of Hugh Abarca filter as the report filter. The results are displayed below.

		Metrics	Revenue
Customer Region	Customer		
Central	Abarca Hugh		\$2,755

The only row is Hugh Abarca; no other customers in the Central Customer Region are included. The filter qualification of Customer = Hugh Abarca has been applied to the entire report.

Next, edit the Customer Region of Hugh Abarca filter, clearing the **Also apply this qualification independently of the relationship filter** check box. When you re-execute the report, 2346 rows are displayed—all the customers in the Central Customer Region. A sample of the results is displayed below.

Customer Region Customer		Metrics	Revenue
Central	Abarca	Hugh	\$2,755
	Abern	Brooks	\$1,928
	Abram	Ross	\$1,133
	Adler	Keith	\$2,172
	Alyea-Burkell	Xavier	\$2,062
	Anthony	Nicholas	\$1,541

Clearing the option applies the filter qualification to the relationship qualification, not to the entire report. This ensures that all customers in the selected Customer Region are listed on the report, not just Abarca.

Custom expressions and database-specific functions

This section is divided into two subsections—one on custom expressions, and one on database-specific functions, also referred to as Apply functions. Custom expressions provide a degree of specificity in filters not possible within MicroStrategy's graphical interface, while Apply functions extend this functionality even further by leveraging the computational power of your data warehouse.

Custom expressions

Custom expressions allow you to create highly customized filter expressions. By typing syntax instead of using point-and-click functionality within MicroStrategy, you are able to dictate filter criteria that is more specific and sophisticated than the MicroStrategy interface allows. For an introduction to custom expressions, see [Basics of custom expressions, page 155](#).

You can use custom expressions in the following situations:

- Custom expression qualifications, which evaluate the custom expression you enter. For more information and examples, see [Custom expression qualification filters, page 156](#) in this chapter.
- Attribute-to-attribute qualifications, which compare an attribute and a custom expression. For more information and examples, see [Custom expressions in attribute-to-attribute qualification filters, page 160](#) in this chapter.
- Relationship (metric) qualifications, which compare metric results to a selected value (in this case a custom expression). For more information and an example, see [Custom expressions in relationship qualification filters, page 161](#). For more information on metric qualifications in general, see the *Basic Reporting Guide*.

Basics of custom expressions

Custom expressions are written so that data can be tested against them, and whatever data qualifies as true when it is inserted into the custom expression is returned in the data result set.

All custom expressions must evaluate as either true or false. For example, if you enter $5 = 6$ as the custom expression, the expression never evaluates as true, and a filter based on this custom expression excludes all data from a report that uses it. On the other hand, a custom expression consisting of $5 = 5$ always evaluates as true, so the filter includes everything. No data is filtered out of a report containing this filter.

As an example of a scenario that uses a custom expression, you notice that your mailing campaigns have been impaired because longer last names do not print correctly on the envelopes you have been using. Therefore, your marketing manager needs a list of customers with last names greater than 15 characters long. This report requires a filter that uses a custom expression and a string function to count the number of characters in a name. (See [Example: Test the length of last names, page 157](#) for details.)

In another scenario, your shipping company is trying to save freight costs by optimizing shipments, and it offers you a discount for all freight that you ship between 4 A.M. and 6:59 A.M. To determine if you can save substantially on shipping by taking advantage of this offer, you generate a report showing orders that shipped in the three-hour period specified. This report uses a filter that employs a custom expression and a date-manipulation function to

test for orders that are shipped between 4 A.M. and 6:59 A.M. (See [Example: Test the shipping hour of orders, page 158](#) for details.)

Anything that you can specify in a filter can be specified using a custom expression, though if you can accomplish your goal using the standard point-and-click functionality of MicroStrategy, that will usually be the most efficient method to create your filter. In other words, when point-and-click functionality is not enough for your purpose, use a custom expression to create your filter.

There are certain reports that require a filter with capabilities outside of what MicroStrategy currently offers. These reports, too, use filters with custom expressions, but instead of using built-in MicroStrategy functions, these filters use the functionality of your particular Relational Database Management System (RDBMS). Further details on this functionality are available in [Apply functions in metrics: Using database-specific functions, page 164](#).

Custom expression qualification filters

Custom expression qualification filters are quite flexible and allow you to create highly customized filters that can test data based on text content, very specific date specifications, or other criteria. These tests are based on the structure of a custom expression qualification, since the two sides of the expression are separated by a comparison operator such as =, <, >, like, and so on.

A custom expression qualification filter is created in the Advanced Qualification pane of the Filter Editor. Two examples of custom expression qualification filters follow the steps below.

To create a custom expression qualification

- 1 In MicroStrategy Developer, point to **New** on the **File** menu, and then select **Filter**. The Filter Editor opens.

If the New Filter dialog box is displayed before the Filter Editor opens, click the **Empty Filter** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Double-click in the Filter definition pane. The Filtering Options dialog box opens.
- 3 Select **Add an Advanced qualification** and click **OK**. The Advanced Qualification pane opens.
- 4 From the **Option** drop-down list, select **Custom expression**.
- 5 Type a custom expression in the **Custom expression** box. (For general information on custom expressions, see [Basics of custom expressions, page 155](#). Examples of filters that use custom expression follow these steps.)
 -  You can also enter the custom expression by doing either of the following:
 - Drag objects from the Object Browser into the **Custom expression** box. Select the operators you want to use from the toolbar above the Custom expression box, or simply type them.
 - Use the Insert Function Wizard to guide you through the process of creating an expression that uses a function, by clicking **f(x)** on the toolbar above the Custom expression box. (For details, click the **Help** menu.)
- 6 Click **Validate** to validate the expression. This helps ensure you spelled and punctuated the expression correctly.
- 7 Click **OK** to close the Advanced Qualification pane.
- 8 Click **Save and Close** to save the filter and close the Filter Editor.

Example: Test the length of last names

In this example, your marketing manager needs a list of people with last names greater than 15 characters long. She needs this type of list because customers with such names are not receiving your advertising brochures due to misprinting on the envelopes that you have been using.

To produce the list of names, within MicroStrategy Tutorial, create a custom expression qualification filter using the steps in [Custom expression qualification filters, page 156](#). In the **Custom expression** box, type the following custom expression:

```
Length(Customer@[Last Name]) > 15
```

The string-manipulation function `Length` measures the `Last Name` form of the `Customer` attribute elements in your data warehouse. If the length of a particular `Customer` attribute element is greater than 15 characters long, the custom expression statement evaluates as true, and the `Customer` attribute element is included in your data. If the length of the attribute element is 15 characters or less, however, the statement evaluates as false, and that element is filtered out of the report data.

A report that uses a filter based on the custom expression above appears below.

Customer
Fournier-Roberts
Iannuccillo-Owenby
Ligocki-Blackwell
Lorenzana-Romero
Lorenzana-Romero
Lorenzana-Romero
Narvaez-Sanderson

The `Last Name` form is displayed on the report above. Note that in each entry, the `Last Name` is greater than 15 characters long.



Be sure to specify the attribute form that you are testing. Moreover, if the attribute form is composed of two or more words—as in the above example, where `Last Name` was used—enclose the attribute form in brackets or your custom expression will not be validated.

Example: Test the shipping hour of orders

In this scenario, the shipping company you use is trying to reduce costs by shipping more uniformly throughout the day, and it offers your company a new plan that includes discounts for any packages picked up from your facility between 4 A.M. and 6:59 A.M. To predict whether or not your company would benefit from switching to this plan, your shipping manager requires a list of orders shipped in the three-hour time period specified.

To produce this report, create a custom expression qualification filter using the steps in [Custom expression qualification filters, page 156](#) to test the hour

of the Ship Date/Time attribute in your data warehouse. Notice the syntax required for the custom expression.

Advanced Qualification

Choose an option

Option: Custom expression

Custom expression

Please enter a filter expression or begin by dragging an object from the object browser.

() + - * / f(x) Clear Validate

(Hour([Ship Date/Time]@ID)>3) AND (Hour([Ship Date/Time]@ID)<7)

Valid expression

OK Cancel Help

The expression is actually composed of two parts joined by a logical operator, AND. The first part tests whether the Hour component of the Ship Date/Time attribute element being tested is greater than 3. In other words, the first part checks to see whether the item is shipped at 4 A.M. or later. Similarly, the second part of the custom expression tests whether the Hour component of the Ship Date/Time attribute element tested is less than 7 (the time is no later than 6:59:59 A.M.) Since the two parts are joined by a logical AND, the custom expression tests whether the Ship Date/Time attribute element is between 4:00 A.M. and 6:59:59 A.M.

Executing a report using a filter based on the custom expression above yields results similar to those below.

Order	Ship Date/Time
10000	1/5/2004 4:35:00 am
10001	1/4/2004 4:27:00 am
10002	1/4/2004 5:32:00 am

Notice that every Order element listed shipped between 4 A.M. and 6:59 A.M.



This report cannot be created within the MicroStrategy Tutorial because a) the Ship Date/Time attribute does not exist in that project; and b) all attribute elements in Tutorial that are formatted with date information are in Date format instead of Datetime format, and the Hour function does not yield expected results when it uses an argument in Date format since no time information is stored with the date.

Custom expressions in attribute-to-attribute qualification filters

Attribute-to-attribute qualification filters use one or more criteria that relate an attribute to a custom expression. Often, with such a filter you first select one attribute and then drag a second attribute into a custom expression box. This causes MicroStrategy to generate a custom expression which is very simple—for instance, `ShipDate@ID`—so that it is not immediately obvious that you are using a custom expression. For an example of this type of attribute-to-attribute qualification filter, see [Example: Customers who live in the same city as call centers, page 131](#). For more general information on these filters, as well as additional examples of their use, see [Attribute-to-attribute qualification, page 130](#).

This section deals with attribute-to-attribute qualification filters where the custom expression is authored by the user rather than generated through a simple drag-and-drop action using the MicroStrategy interface. When you type a custom expression instead of dragging an object into the custom expression field, you have a greater degree of control over your filter and you can create more exact and sophisticated filtering criteria.

For example, by comparing the order date and ship date attributes, you can create a report that displays the orders that were shipped 28 days or later than the order date. For a report sample and steps to create this filter, see [Creating an attribute-to-attribute qualification filter, page 160](#).

Creating an attribute-to-attribute qualification filter

The procedure below creates a filter within a report; it does not create a stand-alone filter that can be used on other reports, metrics, and so on.

To create an attribute-to-attribute qualification filter

- 1 From within a blank report in MicroStrategy Tutorial, double-click inside the **Report Filter** pane to create a filter.
- 2 Select **Add an Attribute qualification** and click **OK**. The Attribute Qualification pane opens.
- 3 Find the attribute **Ship Date** in the Object Browser (in the Customer hierarchy) and drag it to the Attribute field.
- 4 Change Qualify on to **ID**.

5 Change the Operator to **Greater than**.

6 Select **Custom** and enter the following:

`(Day@ID + 27)`

This adds 27 days to the Day attribute (which is the order date). The Ship Date is compared to this value.

7 Click **OK** to return to the report.

A portion of the executed report is shown below. Notice that the only data returned in this report is that which satisfies the condition that Ship Date is 28 days or later than Day (the order date).

Order	Day	Ship Date	Metrics	Revenue	Cost	Profit
39025	12/31/2002	1/28/2003		\$ 10	\$ 10	\$0
39030	12/6/2002	1/3/2003		\$ 14	\$ 14	\$0
39054	12/30/2002	1/27/2003		\$ 190	\$ 139	\$51
39112	12/27/2002	1/24/2003		\$ 15	\$ 13	\$2
39132	12/30/2002	1/27/2003		\$ 13	\$ 10	\$3
39143	12/9/2002	1/6/2003		\$ 74	\$ 56	\$18
39236	12/31/2002	1/28/2003		\$ 13	\$ 10	\$3
39248	12/29/2002	1/26/2003		\$ 38	\$ 29	\$9
39282	12/28/2002	1/25/2003		\$ 130	\$ 94	\$36
39320	12/12/2002	1/9/2003		\$ 199	\$ 148	\$51

Custom expressions in relationship qualification filters

Attributes can be related through lookup, relationship, and fact tables. Creating a report that includes attributes that are not related in any of these ways, however, usually results in a very large result set—known as a Cartesian product—that is not useful since the rows comprising the report are simply a list of every possible combination of the attributes.

There are circumstances where you require a report that consists only of attributes, and no metrics. Even though a fact table exists in your data warehouse that relates the attributes on your report, your attributes are effectively not related by it since no metrics are present on your report. Relationship qualification filters allow you to create meaningful reports in such situations. In fact, relationship qualification filters allow you to create meaningful reports in cases where the attributes on your report are not explicitly related through lookup, relationship, or fact tables—where no lookup table, relationship table, or metric exists to relate your attributes.

Another use of relationship qualification filters is to specify the exact lookup, relationship, or fact table that you want to use to relate the attributes on your report. Choosing the right lookup, relationship, or fact table can impact query performance and report execution time, among other things.



Generally, you must know the structure of your data warehouse to use relationship qualifications.

Relationship qualification filters are created using either the Set Qualification or the Advanced Qualification option in the Filter Editor. Choosing the Set Qualification option provides a wizard to guide you through this process, while the Advanced Qualification option allows you to enter custom expressions. For more information on using the wizard, see the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*.

Syntax of custom expressions in relationship qualification filters

The following syntax is used to create a relationship qualification filter using a custom expression:

```
<relation; filter qualification>
{list of output attributes}
```

where:

- The *relation* can be a fact, a table, or an empty filter. The fact or table chosen provides a means for MicroStrategy to relate either two attributes which are not in the same hierarchy, or an attribute and a metric. You must know the structure of your data warehouse to choose an appropriate fact or table through which MicroStrategy's engine can provide the necessary database joins to relate either the two attributes or the attribute and metric that you choose.



If a relationship is left empty, your project's schema is used to select the appropriate table.

- The *filter qualification* specifies the criteria that you use for your relationship qualification filter. You may use any of the following types of qualifications:
 - Attribute qualification
 - Filter qualification (the name of a pre-existing filter)
 - Metric qualification

- The *list of output attributes* is a comma-separated list of the attributes whose elements are filtered for your report based on the filter qualification specified (immediately beforehand in the syntax). In other words, the *filter qualification*—explained in the previous bullet—limits the set of attribute elements contained in your report, and the resultant set of attribute elements is what actually constitutes your filter.



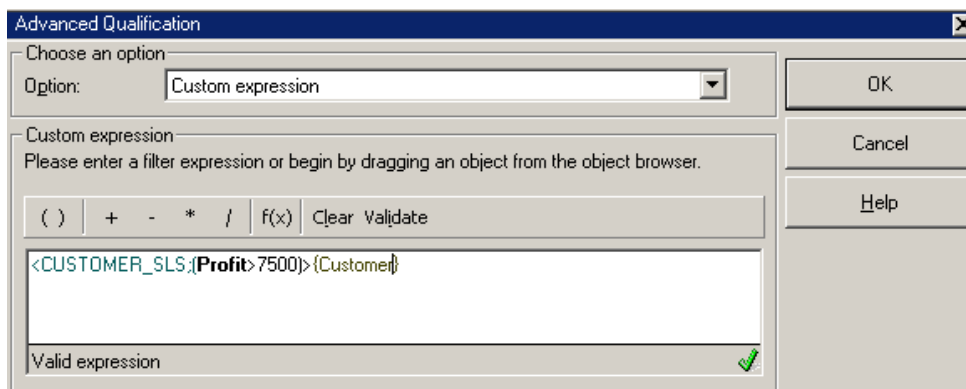
Note the following:

- If your regional settings are not set to English, the list separator must be whatever is defined in your regional settings.
- To specify output attributes, it is easiest to simply drag an attribute from the Object Browser into the **Custom expression** box. If you manually enter the attribute, it must be in the format `[attributename]@ID` or `[attributename]@DESC`.

Example: Customers who have generated more than \$7500 in profit

You require a report that lists the customers who have each generated more than \$7500 in profits. While you can accomplish this with the Set Qualification option, you can also do this quite easily selecting the **Advanced Qualification** option and then, from the **Option** drop-down list, selecting **Custom expression**.

Type the syntax required in the box, click **Validate**, and then click **OK**. The image below shows the syntax required:



The syntax is broken down into its components:

- To relate the Profit metric to the Customer attribute, the custom expression specifies the CUSTOMER_SLS table in your data warehouse as the relation between them. This table is chosen because it contains

columns for the Customer attribute and the Profit fact, effectively relating the Customer attribute to the Profit metric, since the Profit metric is an aggregation over individual Profit facts.

- The filter qualification is specified by a metric qualification: `Profit > 7500`, which allows only data for which Profit exceeds \$7500.
- The only variable left to choose in the syntax is the attribute or attributes that constitute the filter, based on the criterion specified—`Profit > 7500`—and the relationship table chosen, `CUSTOMER_SLS`. Since you require a list of customers that satisfies your profit condition, you choose `Customer` in the custom expression as the output attribute. The list of Customer elements generated from this entire process will be your filter.

Using this filter and the Profit metric on a report results in the data below:

Customer		Metrics	Profit
Westgate	Bris		\$8,450
Whittaker	Michele		\$8,039
Bruesch	Nadim		\$8,030
Andrade	Guynelle		\$7,847
Rainbolt	Nadine		\$7,841
Whiting	Sandor		\$7,818
Marrero	Ferdinand		\$7,747
Mattingley	Elaina		\$7,732
Bartholomay	Dewit		\$7,696
Orsini	Tara		\$7,556

Note that every customer listed on this report has generated a profit greater than \$7500.

Apply functions in metrics: Using database-specific functions

Custom expressions sometimes use Apply functions, also known as Pass-through functions. Using the full power of custom expressions, therefore, may require an understanding of Apply functions. In MicroStrategy, these functions provide access to functionality and syntactic constructs that are not standard in MicroStrategy, but are found in various RDBMS platforms.



It is not advisable to use Apply functions when functions within the MicroStrategy product can accomplish your goal, because using Apply functions bypasses the validations that MicroStrategy provides. Apply functions should only be used when you require functionality that MicroStrategy does not currently provide.

Apply function types

There are several types of Apply functions, each of which corresponds to a basic type of MicroStrategy function and replaces that function when it is necessary to use a corresponding RDBMS function, instead. (See the *Standard Functions* chapter of the *Functions Reference* for further details about when it is necessary and appropriate to use MicroStrategy Apply functions instead of their simpler MicroStrategy function counterparts.) Apply functions act as containers that hand off data from MicroStrategy to your RDBMS. The Apply functions also hand off to your RDBMS functions that are specific to your database. MicroStrategy provides Apply functions as a means of handing off these data and RDBMS-specific functions to your database, which results in your database—not MicroStrategy—performing the computations you require.

The Apply function types are listed below, along with examples of the MicroStrategy functions that they can replace:

- **ApplySimple:** Simple single-value functions, including arithmetic operators such as + and -, date and string manipulation functions, and so on
- **ApplyAgg:** Group-value (otherwise known as aggregate) functions such as Sum and Max
- **ApplyOLAP:** OLAP functions such as Rank and NTile
- **ApplyComparison:** Comparison operators such as >, <, and Like
- **ApplyLogical:** Logical operators such as And and Or

For filters, the ApplyComparison and ApplySimple functions are most useful, although you can also use Apply functions to create metrics and prompts. For examples of Apply functions in filters, see the rest of this section.



While an Apply function can be used wherever the function group it belongs to is applied, you should not use any Apply functions when standard MicroStrategy functions can be used to achieve the goal. This is because using Apply functions effectively bypasses the validations and other benefits of the product. Therefore, use Apply functions ONLY when support does not exist in the MicroStrategy product for your needs, and please submit an enhancement request so that MicroStrategy can evaluate integrating the functionality you are using in a future product release.

Apply functions: syntax

This section provides a brief overview of the syntax for Apply functions. For more comprehensive information, consult the *Standard Functions* chapter of the *Functions Reference*.

Basic syntax of Apply functions

The syntax for Apply functions is as follows:

```
ApplyFUNNAME("expression_with_
  placeholders", argument_0, ...,
  argument_n)
```

The placeholders are represented by #0, #1, and so on. The character # is a reserved character for MicroStrategy and precedes the number of the argument, which starts at 0 and increases in increments of 1. For example:

```
ApplyComparison("<ComparisonFunction(#0,#1)>",
  attribute0@ID, attribute1@ID)
```

Or

```
ApplyComparison("#0 <ComparisonOperator>
  #1", attribute0@ID, attribute1@ID)
```

In these examples, the code marked with angled brackets (<>) can be replaced with database-specific functions or operators such as *between*, *+*, and *DateAdd*. See [Custom expression qualification filters, page 156](#) and [Custom expressions in attribute-to-attribute qualification filters, page 160](#) for examples using these RDBMS functions. Argument placeholders must be placed in the correct positions to match the syntax of the database-specific functions.

Argument types in Apply functions

The number of allowable arguments in an Apply function depends on which particular Apply function you are using, among other considerations. In some instances, the number of arguments varies from case to case within the same Apply function. This is common among the logical Apply functions, in particular.

MicroStrategy does not verify arguments until the parameter markers (for instance #0, #1, and so forth) are replaced at parsing, which occurs when you click **OK** or **Validate** in an expression editor. At parsing time, the engine searches for acceptable argument types for the specific Apply function used

in your custom expression. An acceptable argument type is a `MicroStrategy` object type or an argument that contains the name of a `MicroStrategy` object.

`MicroStrategy` does not check whether the arguments used in the `Apply` functions are correct data types for your specific database functions. `MicroStrategy` does, however, verify that any objects included as arguments actually exist in `MicroStrategy`.

Below are examples of `Apply` functions. More examples can be found in the *Standard Functions* chapter of the *Functions Reference*.

Example: Qualifying on case-sensitive elements

Your company name has been entered into your data warehouse without regard to the correct case. For example, `MicroStrategy` has been entered as `MicroStrategy, Inc.`; `MicroStrategy`; `Microstrategy`; `MICROSTRATEGY`; and `micro strategy`. Rather than creating filters for all the possible combinations, you can create one custom expression qualification filter using the `ApplyComparison` function. The custom expression used in that filter is shown below:

```
ApplyComparison ("Ucase ( #0 ) like #1",  
Company@DESC, "STRATEGY")
```

Each piece of the custom expression is explained below:

- The `Ucase` command changes the case of the argument to one that is in all capitals—for instance, `MicroStrategy` to `MICROSTRATEGY`.
- `( #0 ) like #1` is the actual comparison, comparing the first argument, `#0`, with the second argument, `#1`. This comparison evaluates as true if the left side of the expression, `Ucase ( #0 )`, contains the right side of the expression, `STRATEGY`.



The right side of the expression is actually `#1`, but that argument is replaced with `STRATEGY` in this example.

- `Company@DESC` sets the first argument as the description form of the `Company` attribute—instead of the ID form or any other attribute form that exists.
- `"STRATEGY"` is the second argument, a string that does not vary.

Using the filter on a report containing `Company` and `Department` generates the report below. Note that no matter how the `Company` attribute element

has been spelled or capitalized, the report includes the element as long as it contains the string “strategy”.

Company	Department
micro strategy	Tech. Support
MicroStrategy	Sales
Microstrategy	Finance
MICROSTRATEGY	ADVERTISING
MicroStrategy, Inc.	Legal



This report serves simply as an example; you cannot re-create it in MicroStrategy Tutorial.

Example: Qualifying on an attribute element containing a wildcard symbol

Your objective is to generate a report that contains all Store elements that have an * in their names. If you attempt to do this using the standard Add an Attribute qualification option in the Filter Editor, no filtering will take place, because MicroStrategy assumes that you wish to include all results when you use the * wildcard. Instead, you can use an ApplyComparison function in a custom expression qualification filter to search for this symbol.

Use the steps in [Custom expression qualification filters, page 156](#) along with the custom expression below, to create the filter you need:

```
ApplyComparison ("#0 like '*' ", Store@DESC)
```

Each piece of the custom expression is explained below:

- The `ApplyComparison` function is used for comparison operators, such as the `like` operator used in this example.
- The actual comparison is placed in double quotes, for example, `"#0 like '*' "`. The first argument, `#0`, is compared with the wildcard symbol you are searching for. Note that the asterisk is placed in single quotes since the whole comparison expression itself is in double quotes.
- `Store@DESC` sets the first argument as the description form of the Store attribute.

Using the filter described above in a report containing Store and Revenue, results in the data below, which contains only Store attribute elements with asterisks in their names. Had you used an attribute qualification instead of the ApplyComparison function, all stores would have been returned in the report data.

Store	Revenue
B*E*S*T	\$123,456
Marcy*s	\$456,789
Q*Mart	\$78,901
Wall*Mart	\$234,567,890



This report serves simply as an example; you cannot re-create it in MicroStrategy Tutorial since Tutorial does not contain a Store attribute.

Example: Filtering for last month's data using ApplySimple

Example: Filtering for last month, without transformations

An attribute-to-attribute qualification filter can be used to restrict your result set on a report to last month's data. Follow the steps below to create this filter.



Even though this example can be duplicated using a time-based transformation (information for which can be found in [Time-based transformations, page 91](#)), this example uses an Apply function to demonstrate the functionality of custom expressions and Apply functions. In general, MicroStrategy strongly advises against using Apply functions except when you are using them for functionality which does not currently exist in the software. See [Apply functions in metrics: Using database-specific functions, page 164](#) for more information.

To re-create this attribute-to-attribute qualification filter example

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then select **Filter**. The Filter Editor opens.

If the New Filter dialog box is displayed, click the **Empty Filter** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 In the Filter Editor, drag the **Month** attribute from the **Object Browser** to the **Filter definition** pane. The Attribute Qualification pane opens.
- 3 From the **Qualify On** drop-down list, select **ID**.
- 4 From the **Operator** drop-down list, select **Exactly**.
- 5 From the drop-down list, choose **Custom**.
- 6 Type the following custom expression in the text box:

```
ApplySimple("DatePart('yyyy',  
DateAdd('m', -1, date())) * 100 + DatePart('m',  
DateAdd('m', -1, date()))",0)
```



Your Relational Database Management System (RDBMS) might require slightly different syntax than the above. Any time you use an Apply function, it must be customized to the syntax of your particular RDBMS. The functions in this ApplySimple expression are specific to an Access database, which is recommended only for certain pre-production environments but never for an actual production environment.

- 7 Click **OK** to close the Attribute qualification pane.
- 8 Save the filter.

A filter which uses the custom expression above, compares the dates of data in your data warehouse to the system date, returning only data from last month. Therefore, whenever you execute the report, you always see results for the previous month.

Place this filter on a report with the Month and Region attributes and the Revenue metric to display regional revenue data for last month. The report shown below was executed in Oct 2006, so report generation yielded only

data from Sep 2006. If you regenerate the report in Dec 2007, the new report will return data only from Nov 2007.

		Metrics	Revenue
Month	Region		
Sep 2006	Northeast		\$76,414
	Mid-Atlantic		\$83,890
	Southeast		\$65,294
	Central		\$39,518
	South		\$45,829
	Northwest		\$40,413
	Southwest		\$106,484
	Web		\$46,531

Analyzing the custom expression's syntax

The report is executed on Oct 30, 2006. The custom expression takes the current, or system, date and decrements it by one month, changing it from 20061030 to 20060930. This is actually done twice but separately in the expression so that first, the year, 2006, can be extracted, and then the decremented month, 09, can be extracted. (Even though the date is decremented twice, it is decremented in separate places, yielding the same result in both places, 20060930, instead of a “double decremented” result of 20060830.) The year is then multiplied by 100, yielding 200600. Finally, 200600 is added to the month, 09, resulting in 200609. This is the date of the previous month, in YYYYMM format and is used to filter out all data except that from Sep 2006.

Each piece of the custom expression is explained below:

- The `DateAdd` function allows you to shift a date forward or backward. In both instances that the `DateAdd` function is used in this example, `-1` subtracts one month (indicated by `m`) from the system date (returned by the `Date()` function).

In this example, the report has been executed on October 30, 2006, so the system date is 20061030. The `DateAdd` function returns 20060930, the date that is one month prior to the system date.

- The `DatePart` function returns a specified part of a given date. The first `DatePart` function extracts the year (as specified by `yyyy`) from the result of the `DateAdd` function. The second `DatePart` function extracts the month (specified by `m`) from the result of the `DateAdd` function.

In this example, the first `DatePart` function returns 2006; the second returns 09 (after the month has been decremented).

- The result of the first `DatePart` function is multiplied by 100 and then added to the result of the second `DatePart` function, 09, yielding 200609, using the `ApplySimple` function. `ApplySimple` functions are used for simple single-value functions such as the arithmetic operators + and -. More information on `ApplySimple` and other functions can be found in [Apply function types, page 165](#), as well as the *Standard Functions* chapter of the *Functions Reference*.

Joint element lists: Grouping elements from multiple attributes

Joint element lists allow you to choose attribute elements from different attributes to filter the report result set. Unlike attribute qualifications, joint element lists allow you to join attribute elements and then filter on that attribute result set. In other words, you can select specific element combinations, such as quarter and category. As in the report sample included below, you can filter on electronics in Q1 2003 and music in Q3 2003.

A joint element list is an advanced qualification.

Joint element list example

This example refers to objects saved in the MicroStrategy Tutorial. The directory path within MicroStrategy Developer is `Public Objects\Reports\MicroStrategy Platform Capabilities\Advanced Reporting Guide`. You can follow the steps to interact with the filters and report, or you can view the samples without creating your own.



Remember to save any objects that you create under a different name, so that you do not overwrite the samples in the MicroStrategy Tutorial.

Before creating a joint element list, you must ensure that the Advanced Qualification option is displayed on the Filter Editor. From MicroStrategy Developer, complete the following steps:

- 1 Select **My Preferences** from the **Tools** menu.
- 2 Choose the **Editors** tab.

- 3 Click **Filter Options**.
- 4 Select **Show advanced qualification**, if it is not already selected.
- 5 Click **OK** to return to MicroStrategy Developer.

Open the Basic Report. Note that Leanne Sawyer's revenue is \$316,786. This is sales for all time and all categories. You need to see revenue for specific quarter and category combinations, for example, electronics in Q1 2003 and music in Q3 2003. To do this, switch to Design View and create a joint element list, as described below.

To create a joint element list

- 1 Double-click in the Report Filter pane to add a new qualification.
- 2 Select **Add an Advanced Qualification** and click **OK**. The Advanced Qualification pane opens.
- 3 Select **Joint Element List** from the Option pull-down list.
- 4 Select **Category** and **Quarter** from the Available attributes list and click **>** to add them to the Selected attributes list.
- 5 Click the **Add** icon to the right of the Element list. The first value in each attribute is added to the list.
- 6 Click the **Modify** icon to the right of the Element list. The Select Element List dialog box opens.
- 7 Double-click **Electronics** to change the category.
- 8 Select **Quarter** from the Available Elements drop-down list.
- 9 Double-click **Q1 03** to change the Quarter.
- 10 Click **OK** to return to the Advanced Qualifications dialog box.
- 11 Click the **Add** icon to add another element. Again, the first value in each attribute is added by default.
- 12 Select the new element and then repeat steps 6 through 10, this time changing Category to **Music** and Quarter to **Q3 03**.

13 Click **OK** to save the new qualification.

Execute the report. The results are displayed below:

Region	Employee	Metrics	Revenue	Cost	Profit
Northeast	De Le Torre Sandra		\$ 33,842	\$ 25,951	\$7,891
	Kelly Laura		\$ 16,767	\$ 12,921	\$3,846
	Kieferson Jack		\$ 31,292	\$ 23,584	\$7,708
	Sawyer Leanne		\$ 18,901	\$ 14,488	\$4,413
	Sonder Melanie		\$ 28,152	\$ 21,485	\$6,667
	Yager Beth		\$ 21,608	\$ 16,613	\$4,995



This report is saved as **Joint Element List**.

Notice that Sawyer's revenue is now only \$18,901. The decreased revenue reflects the qualification, since only sales for electronics in the first quarter of 2003 and the sales for music in the third quarter of 2003 are included in the metric calculations.

Prompted filters: Asking for user input for filter conditions

You can create filters that prompt you for answers when you run reports. These prompted filters allow a report to have a dynamic report definition, which can change with each query when the information in the prompt dialog box is altered.

In other words, a prompted qualification in a filter allows you to dynamically modify the contents of a report at run time. (Recall that a filter is composed of at least one qualification.) By applying certain conditions on one or more of the attributes, attribute elements, metrics, or other objects that a report can include, it is possible to:

- Apply conditions or modify components in a report at run time, eliminating the need for preliminary definitions
- Execute a report multiple times, selecting a different answer to a given prompt each time, and have, as result, a set of immediate answers to compare

Most types of qualifications allow prompts. For more information on each, see:

- [Prompted attribute qualifications, page 175](#)
- [Prompted metric qualifications, page 175](#)
- [Report object prompts, page 176](#) (using shortcut-to-a-report qualifications)
- [Filter object prompts, page 176](#) (using shortcut-to-a-filter qualifications)



For more information on prompts, see [Chapter 6, Advanced Prompts](#).

Prompted attribute qualifications

A prompted attribute qualification allows you to qualify on the values of attribute elements, attribute forms, or operators when you run a report. You can create the following types of attribute qualification prompts:

- **Choose from all attributes in a hierarchy** allows you to choose an attribute to qualify on when you run a report. You are, however, restricted to choosing only the attributes from the selected hierarchy. After selecting the attribute, you can qualify on the ID or create an element list filter.
- **Choose from an attribute element list** allows you to apply qualifications to an attribute form. You can choose an attribute from a list of attributes and qualify on the elements of the attribute.
- **Value prompt** allows you to select a single value on which to qualify, such as a date, a specific number, or a specific text string.
- **Qualify on an attribute** allows you to apply qualifications to an attribute form. You can choose an attribute from a list of attributes and qualify on an attribute form.

Prompted metric qualifications

A prompted metric qualification allows you to select a function, an operator, or a value for a metric when you run a report. You can create the following types of metric qualification prompts:

- **Qualify on a metric prompt** allows you to qualify on a metric. You can choose a metric by specifying a single metric to use, or by specifying a

search object to restrict the list of metrics from which you can choose a metric, when a report is run.

- **Metric Object prompt** allows you to select one or more metrics that meet specific criteria when a report is run. For example, you could use a search to return a list of all metrics that contain a certain fact. From that list of metrics, you can then choose the metrics that you want to see on the report.
- **Value prompt** allows you to select a single value on which to qualify, such as a date, a specific number, or a specific text string.

Report object prompts

Choosing a prompt for a shortcut-to-a-report qualification creates a report object prompt that uses the selected report as the default answer. When you use this report object prompt in a report, it prompts you to choose a report as the filtering criteria during run time.



A shortcut-to-a-report qualification uses the results of one report to be included in another report. For more information, see the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*.

You can define a report object prompt by specifying a search object or specifying a predefined list of objects to choose from, while executing a report.

Filter object prompts

Choosing a prompt for a shortcut-to-a-filter qualification creates a filter object prompt and allows you to choose the filters to be included in a report. You can define a filter object prompt by specifying a search object or specifying a predefined list of objects to choose from, while executing a report.



A shortcut-to-a-filter qualification uses an existing filter, either as is or with additional conditions, on a report. For more information, see the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*.

Shortcut vs. embedded filters

You can add a filter to a report in either of the following ways:

- Apply a previously created, stand-alone filter to a report during report creation. For steps, see [Adding a stand-alone filter to a report, page 177](#).

When you add a stand-alone filter to a report, you can choose to create a shortcut to the filter or to embed the filter in the report. The differences are described below:

- When you create a shortcut to a stand-alone filter, changes that you make to the filter are propagated to this report and to all other reports that use the filter as a shortcut. Changes that you make to this filter within this report are propagated to the filter, and to all other reports that use the filter as a shortcut. This is sometimes referred to as a linked filter.

Use this option to share report caches. For detailed information about report caches, see the *Caching* chapter in the *System Administration Guide*.

- When you embed a copy of the filter in the report, changes that you make to the stand-alone filter are not propagated to this report. Changes that you make to the embedded filter within this report are not propagated to the filter. This is also called a local filter.
- Create a filter at the same time that you create a new report. This filter is embedded within the report and cannot be used in another report. For steps, see the *MicroStrategy Developer help* or the *Building Query Objects and Queries, for Designers* chapter in the *Basic Reporting Guide*.

When you save a report that contains a stand-alone filter, you can choose to keep the shortcut or to embed a copy of the filter in the report. For steps, see [Saving a report with a shortcut to a stand-alone filter, page 178](#).

Analysts of report data can also apply filtering conditions to a report, for the individual instance of the report that they are currently viewing. This is called a view filter, and requires the MicroStrategy OLAP Services product. For steps to use a view filter, see the *MicroStrategy Developer help*.

Adding a stand-alone filter to a report

The following procedure describes how to apply a previously created, stand-alone filter to a report during report creation.

To add a filter to a report



This procedure assumes that you have already created a stand-alone filter. For steps to create a new filter, see the *MicroStrategy Developer help*.

- 1 Edit the report to which you want to add a filter.
- 2 If the Object Browser is not already displayed, open it, by selecting **Object Browser** from the **View** menu.
- 3 Use the Object Browser window to search for and locate the filter that you want to add to the report.
- 4 Right-click the filter, and select one of the following (the options are described in more detail above):
 - **Replace Report Filter with a shortcut to this filter** to connect the stand-alone filter to this report. Changes to the stand-alone filter are propagated to the report, and vice versa.
 - **Replace Report Filter with a copy of this filter** to embed the filter in the report. Changes to the stand-alone filter are not propagated to the report, and vice versa.

The filter appears in the Template definition section. If it is a shortcut, the filter name is displayed. An embedded filter is named Local Filter.



If you drag and drop the filter into the Report Filter definition pane, the filter is embedded as a copy.

- 5 Save the report. If you added the filter as a shortcut, you must save the report with the shortcut to retain that functionality. See [To save a report with a shortcut to a stand-alone filter, page 179](#) for steps.

Saving a report with a shortcut to a stand-alone filter

When you save a report that contains a stand-alone filter, you can choose to save the filter as either a copy or a shortcut.

- When you create a **shortcut** to a stand-alone object, changes that you make to the object are propagated to this report and to all other reports that use the object as a shortcut. Changes that you make to this object within this report are propagated to the object, and to all other reports

that use the object as a shortcut. Use this option to share report caches. This is sometimes referred to as a linked template or filter.



Reports with shortcuts to stand-alone templates and filters allow you to share report caches. If this is your intent, you must select the **Retain the shortcuts** option in the procedure. For detailed information about report caches, see the *Caching* chapter in the *System Administration Guide*.

- When you **embed a copy** of the object in the report, changes that you make to the stand-alone object are not propagated to this report. Changes that you make to the embedded object within this report are not propagated to the object. This is sometimes referred to as a local or report-specific template or filter.

To save a report with a shortcut to a stand-alone filter

1 Save the report with a shortcut to a stand-alone filter:

- a Click **Save and Close** on the toolbar of the Report Editor or Report Viewer. The Save Report As dialog box opens.
- b Select the folder in which to save the report, then type the name of the new report.
- c Click **Save**. The Advanced Save Options dialog box opens.



If the report contains prompts, the Save Options dialog box opens after you have named the report. To access the Advanced Save Options dialog box, click **Advanced** on the Save Options dialog box. For steps to save a prompted report, see the *MicroStrategy Developer help*.

2 Select one of the following options for saving the report:

- **Create a local copy of the filter.** Changes that you make to the filter are not propagated to the original object, and vice versa. This is also called a local or embedded filter.
- **Retain the shortcuts to the filter.** Creating shortcuts allows you to use an object on a report, taking advantage of the benefits of object reuse. Changes that you make to the filter are propagated to the original object, and vice versa. Use this option to share report caches.

3 To use the selections that you made above as the defaults, select the **Remember answers the next time I am prompted** check box.

- 4 Click **OK** to save the report.

CUSTOM GROUPS AND CONSOLIDATIONS

Grouping Attribute Elements

Introduction

A custom group is a set of special filters that can be placed on a template. A custom group is made up of an ordered collection of elements called custom group elements.

Consolidations are used to specify the data that you want to view in your report. They allow you to group attribute elements in new ways without changing the metadata and warehouse definitions.

Both of these objects allow you to qualify a report on a row-by-row basis.

The last section of the chapter, *Custom group and consolidation comparison, page 221*, summarizes the differences between these objects.

Information about custom groups in this chapter includes:

- *About custom groups: Grouping attribute elements with banding and filtering, page 182*
- *Custom group effects on SQL generation, page 187*
- *Custom group structure on reports, page 191*

- [Creating a custom group, page 193](#)
- [Editing and formatting a custom group, page 194](#)
- [Determining custom group interaction with the report filter, page 200](#)
- [Deleting a custom group, element, or condition, page 206](#)

Information about consolidations includes:

- [Consolidations: Grouping attribute elements with arithmetic and logical operators, page 207](#)
- [Consolidation elements, page 209](#)
- [Evaluation order, page 213](#)
- [Consolidations and SQL, page 214](#)
- [Consolidation example, page 220](#)

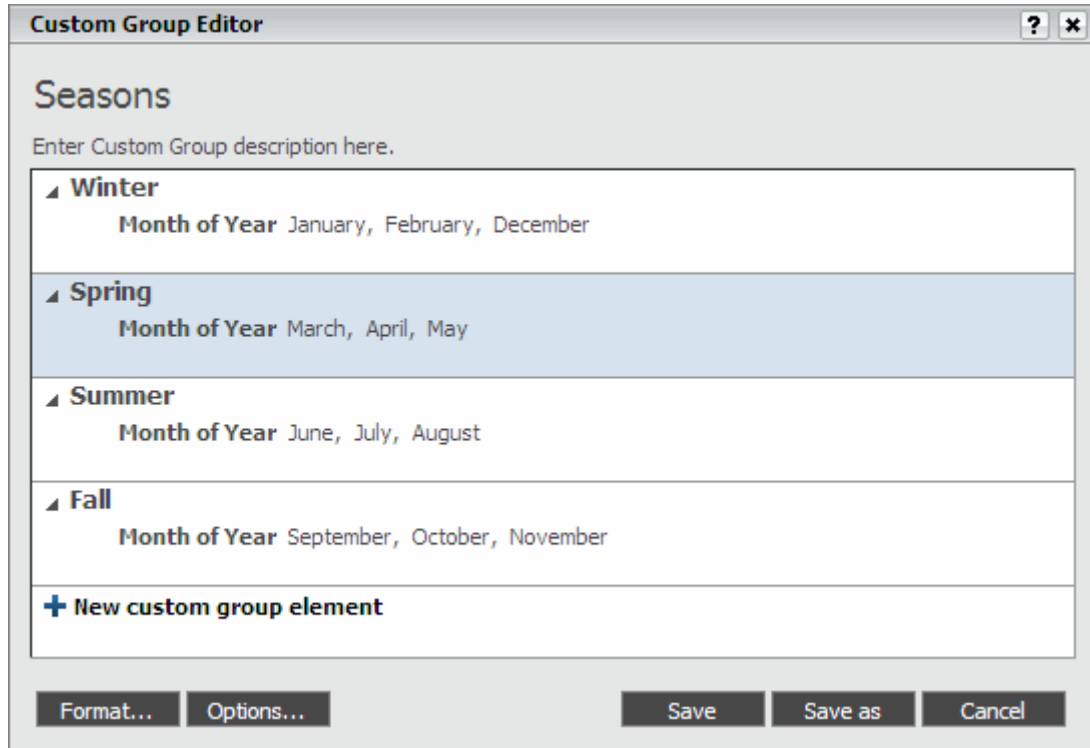
If you have a large number of objects to create or edit, you can also use MicroStrategy Command Manager to work with these objects. Command Manager manipulates MicroStrategy objects by means of user-created scripts rather than a graphical user interface. For more information about Command Manager, see the *Automating Administrative Tasks* chapter in the *System Administration Guide*. For specific Command Manager syntax for working with custom groups or consolidations, open Command Manager and click **Help**.

About custom groups: Grouping attribute elements with banding and filtering

A custom group lets you group attribute elements from the same or different attributes to meet your reporting requirements. This allows you to group attribute elements on a report in a way that is not defined in your data warehouse. From the perspective of the report design process, a custom group is an object that can be placed on a template and is made up of a collection of elements called custom group elements. Each element contains its own set of filtering or banding qualifications.

For example, your data warehouse contains data based on 12 months of the year, but you want to display data in a report for each season in a year. You can create a custom group with elements representing each season, such as

Winter and Spring, then add a condition to each custom group element to display data only for those months in each season. In the image below, which shows a custom group being created in MicroStrategy Web, the Winter custom group element displays data for the months of December, January, and February; the Spring element displays data for March, April, and May; and so on.



About custom group elements

A *custom group element* is a logical expression of qualifications. Each custom group element is made up of:

- A header, which contains the name of the custom group element and is displayed as a row in report results. The header is an arbitrary name that you define when you create the element. This name can be displayed on the report, so you should choose a meaningful name for the grouping of elements that you are defining. The Custom Group Editor provides you with different options for displaying the headers. For details, see [Changing the position of element headers, page 195](#).
- One or more conditions (also called filtering qualifications) that must be met to include data in the custom group element, such as “Quarter=Q1”. You can define any qualification, logical expression, or banding of the

qualification, or you can use previously created filters to build the custom group element.

You can combine different types of filtering qualifications in the same custom group element. For example, you can combine “Region = Northeast” (an attribute qualification) with “Revenue > \$1 million” (a metric qualification). Each set of conditions resolves into a list of attribute elements after the report is run.

You can add that qualification to the report filter so that it determines the values of the custom group elements. You can then specify how the custom group interacts with the report filter, as described in [Determining custom group interaction with the report filter, page 200](#).

Custom group elements can include a logical expression containing any of the following qualification types (“Developer” and “Web” indicate which product the qualification can be created in):

- Attribute qualification (Developer, Web): Restricts data based on the value of an attribute form; lets you view data only for attribute elements in a list; or compares two attributes to filter data.
- Set qualification (Developer, Web): Restricts data based on the value, rank or percentage of a metric, or by comparing the values of two metrics.
- Custom group banding qualification (Developer): Slices data into multiple ranges, or bands, based on metric values.



You cannot combine banding qualifications with other types of qualifications.

- Shortcut-to-a-report qualification (Developer, Web): Restricts data using the results of an existing report. Also known as report qualifications or reports as filters.
- Shortcut-to-a-filter qualification (Developer, Web): Restricts data using an existing filter. Also known as filter qualifications or embedded filters.
- Shortcut-to-a-prompt qualification (Web): Restricts data using the answers to an existing prompt.



MicroStrategy Developer allows prompts in most qualification types. For details, see [Prompted custom groups, page 201](#).

Custom group examples

Store inventory examples: Example 1

In MicroStrategy Developer, you can create the custom group Store Inventory as follows:

- Small stores with low inventory

```
Store Sales < 50
AND
Store Inventory < 200
```

- Large stores with low inventory

```
Store Sales > 50
AND
Store Inventory < 200
```

The custom group elements in this example are:

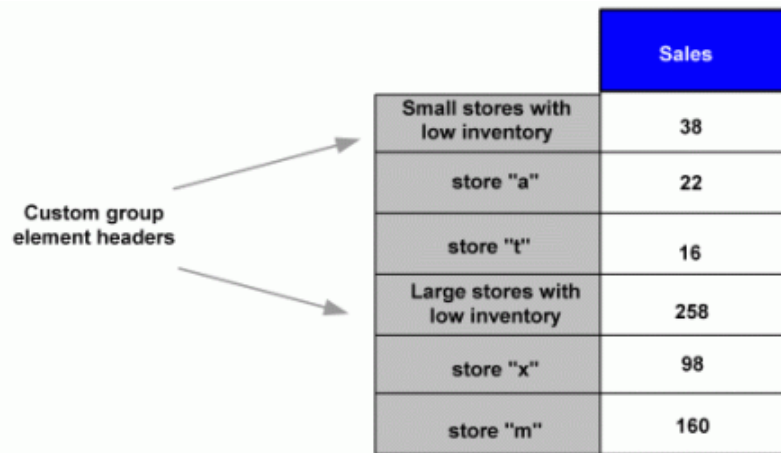
- Small stores with low inventory, which is a logical expression of the following two metric qualifications (MQs):

```
Store Sales < 50           (MQ1)
AND                        (logical operator)
Store Inventory < 200      (MQ2)
```

- Large stores with low inventory, which is a logical expression of the following two metric qualifications:

```
Stores Sales > 50          (MQ1)
AND                        (logical operator)
Store Inventory < 200      (MQ2)
```

Depending on the options that you select in the Custom Group Editor, the custom group could appear on the report in MicroStrategy Developer as shown below.



The diagram shows a table with a custom group element header pointing to a specific row. The table has two columns: 'Sales' (highlighted in blue) and an unnamed column. The rows are grouped into two categories: 'Small stores with low inventory' and 'Large stores with low inventory'. Each category has three sub-rows labeled 'store "a"', 'store "t"', and 'store "m"'. The values in the 'Sales' column are 38, 22, 16, 258, 98, and 160 respectively.

	Sales
Small stores with low inventory	38
store "a"	22
store "t"	16
Large stores with low inventory	258
store "x"	98
store "m"	160

The output level of a custom group (in this example, Store) is based on the filtering conditions of the element. Each element in the custom group can have a different output level. Elements are completely independent. The fact that they are included in the same custom group means only that they are displayed on the same report.

Store inventory examples: Example 2

After analyzing an inventory report covering the past six months, you notice that you have an excess of certain items in your warehouse. As a special promotion, you would like to offer these items at discounted prices to your best customers. To do this, you need to obtain a list of your top ten customers along with a list of your five lowest selling items on the same report.

This requirement is requesting two different reports: the top ten customers and the five lowest selling inventory items. You can create a custom group with the following custom group elements:

- Top ten customers
- Five lowest selling items

Each custom group element will have a different qualification applied to it. In this case, the first element is the top ten customers ranked by revenue. The second element is the bottom five items by revenue. For each element, change the display options to show the element names and individual items. This allows the names of the customers and items to be displayed.

Create a report with this custom group and run it. Your top ten customers and bottom five items, in terms of revenue, are displayed.

Custom group effects on SQL generation

You can consider custom groups as many different reports “stacked up” together. The SQL for a report with a custom group is likely to be very complex. Each of the individual “mini-reports” that make up the entire custom group report will have at least one, if not more, SQL passes of its own. The Analytical Engine stacks up all these “mini-reports” to create the final results. In addition, numerous temporary tables may be created and dropped to hold intermediate data.

Therefore, running a report with a custom group is equivalent to running many different reports and putting them together. As a result, custom groups are SQL-intensive in the sense that they are likely to generate many passes of SQL to the database.

Benefits of using a custom group

The benefit of a custom group is its ability to group attribute elements in a way that is not defined in the data warehouse. You can create “relationships” between the attribute and the custom group. A custom group can organize attribute elements through the following:

- Attribute qualification
- Set qualification
- Banding
- Advanced qualification
- Shortcut-to-a-report qualification
- Shortcut-to-a-filter qualification
- Shortcut-to-a-prompt qualification

Refer to [Chapter 3, Advanced Filters](#), for more information on these qualification types.

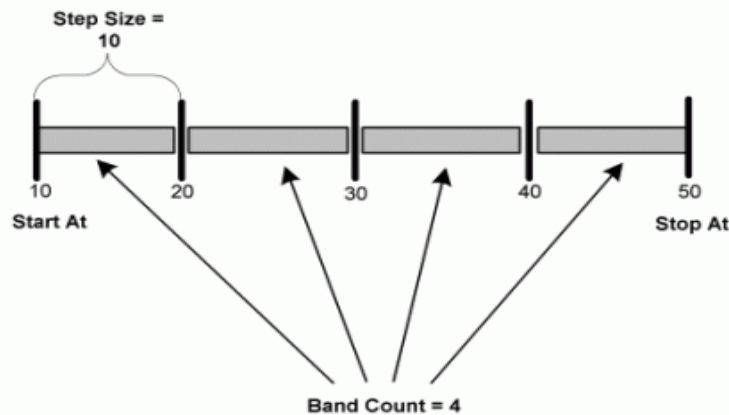
Banding qualification: Grouping by metric value intervals

Banding qualifiers enable you to create banding custom group elements. Banding is a method of displaying the groups of custom group elements. Banding is defined by the output level of elements using the values of a metric. For example, you can slice the list of stores (Store attribute elements) using the values of the Total Sales metric. Suppose you have created a report that ranks stores according to the revenue generated by each store. You might decide to group the stores by creating one group for the top 10 stores, a second group for stores 11-20, and a third group for stores 21-30.

You can apply different types of banding:

- **Band size:** To slice the range of metric values defined by “start at” and “stop at” values into a number of bands, each defined by the parameter “step size.”

For example, in the following diagram the “start at” value is 10, “stop at” is 50, and “step size” is 10. These settings slice the group into four bands.



- **Band count:** To define the number of equal bands into which the range of metric values is sliced. The range is defined by “start at” and “stop at” values, as band size is. However, band size defines the size of each band, while band count defines the number of bands.

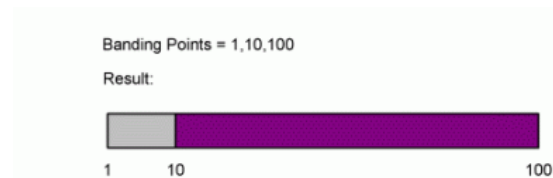
For example, to use band count to achieve the same results as the preceding diagram, set the band count to four, “start at” to 10, and “stop at” to 50. If you set the band count to five instead, each of the five bands have a size of eight.

- **Banding points:** To specify the value where a band is placed. This enables you to produce bands of different sizes.



The engine uses its internal logic to create the bands based on the banding points that you specify.

For example, you want to create a report with two bands, one band showing the top 10 stores and the second band showing stores 11-100. For this, you must use three points—1, 10, and 100—as shown in the following figure.



- **Band for each distinct metric value:** To create a separate band for each value calculated by the metric. This type of banding qualification directly uses the results of a metric as bands. It is very useful with metrics that already contain the logic needed to calculate sequential band numbers. Such metrics use mathematical formulas, NTile functions, Band functions, or Case functions.

For example, a metric uses the NTile function to group revenue values into three groups. The custom group element is therefore sliced into three bands as well.

Banding points example

Report requirements

You want to create a report that ranks employees according to the revenue that each employee generates. Based on their revenue, the report should segregate the employees into the following three groups: Top for the top 10%, Next for the next 40%, and Lowest for the lowest 50%.

Solution

To create the report in MicroStrategy Developer, create a custom group called Employee Revenue and specify the banding points as 0, 10, 50, and

100. Create a report that uses this custom group. A sample report is shown in the following image.

Employee Revenue		Metrics	Revenue
Employee			\$16,936,434
Top			\$3,192,594
	Bates Michael		\$783,886
	Becker Kyle		\$692,441
	Walker Robert		\$1,716,267
Next			\$6,903,527
	Bell Caitlin		\$624,961
	Benner Ian		\$526,867
	Brown Vernon		\$548,862
	De Le Torre Sandra		\$514,524
	Gale Loren		\$575,268
	Gedot Harriet		\$434,199
	Hollywood Robert		\$591,776
	Johnson Andrew		\$474,239
	McClain Sean		\$556,149
	Nelson Arthur		\$486,737
	Pierce Charles		\$513,751
	Schafer Rose		\$514,833
	Strome Fred		\$541,361
Lowest			\$6,840,313
	Bernstein Lawrence		\$403,122
	Conner Beatrice		\$380,503
	Corcoran Peter		\$346,565
	Ellerkamp Nancy		\$432,879
	Folks Adrienne		\$418,967
	Hall David		\$358,542
	Hunt Matthew		\$418,415
	Ingles Walter		\$377,583
	Kelly Laura		\$329,888
	Kieferson Jack		\$389,888
	Lynch Sam		\$391,809

The engine considers 0 as the highest banding point and 100 as the lowest banding point; hence, based on the revenue, it places the highest 10% of employees in the first band and so on.

Metric value banding example

Report requirements

You want to create a report that ranks regions based on their revenue and provides the profit margin for each. The groups are Top Performers, Average Performers, and Under-performers.

Solution

First, create a metric that uses the NTile function to create three groups based on revenue. Use that metric in a custom group that bands based on

metric value and for each metric value. Change the band names to Top Performers, Average Performers, and Under-Performers. Create a report that uses this custom group. A sample MicroStrategy Developer report is shown in the following image.

Revenue	Metrics	Profit Margin
Regional Revenue Ranking		16.91%
Top Performers		15.47%
	Southeast	15.61%
	Web	15.34%
Average Performers		16.21%
	Central	15.70%
	South	17.56%
	Southwest	15.45%
Under-performers		17.40%
	Northeast	17.67%
	Mid-Atlantic	17.29%
	Northwest	17.36%

Note that this custom group shows the element names and items within the element. For more information on this option, see [Custom group structure on reports, page 191](#).

Custom group structure on reports

Custom groups give a natural hierarchical structure to their elements. Each custom group element can be viewed as a set of smaller grouping elements, which can be repeatedly broken down until the actual items are reached. For example, in a Ranking custom group, the top-level element is Sales, which can be separated into the bands of Top Cities, Average Cities, and Bottom Cities. Each band can be further divided into elements such as San Diego and Berlin. By default, only the element names are displayed on the report, as shown in the following report sample in MicroStrategy Developer.

Ranking	Metrics	Dollar Sales
Sales		\$ 191,039.00
Top Cities		\$ 91,826.00
Average Cities		\$ 62,377.00
Bottom Cities		\$ 36,836.00

If you change the display option of the custom group element in the Custom Group Editor, this division can be displayed in more detail. For example, the

following is the same report with the element names and individual items displayed.

Ranking	Metrics	Dollar Sales
Sales		\$ 191,039.00
Top Cities		\$ 91,826.00
	San Diego	\$ 15,633.00
	Washington, DC	\$ 16,848.00
	Miami	\$ 17,047.00
	New York	\$ 18,132.00
	Charleston	\$ 24,166.00
Average Cities		\$ 62,377.00
	Milwaukee	\$ 14,181.00
	New Orleans	\$ 11,595.00
	Boston	\$ 12,414.00
	London	\$ 12,468.00
Bottom Cities		\$ 36,836.00
	Berlin	\$ 8,020.00
	San Francisco	\$ 9,653.00
	Salt Lake City	\$ 4,773.00
	Seattle	\$ 4,670.00
	Paris	\$ 9,720.00

While hierarchical display is similar to drilling on a report, drilling requires additional executions of the report. However, drilling allows you to manipulate a report on the fly.

To view custom groups in a hierarchical structure on a report, you must:

- Expand the item display of at least one custom group element
- Enable hierarchical display for the custom group
- Display the hierarchical view in the report (MicroStrategy Developer only)

These tasks are completed at different levels and in different MicroStrategy interfaces, as described in the following table.

Level	Target	Interface	Settings
Element	Individual custom group element	Display Options in Custom Group Editor	Display: • Only the element names • Only the individual items within this element • The element names, individual items, and expand the items if possible • The individual items and expand them if possible Note: The last option is available only for banding.
Custom group	All elements of a custom group	Options in Custom Group Editor	• Enable hierarchical display • Enable subtotals • Position of element headers

Level	Target	Interface	Settings
Report	All custom groups on a report	Report Data Options in Report Editor (Developer)	Display custom groups in hierarchical or flat view Note: In MicroStrategy Web, custom group display is controlled by the setting in the Custom Group Editor.
Report	Individual custom groups	Report Data Options in Report Editor (Developer)	Enable subtotals Note: In MicroStrategy Web, subtotal display is controlled by the setting in the Custom Group Editor.
Project	All custom groups in a project	My Preferences in MicroStrategy Developer	- Show advanced qualification in Custom Group Editor - Show all prompt buttons in Custom Group Editor - Show tip boxes in Custom Group Editor - Trim leading spaces

Steps are below to create a custom group in MicroStrategy Web. For detailed steps to set options, in the Custom Group Editor click **Help**.

Creating a custom group

The steps below show you how to create a custom group in MicroStrategy Web. For steps to create a custom group in MicroStrategy Developer, see the *MicroStrategy Developer help*.



You can also create, edit, or delete multiple custom groups at the same time by using a Command Manager script. For example, you can edit all custom groups to use a different attribute element. Command Manager is a MicroStrategy tool designed to automate certain tasks and processes. For more information about Command Manager, including steps, see the *Command Manager* chapter of the *System Administration Guide*.

Prerequisite

- You must have the Web Use Custom Group Editor privilege to access the Custom Group Editor.

To create a custom group

- 1 Browse to the folder in which to create the new custom group, then click the **Create Custom Group**  icon. The Custom Group Editor is displayed.
- 2 Type a name for the custom group in the **Enter Custom Group Name Here** field.
- 3 Type a description for the custom group in the **Enter Custom Group Description Here** field.

To define custom group elements

- 4 Click **New Custom Group Element**. The Condition Editor opens.
- 5 Define the conditions to use to filter data by selecting an attribute, metric, report, or filter, then specifying the appropriate options. For detailed steps to define a condition, click **Help** in the Condition Editor.
- 6 Specify a name for the custom group element by clicking on the name of the element and typing a name in the field.
- 7 Repeat the steps above, beginning at *To define custom group elements*, to define each additional custom group element.

To save the custom group

- 8 Click **Save**. The Save As dialog box opens.
- 9 Browse to the location to save your custom group in, then click **Save**.
- 10 Click **OK** to return to the Custom Group Editor.

Editing and formatting a custom group

You can use any of the following ideas to further design your custom group so it appears on reports the way that you want it to:

- *[Changing the position of element headers, page 195](#)*
- *[Enabling subtotals for custom groups, page 196](#)*

- [Sorting custom groups, page 197](#) (based on attribute ID or metric value)
- [Determining custom group interaction with the report filter, page 200](#)
- [Prompted custom groups, page 201](#)

You can also edit the custom group elements and the conditions in a custom group element. For steps, see [Editing custom group elements, page 202](#) and [Editing the conditions in a custom group element, page 204](#).

Changing the position of element headers

You can change the position of the element headers relative to the custom group's elements. You can choose to display the element headers at the top or bottom position, or inherit the default custom group definition. For this, you must set the display option of the custom group to display both the element headers and its elements. If the display option of the custom group is set to display only either the element header or the elements, you cannot change the position of the element headers.

For example, the following MicroStrategy Developer report displays the element headers below their respective elements:

Areas	Metrics	Revenue
San Diego		\$ 2,397,919
San Francisco		\$ 1,050,983
Salt Lake City		\$ 418,415
Seattle		\$ 434,199
West		\$ 4,301,516
Atlanta		\$ 1,083,016
Washington, DC		\$ 1,413,865
Miami		\$ 933,170
Boston		\$ 1,325,448
New York		\$ 1,009,416
Charleston		\$ 1,999,475
East		\$ 7,764,390
Milwaukee		\$ 1,340,391
New Orleans		\$ 867,240
Fargo		\$ 432,879
Memphis		\$ 513,751
Central		\$ 3,154,261

Enabling subtotals for custom groups

As shown in the table in [Custom group structure on reports, page 191](#), you can enable custom group subtotals at the report or custom group level in Developer, as described below:

- The report level is governed by the Report Data Options, which allow you to disable or enable subtotals for each custom group on a particular report. For example, a report contains two custom groups. You can choose to subtotal on the Age Groups custom group but not on the Customer Deciling group. Alternatively, you can use the default setting, which is determined in the Custom Group Editor.
- The setting in the Custom Group Editor allows you to enable or disable subtotals for a particular custom group, regardless of the report in which it is used.

In Web, custom group totals are controlled by the setting in the Custom Group Editor.

Changing the position of totals

You can change the position of the totals for the custom group. This option is enabled only if subtotals have been enabled for the custom group. You can choose to display the totals at the top or bottom position, or inherit the default custom group definition.

For example, the following MicroStrategy Developer report displays the Total Revenue at the top of the report:

Areas	Metrics	Revenue
Total		\$ 30,440,334
West		\$ 4,301,516
San Diego		\$ 2,397,919
San Francisco		\$ 1,050,983
Salt Lake City		\$ 418,415
Seattle		\$ 434,199
East		\$ 7,764,390
Atlanta		\$ 1,083,016
Washington, DC		\$ 1,413,865
Miami		\$ 933,170
Boston		\$ 1,325,448
New York		\$ 1,009,416
Charleston		\$ 1,999,475
Central		\$ 3,154,261
Milwaukee		\$ 1,340,391
New Orleans		\$ 867,240
Fargo		\$ 432,879
Memphis		\$ 513,751

Sorting custom groups

A custom group is a convenient method to group attributes at the display level. Therefore, by default, custom groups are not sorted on reports and you cannot use the sort accessible from the toolbar or right-click menu option to sort custom groups. However, you can use the advanced sorting functionality to sort a custom group in either of the following ways:

- **Sort on the IDs of the attributes** that compose the custom group.
- **Inherit the attribute sort**, which uses the default sort of the attribute that is set to display first. The display order is set in the Report display forms list on the Attribute Editor Display tab. The default sort for each form is determined in the New Attribute Form dialog box.

For example, a custom group uses the Call Center attribute, which has both a numeric ID and a description. The description is defined to display on reports, and its default sort is ascending. If the sort criteria is defined to

inherit the attribute sort, the custom group is sorted in alphabetical order by the description, as shown below when displayed in MicroStrategy Developer:

Areas	Metrics	Revenue
West		\$5,455,319
San Diego		\$2,962,719
San Francisco		\$1,021,447
Salt Lake City		\$731,413
Seattle		\$739,741
East		\$15,246,981
Atlanta		\$1,052,108
Washington, DC		\$3,135,283
Miami		\$1,187,843
Boston		\$1,487,936
New York		\$7,066,478
Charleston		\$1,317,332
Central		\$10,418,647
Milwaukee		\$4,182,139
New Orleans		\$3,305,039
Fargo		\$847,227
Memphis		\$2,084,241

If ID sort is used, the custom group is sorted by the numeric ID of the attribute, as shown below:

Areas	Metrics	Revenue
West		\$5,455,319
Seattle		\$739,741
Salt Lake City		\$731,413
San Francisco		\$1,021,447
San Diego		\$2,962,719
East		\$15,246,981
Charleston		\$1,317,332
New York		\$7,066,478
Boston		\$1,487,936
Miami		\$1,187,843
Washington, DC		\$3,135,283
Atlanta		\$1,052,108
Central		\$10,418,647
Memphis		\$2,084,241
Fargo		\$847,227
New Orleans		\$3,305,039
Milwaukee		\$4,182,139



These settings are found in the Criteria column on the Sorting dialog box, which is accessed from the **Advanced Sorting** option on the **Data** menu in MicroStrategy Developer. For steps to sort custom groups, see the *MicroStrategy Developer help*.

Sorting by metric values of items

You can use the Keep Group Structure option to sort by the metric values of the items in each custom group element. For example, the Areas custom group used previously is defined to display the individual items within its elements. It is placed on a report with the Revenue metric. Before sorting, the report is displayed in MicroStrategy Developer as shown below:

Areas	Metrics	Revenue
West		\$5,455,319
San Diego		\$2,962,719
San Francisco		\$1,021,447
Salt Lake City		\$731,413
Seattle		\$739,741
East		\$15,246,981
Atlanta		\$1,052,108
Washington, DC		\$3,135,283
Miami		\$1,187,843
Boston		\$1,487,936
New York		\$7,066,478
Charleston		\$1,317,332
Central		\$10,418,647
Milwaukee		\$4,182,139
New Orleans		\$3,305,039
Fargo		\$847,227
Memphis		\$2,084,241

To sort each Call Center (item) within each area (element), add the custom group as a sort and change the **Criteria** to **Keep Group Structure**. Add the

metric as a second sort. The report now displays the call centers in order of their revenue values, within each element:

Areas	Metrics	Revenue
West		\$5,455,319
	Salt Lake City	\$731,413
	Seattle	\$739,741
	San Francisco	\$1,021,447
	San Diego	\$2,962,719
East		\$15,246,981
	Atlanta	\$1,052,108
	Miami	\$1,187,843
	Charleston	\$1,317,332
	Boston	\$1,487,936
	Washington, DC	\$3,135,283
	New York	\$7,066,478
Central		\$10,418,647
	Fargo	\$847,227
	Memphis	\$2,084,241
	New Orleans	\$3,305,039
	Milwaukee	\$4,182,139

Determining custom group interaction with the report filter

You cannot combine a banding qualification with another type of qualification (such as attribute qualification or metric qualification) in a custom group element. However, you can add that qualification to the report filter so that it will determine the values of the custom group elements. You can then specify how the custom group interacts with the report filter.

When a custom group that uses attribute qualifications is included on a report with a report filter, the report filter is always applied to the individual custom group elements. However, if you create a custom group using metric qualifications or banding qualifications, by default the report filter is not applied to the custom group elements.

Incorrect data can be returned in some scenarios. For example, a custom group displays revenue totals for customers in columns that represent the range of revenue that the customer has contributed. A customer that has contributed \$7,500 in revenue is shown in the column for customers that contributed \$5,000 to \$10,000 in revenue. This custom group is included on a report that has a report filter that restricts results to data only for the year 2007.

In this scenario, the report filter is evaluated after the custom group. If the same customer that has a total of \$7,500 only had \$2,500 in 2007, then the report would only display \$2,500 for that customer. However, the customer would still show in the \$5,000 to \$10,000 in revenue range because the custom group did not account for the report filter.

You can define the interaction between the report filter and the custom group to avoid this scenario, by using the **Custom Group interaction with report filter** setting. This setting has the following options:

- **No interaction - static custom group:** Report filter qualifications are not applied to custom groups that use metric qualifications or banding qualifications. Filtering is applied only after the custom group has been evaluated.
- **Apply report filter to custom group:** Report filter qualifications are applied to custom groups and are used to determine the values for each custom group element.
- **Apply report filter to custom group, but ignore related elements from the report filter:** Report filter qualifications that do not qualify on attribute elements that are used to define the custom group elements are applied to custom groups. These filter qualifications are used to determine the values for each custom group element. For example, a report filter that qualifies on the Customer attribute is not applied to a custom group that also uses the Customer attribute to define its custom group elements.



A report limit defined on a custom group report is ignored.

Prompted custom groups

A prompted qualification in a custom group allows you to dynamically modify the contents of a report at run time. By applying certain conditions on one or more of the attributes, attribute elements, metrics, or other objects that a report can include, it is possible to:

- Apply conditions or modify components in a report at run time, thus eliminating the need for preliminary definitions
- Execute a report multiple times, selecting a different answer to a given prompt each time, and have, as a result, a set of immediate answers to compare

The following types of qualifications allow prompts:

- Attribute qualifications
- Metric qualifications
- Shortcut-to-a-report qualifications
- Shortcut-to-a-filter qualifications

For more information on creating these types of prompts, see the *Basic Reporting Guide*.

Editing custom group elements

The steps below show you how to perform various editing tasks in MicroStrategy Web. For steps in MicroStrategy Developer, click **Help**.

Prerequisite

- You must have the Web Use Custom Group Editor privilege to create, edit, or delete custom groups in the Custom Group Editor in MicroStrategy Web.

The table below lists the different ways in which you can edit a custom group element, and steps to perform each task in MicroStrategy Web.

Goal	Steps in MicroStrategy Web
Create a new custom group element	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 Click Create New Custom Group Element, then select the appropriate options to define a filtering condition for the custom group element. For detailed steps to create a custom group and filtering conditions, see Creating a custom group, page 193. 3 Click Save to save changes.
Rename a custom group element	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 Click the name of a custom group element and type a new name in the field. 3 Click Save to save changes.

Goal	Steps in MicroStrategy Web
Add a condition to a custom group element	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 Click the arrow to the left of a custom group element to expand it. 3 Click New Condition, then select the appropriate options to define the filtering condition. For detailed steps to create a filtering condition, click Help within the Condition Editor. 4 Click Save to save changes.
Edit the conditions in a custom group element	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 Click the arrow to the left of a custom group element to expand it. 3 To edit a condition, click the condition's name, then select the appropriate options to redefine the condition. For detailed steps, see Editing the conditions in a custom group element, page 204. 4 Click Save to save changes.
Duplicate a custom group element	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 Hover the cursor over the element to duplicate. 3 From the right, click the Duplicate  icon. A copy of the element appears in the list. 4 Click Save to save changes.
Move a custom group element	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 Click the custom group element, then drag it to a new location. 3 Click Save to save changes.
Format a custom group element	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 From the left, click Format. The Format Custom Group dialog box opens. 3 Select a custom group element to format from the first drop-down list. 4 Select the part of the custom group element to format. The options are: • Element Header: Specify the formatting for the name of the custom group element. • Element Value: Specify the formatting for the data for the custom group element. • Individual Items Header: Specify the formatting for the individual items in the custom group element. • Individual Items Value: Specify the formatting for the data for each individual item. 5 Select the appropriate formatting options to format the custom group element. For more information on the formatting options available, click Help in the Format Custom Group dialog box. 6 Click OK to apply changes. 7 Click Save to save changes.

Goal	Steps in MicroStrategy Web
Specify how a custom group element is displayed in a grid report You can display the custom group element using the element name, individual items in the element, or both	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 Hover the cursor over the element to edit. 3 From the right, click the Element Display Options  icon. The Element Options dialog box opens. From the Display drop-down list, select one of the following: • Display element name only (Default): Displays the name of the custom group element. • Display individual items only: Displays the names of individual items in the custom group element. • Display element name and individual items: Both the name of the custom group element and the items in the custom group element are displayed. 4 Click Save to save changes.
Edit options such as whether attribute elements in a custom group element are indented, and how the custom group interacts with the report filter	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 From the left, click Options. The Custom Group Options dialog box opens. 3 Select the appropriate options, such as whether or not to display subtotals in the custom group. For details on the options available, click Help. The Preview pane displays a preview of the custom group when the currently selected options are applied. 4 Click OK to apply changes. 5 Click Save to save changes.

Editing the conditions in a custom group element

Once you have created a custom group, you can edit the conditions in a custom group element. For example, you can specify the order in which conditions are evaluated, or change the logical operator between two conditions.

The steps below show you how to perform various editing tasks in MicroStrategy Web. For steps in MicroStrategy Developer, click **Help**. For steps to create a custom group, see [Creating a custom group, page 193](#).

Prerequisite

- You must have the Web Use Custom Group Editor privilege to access the Custom Group Editor in MicroStrategy Web.

The table below contains editing tasks that you can perform on a condition in a custom group element, and steps to perform them in MicroStrategy Web.

Goal	Steps in MicroStrategy Web
Create or edit a condition	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 To expand the custom group element in which to create or edit a condition, click the arrow next to the element's name. 3 Do one of the following: • To create a condition, click Add Condition. • To edit an existing condition, click the condition. 4 Define or redefine the condition by selecting an attribute, metric, report, or filter, then specifying the appropriate options. For detailed steps to define a filtering condition, click Help in the Condition Editor. 5 Click Save to save changes.
Shift a condition up or down in the custom group element Conditions at the top of the custom group element are evaluated first	Before you can move a condition, the custom group element must include at least two conditions. 1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 To expand the custom group element in which to create or edit a condition, click the arrow next to the element's name. 3 Click the name of the condition, then drag it to a new location. 4 Click Save to save changes.
Group conditions to be evaluated together	Before you can group conditions, the custom group element must include at least three conditions. 1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 To expand the custom group element in which to create or edit a condition, click the arrow next to the element's name. 3 Click the right arrow between the two conditions to group. The conditions are grouped together and indented to the right. 4 Click Save to save changes.
Ungroup two conditions	1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 To expand the custom group element in which to create or edit a condition, click the arrow next to the element's name. 3 Click the left arrow between the conditions to be ungrouped. The conditions are ungrouped. 4 Click Save to save changes.
Change the logical operator between two conditions	Before you can change the logical operator, the custom group element must include at least two conditions. 1 Right-click the custom group, then select Edit. The Custom Group Editor opens. 2 To expand the custom group element in which to create or edit a condition, click the arrow next to the element's name. 3 Click the operator to change, then select a new logical operator from the menu that is displayed. The operator is changed and the two conditions are grouped together. 4 Click Save to save changes.

Deleting a custom group, element, or condition

Prerequisite

- You must have the Web Use Custom Group Editor privilege to access the Custom Group Editor in MicroStrategy Web.

To delete a custom group

- 1 Browse to the folder in which the custom group is stored.
- 2 Right-click the custom group, then select **Delete**.
- 3 Click **OK**. The custom group is deleted.

To delete a custom group element

- 1 Right-click the custom group, then select **Edit**. The Custom Group Editor opens.
- 2 Hover the cursor over the element to delete. From the right, click the **Delete element**  icon. The element is deleted.
- 3 Click **Save** to save changes.

To delete a custom group element condition

- 1 Right-click the custom group, then select **Edit**. The Custom Group Editor opens.
- 2 Expand the custom group element that contains the condition to delete, by clicking the arrow next to the element's name.
- 3 Hover the cursor over the condition to delete, then click the **Delete condition**  icon. The condition is deleted.
- 4 Click **Save** to save changes.

Consolidations: Grouping attribute elements with arithmetic and logical operators

Consolidations enable you to group together and to pick specific attribute elements. Further, consolidations allow you to place this grouping of attribute elements on a template just like an attribute. The elements of the consolidation appear as rows on your report, and they can have arithmetic calculations.

For example, suppose you want to see each season of the year as a separate row on a report, but Season does not exist as an attribute in your project. A consolidation allows you to group together the elements of the Month of Year attribute into various seasons and place them on the template. This consolidation will contain four consolidation elements, one for each season.

Summer consists of June + July + August, fall consists of September + October + November, and so on. The consolidation is placed in the rows of your report with the desired metrics in the columns. Therefore, when a user runs the report, the metric values for June, July, and August are added together to yield the value for Summer. This occurs for each of the seasons.



Consolidation elements do not have to be based on a single attribute, as described in this example. You can use attributes at different levels, such as Region and Country, or unrelated attributes, such as Country and Year. For details on creating consolidation element expressions, see [Consolidation elements, page 209](#).

In general, consolidations provide two powerful functions that enhance your reporting needs. These functions are:

- Create a virtual attribute
- Perform row level math

Create a virtual attribute

In the above example of the Seasons consolidation, the four different season consolidation elements are made up by adding together the respective Months of Year that belong to the different seasons. The fact that you can add together attribute elements in groups means you can aggregate data on a report at a level other than one of the predefined attributes. The effect

appears as if you had a Seasons attribute in your data model, as shown below.

		Metrics	Revenue
Seasons	Year		
Summer	2003		\$1,784,523
	2004		\$1,912,180
Winter	2003		\$2,539,620
	2004		\$2,003,397
Fall	2003		\$1,947,487
	2004		\$2,102,240
Spring	2003		\$2,605,000
	2004		\$2,041,987

Of course, you can get the same effect if you change your data model, and actually add a Seasons attribute to your Time hierarchy. However, adding an attribute is generally a very complex task because you have to ensure that the proper lookup and relationship tables exist in the warehouse. Consolidations allow you to avoid changing the data model, although in a limited way.

This Seasons consolidation is built by adding together respective Months of Year in the different seasons. But you are not limited to just adding. In fact, you can perform any simple arithmetic operation while building your consolidation.

Perform row level math

Consolidations allow mathematical operations between elements or element groups. You can perform arithmetic operations such as addition, multiplication, division, and subtraction. You can even use constants while specifying your elements.

This feature makes consolidations a powerful tool in reporting. It allows you to specify row level math for a report, so that a row in a report is specified by a mathematical operation.

Continuing with the Seasons example, the ratio of sales between different seasons can be calculated using row level math in a consolidation, as shown below:

Seasons	Year	Metrics	Revenue
Summer	2003		\$1,784,523
	2004		\$1,912,180
Winter	2003		\$2,539,620
	2004		\$2,003,397
Fall	2003		\$1,947,487
	2004		\$2,102,240
Spring	2003		\$2,605,000
	2004		\$2,041,987
Spring/Fall	2003		133.76%
	2004		97.13%
Summer/Winter	2003		70.27%
	2004		95.45%

Spring/Fall is the consolidation element Spring divided by the consolidation element Fall. Similarly, Summer/Winter is the consolidation element Summer divided by the consolidation element Winter. The Seasons consolidation performs row level math that makes this report possible. Without consolidations, creating this analysis would be cumbersome.



Notice that elements are formatted as dollars (\$) and percentages (%) in the same consolidation. You can format individual consolidation elements in the Consolidation Editor. However, element value formatting cannot be overridden by report formatting. To apply report formatting on element values, element values must be set to the default formatting. For details on the interaction of formatting layers, see [Order of layers, page 322](#).

Consolidation elements

Consolidation elements are attribute elements that define the consolidation. Consolidation elements can also be an expression of attribute elements that make up a consolidation. They can be defined from any of the following:

- Elements from the same attribute, such as two cities
- Attribute elements from different levels of the same hierarchy, such as Region and Call Center in the Geography hierarchy
- Elements from attributes in different hierarchies, such as Country (from the Geography hierarchy) and Year (from the Time hierarchy)

- Existing consolidation elements, such as the ratio of Spring and Summer sales to Fall and Winter sales
- Elements from any other consolidation in the project (elements imported from an existing consolidation into another one)

You can combine the elements with simple mathematical expressions. For example, you can have an expression that adds attribute elements together, such as combining June, July, and August to get a Summer consolidation element. A consolidation element can also contain the logical operator AND. The following example demonstrates the use of mathematical expressions (addition and subtraction) and the AND operator.

Example of AND used in a consolidation element expression

You must report on the difference in revenues between the USA and Web for the winter of 2002. Create the following consolidation elements:

- USA: Winter 2002

$(\{\text{Month}=\text{Jan } 02 \text{ AND Country}=\text{USA}\} + \{\text{Month}=\text{Feb } 02 \text{ AND Country}=\text{USA}\} + \{\text{Month}=\text{Mar } 02 \text{ AND Country}=\text{USA}\})$

- Web: Winter 2002

$(\{\text{Month}=\text{Jan } 02 \text{ AND Country}=\text{Web}\} + \{\text{Month}=\text{Feb } 02 \text{ AND Country}=\text{Web}\} + \{\text{Month}=\text{Mar } 02 \text{ AND Country}=\text{Web}\})$



You cannot type AND into the expression. You must drag and drop an attribute element into the expression to trigger the AND operator. For more details, see the online help.

Create a consolidation element that uses the above elements to calculate the difference:

- USA - Web: Winter 2002

$([\text{USA: Winter 2002}] - [\text{Web: Winter 2002}])$

Finally, create a report with this consolidation and the Revenue metric. The report looks like the following:

Winter 2002: USA & Web	Metrics	Revenue
USA: Winter 2002		\$1,716,103
Web: Winter 2002		\$168,341
USA - Web: Winter 2002		\$1,547,762

Elements from the same attribute

A consolidation can contain elements from the same attribute, such as (March) and (April), both elements of the attribute Month of Year. With reference to the previous example, consolidation elements allow you to expand the consolidation to see the values for each month. For example, using elements from the same attribute, you can modify the report result set as follows by adding the following three elements to the consolidation:

- Element 1 (March)

Month of Year=March

- Element 2 (April)

Month of Year=April

- Element 3 (March-April)

{March}-{April}

With the use of consolidation elements, the report can now display the following.

	Metrics	Dollar Sales	Profit
Top Months			
March		\$ 27,545.00	\$ 4,940
April		\$ 11,502.00	\$ 1,957
March-April		\$ 16,043.00	\$ 2,983

A consolidation can contain any expression on the pairs of elements, such as (March - April). Using another example, an element expression can also be [DC, 2002] / [DC, 2003].

Elements from different levels of the same hierarchy

A consolidation can contain elements from different levels within the same hierarchy, such as Item and Subcategory from the Products hierarchy. For example, you may want to compare the contribution of different items to the Subcategory sales. Your consolidation, for the items Chocolate Roses and Chocolate Spoons, looks like:

- Element 1 (Roses percent)

[{Item=Chocolate Roses} / {Subcategory=Chocolate}]

- Element 2 (Spoons percent)

[{Item=Chocolate Spoons} / {Subcategory=Chocolate}]

With the use of consolidation elements, the report displays the following.

	Metrics	Dollar Sales
Chocolate percentages		
Roses percent		2.13%
Spoons percent		10.66%

Elements from attributes in different hierarchies

A consolidation element can contain elements from attributes in different hierarchies. For example, you can calculate the difference between two different regions for a particular month. For the months March and April, the consolidation could contain the following elements:

- Element 1 (March Southeast - Southwest)
 [Month of Year=March AND Region=South-East]
 - [Month of Year=March AND Region=South-West]
- Element 2 (April Southeast - Southwest)
 [Month of Year=April AND Region=South-East]
 - [Month of Year=April AND Region=South-West]

The report now appears as follows:

	Metrics	Dollar Sales	Profit
Region and Month			
March Southeast - Southwest		\$ 1,659.00	\$ 216
April Southeast - Southwest		\$ 43.00	-\$ 15

Existing elements

Using existing elements allows you to perform row level math, as described previously. For an example, see [Perform row level math, page 208](#).

Importing elements from other consolidations

You can import consolidation elements from an existing consolidation. When a consolidation element is imported, a new consolidation element is created and embedded into the consolidation.

Evaluation order

If you want to place two or more consolidations on a report, the order the engine evaluates them is significant and can change your result set. If one of the consolidations involves multiplication or division and the other involves addition or subtraction, which consolidation is calculated first matters. When performing a mathematical calculation, the product of a sum is not always equal to the sum of the product.

For example, a report contains the Dollar Sales metric and two consolidations. One consolidation is Seasons, as discussed in the previous examples. The other is called Years and is composed of three elements: 2002, 2003, and 2002/2003. The row for Spring 2002/2003 can be calculated either as (March 2002 + April 2002 + May 2002) / (March 2003 + April 2003 + May 2003) or as (March 2002 / March 2003) + (April 2002 / April 2003) + (May 2002 / May 2003). When the first calculation is used, so that the Seasons consolidation is evaluated first, the following report results.

Years	Seasons	Metrics	Dollar Sales
2002	Spring		\$2,605,000.00
	Summer		\$1,784,523.00
	Fall		\$1,947,487.00
	Winter		\$2,539,620.00
2003	Spring		\$2,041,987.00
	Summer		\$1,912,180.00
	Fall		\$2,102,240.00
	Winter		\$2,003,397.00
2002/2003	Spring		\$1.28
	Summer		\$0.93
	Fall		\$0.93
	Winter		\$1.27

When the second calculation is used, so that the Years consolidation is evaluated first, the following report results. Notice the difference in the 2002/2003 rows.

Years	Seasons	Metrics	Dollar Sales
2002	Spring		\$2,605,000.00
	Summer		\$1,784,523.00
	Fall		\$1,947,487.00
	Winter		\$2,539,620.00
2003	Spring		\$2,041,987.00
	Summer		\$1,912,180.00
	Fall		\$2,102,240.00
	Winter		\$2,003,397.00
2002/2003	Spring		\$3.97
	Summer		\$2.79
	Fall		\$2.83
	Winter		\$4.21

The evaluation order is set in the Report Data Options dialog box of the Report Editor when you create the report. To access this dialog box, select **Report Data Options** from the **Data** menu.

Consolidations and SQL

The calculations associated with a consolidation are done by the Analytical Engine component of Intelligence Server. The SQL Engine writes the SQL query that gets the required data from the warehouse, and then passes it to the Analytical Engine to do any mathematical operation that is needed to create the report.

For example, the following SQL is for the dollar sales by season report discussed in [Create a virtual attribute, page 207](#).



Notice that the seasons are not mentioned in the SQL. The query retrieves the data for the Months of Year, and then the Analytical Engine performs the necessary calculations to present the data in terms of seasons.

```
select a12.[MONTH_OF_YEAR] AS MONTH_OF_YEAR,
max(a13.[MONTH_OF_YEAR_NAME]) AS MONTH_OF_YEAR_NAME,
a12.[YEAR_ID] AS YEAR_ID, sum(a11.[TOT_DOLLAR_SALES]) as
DOLLARSALES
```

```
from [MNTH_CATEGORY_SLS] a11, [LU_MONTH] a12,
[LU_MONTH_OF_YEAR] a13
```

```

where a11.[MONTH_ID] = a12.[MONTH_ID] AND
a12.[MONTH_OF_YEAR] = a13.[MONTH_OF_YEAR] AND
a12.[MONTH_OF_YEAR] in (3, 4, 5, 6, 7, 9, 10, 11, 12, 1,
2)

group by a12.[MONTH_OF_YEAR], a12.[YEAR_ID]

```

Consolidations and subtotals

By default, subtotals for consolidations are disabled. This means that while you can subtotal a consolidation on a report, you cannot subtotal an object placed to the left of the consolidation on the report. Therefore, any subtotal which is defined across the consolidation and its higher levels is not shown on the report.

For example, the following report contains Year, Month of Year, Category, and the Revenue metric, in that order from left to right. (Only a portion of the report is included.) Notice that a consolidation is not included.

Year	Month of Year	Category	Metrics	Revenue
2005	January	Books		\$71,125
		Electronics		\$1,009,714
		Movies		\$49,751
		Music		\$39,332
	February	Books		\$78,374
		Electronics		\$1,142,328
		Movies		\$56,351
		Music		\$42,169
	March	Books		\$84,532
		Electronics		\$1,177,629
		Movies		\$60,778
		Music		\$44,716
	April	Books		\$77,250
		Electronics		\$1,107,233
		Movies		\$56,473
		Music		\$44,726
	May	Books		\$95,050
		Electronics		\$1,277,494
		Movies		\$69,325
		Music		\$53,376
	June	Books		\$88,444
		Electronics		\$1,262,867
		Movies		\$61,956
		Music		\$49,134

If you add grand totals, a total is calculated for the entire report, as shown in the portion of the report displayed below. (Add grand totals by pressing F11.)

2007	October	Books	\$128,717
		Electronics	\$2,001,325
		Movies	\$112,959
		Music	\$79,307
	November	Books	\$143,515
		Electronics	\$2,141,627
		Movies	\$127,415
		Music	\$88,826
	December	Books	\$147,331
		Electronics	\$2,385,623
		Movies	\$132,808
		Music	\$92,164
Total			\$68,366,469

If you add subtotals, a monthly total, yearly total, and grand total are displayed. (Add subtotals by selecting **Total** on the Subtotals dialog box.) Selected portions of the report are included below to show monthly and yearly totals.

Year	Month of Year	Category	Metrics	Revenue
2005	January	Books		\$71,125
		Electronics		\$1,009,714
		Movies		\$49,751
		Music		\$39,332
		Total		\$1,169,922
	February	Books		\$78,374
		Electronics		\$1,142,328
		Movies		\$56,351
		Music		\$42,169
		Total		\$1,319,222
	November	Books		\$116,039
		Electronics		\$1,612,361
		Movies		\$89,350
		Music		\$65,366
		Total		\$1,883,116
December		Books		\$106,075
		Electronics		\$1,553,682
		Movies		\$78,578
		Music		\$60,022
		Total		\$1,798,356
Total				\$18,474,075

If you subtotal across level, you can choose Year, Month of Year, and/or Category. Subtotaling across levels means to group by attributes to the left of the selected attribute. In the report below, Category is selected as the subtotal level, so the subtotals are grouped by month. Selected portions of

the report are included to show that monthly totals are calculated, but not yearly totals.

Year	Month of Year	Category	Metrics	Revenue
2005	January	Books		\$71,125
		Electronics		\$1,009,714
		Movies		\$49,751
		Music		\$39,332
		Total		\$1,169,922
	February	Books		\$78,374
		Electronics		\$1,142,328
		Movies		\$56,351
		Music		\$42,169
		Total		\$1,319,222
	November	Books		\$116,039
		Electronics		\$1,612,361
		Movies		\$89,350
		Music		\$65,366
		Total		\$1,883,116
	December	Books		\$106,075
		Electronics		\$1,553,682
		Movies		\$78,578
		Music		\$60,022
		Total		\$1,798,356



For more information on subtotal across level, see [Totaling and subtotaling data on reports, page 288](#).

In contrast, a similar report contains Year, a Seasons consolidation, Category, and the Revenue metric, in that order from left to right.

Year	Seasons	Category	Metrics	Revenue
2005	Winter	Books		\$255,574
		Electronics		\$3,705,724
		Movies		\$184,680
		Music		\$141,523
	Spring	Books		\$256,832
		Electronics		\$3,562,357
		Movies		\$186,576
		Music		\$142,818
	Summer	Books		\$289,018
		Electronics		\$3,952,618
		Movies		\$205,770
		Music		\$157,372
	Fall	Books		\$338,938
		Electronics		\$4,663,195
		Movies		\$246,202
		Music		\$184,879

If you add grand totals to the report, no totals are displayed because the entire report is at a higher level than the consolidation. If you add subtotals, the only total displayed is for Season. This occurs because year and grand totals are at a higher level than the consolidation. Therefore, Year is to the left of Season on the report.

2007	Winter	Books	\$376,027
		Electronics	\$5,900,743
		Movies	\$329,255
		Music	\$236,239
		Total	\$6,842,265
	Spring	Books	\$370,661
		Electronics	\$5,563,632
		Movies	\$309,126
		Music	\$213,653
		Total	\$6,457,071
	Summer	Books	\$398,499
		Electronics	\$5,941,766
		Movies	\$332,935
		Music	\$234,434
		Total	\$6,907,634
	Fall	Books	\$418,100
		Electronics	\$6,247,890
		Movies	\$361,811
		Music	\$255,810
		Total	\$7,283,610

If you subtotal across level, you can only select Category. Year is displayed, but cannot be selected because it is to the left of Season, because Year is on a higher level than Season. Since subtotals are disabled for the consolidation, Season is not displayed. The report is subtotaled by season only, as shown below.

2007	Winter	Books	\$376,027
		Electronics	\$5,900,743
		Movies	\$329,255
		Music	\$236,239
		Total	\$6,842,265
	Spring	Books	\$370,661
		Electronics	\$5,563,632
		Movies	\$309,126
		Music	\$213,653
		Total	\$6,457,071
	Summer	Books	\$398,499
		Electronics	\$5,941,766
		Movies	\$332,935
		Music	\$234,434
		Total	\$6,907,634
	Fall	Books	\$418,100
		Electronics	\$6,247,890
		Movies	\$361,811
		Music	\$255,810
		Total	\$7,283,610

Enable subtotals for the Season consolidation and then re-execute the report. If you add grand totals, a total is calculated for the entire report. (The end of the report is shown below.)

2007	Winter	Books	\$376,027
		Electronics	\$5,900,743
		Movies	\$329,255
		Music	\$236,239
	Spring	Books	\$370,661
		Electronics	\$5,563,632
		Movies	\$309,126
		Music	\$213,653
	Summer	Books	\$398,499
		Electronics	\$5,941,766
		Movies	\$332,935
		Music	\$234,434
	Fall	Books	\$418,100
Electronics		\$6,247,890	
Movies		\$361,811	
	Music	\$255,810	
Total			\$68,366,469

If you add subtotals, a seasonal total, yearly total, and grand total are displayed. (Again, only the bottom of the report is displayed below.)

2007	Winter	Books	\$376,027
		Electronics	\$5,900,743
		Movies	\$329,255
		Music	\$236,239
		Total	\$6,842,265
	Spring	Books	\$370,661
		Electronics	\$5,563,632
		Movies	\$309,126
		Music	\$213,653
		Total	\$6,457,071
	Summer	Books	\$398,499
		Electronics	\$5,941,766
		Movies	\$332,935
		Music	\$234,434
		Total	\$6,907,634
	Fall	Books	\$418,100
		Electronics	\$6,247,890
		Movies	\$361,811
		Music	\$255,810
	Total	\$7,283,610	
	Total		\$27,490,580
Total		\$68,366,469	

If you subtotal across level, you can choose Year, Season, and/or Category. Subtotaling across levels means to group by attributes to the left of the selected attribute. In the report below, Category is selected as the subtotal

level, so the subtotal is grouped by season. Selected portions of the report are included to show that seasonal totals are calculated, but not yearly totals.

2007	Winter	Books	\$376,027
		Electronics	\$5,900,743
		Movies	\$329,255
		Music	\$236,239
		Total	\$6,842,265
	Spring	Books	\$370,661
		Electronics	\$5,563,632
		Movies	\$309,126
		Music	\$213,653
		Total	\$6,457,071
	Summer	Books	\$398,499
		Electronics	\$5,941,766
		Movies	\$332,935
		Music	\$234,434
		Total	\$6,907,634
	Fall	Books	\$418,100
		Electronics	\$6,247,890
		Movies	\$361,811
		Music	\$255,810
		Total	\$7,283,610

Consolidation example

Report requirement

You have been given a task to understand how products are performing in different sections, or territories, of the country and abroad. This will allow you insight into consumer buying patterns and offer guidance on establishing pricing strategies and promotions. You need to see the territories in the rows of your report and various metrics, such as sales, profit, and revenue in the columns.

How can you accomplish this?

Solution

A Territory attribute does not exist in your project. You must create one.

A consolidation allows you to group together various elements of the Region attribute into various territories and place them on your template. In this example, you need to break down the territories as follows:

- East = Mid-Atlantic, Northeast, Southeast
- Mid-America = Central, South
- West = Northwest, Southwest
- Foreign = Canada, England, France, Germany

These consolidations placed in the rows of your report allow the metrics for values to be added together for a specific territory. For example, the metric values for Northwest and Southwest will be added together to yield the value for West, and so on.

Custom group and consolidation comparison

Both consolidations and custom groups provide flexibility in reports, but the objects differ in their structure and use. The essential distinction is that consolidations work with attributes and custom groups use filters. Consolidations are groupings of attribute elements while custom groups are based on filter criteria. Custom groups are used to apply different filters to different rows of a report. Consolidations are used to create virtual attributes to allow reporting on attributes that do not exist in the data model. Finally, row level math can be performed with consolidations but not with custom groups.

Custom groups are more flexible than consolidations because you do not have to know much about your data to create filters for a custom group. In contrast, consolidations require that you know exactly which attribute elements to select when creating the consolidation. To continue with the examples from the previous sections, you create filters for the Store Inventory custom group, to group small stores with low inventory and large stores with low inventory. For the Seasons consolidations, you need to know the months that make up a season.

The following table outlines other differences between custom groups and consolidations. More information on each section follows the table.

Feature or Action	Custom Group	Consolidation
Arithmetic operations (row level math)	Not allowed	Allowed
Where and how data is calculated	Warehouse	Analytical Engine
SQL efficiency	Low	High
Re-using consolidation elements	No	Yes
Hierarchical display	Flexible and expandable	Fixed at element level only
Subtotals	Yes	Yes

Arithmetic operations support (row level math)

Arithmetic operations such as addition and division are not allowed in custom group definitions. However, complicated mathematical expressions can be created in consolidations using the following operators:

- +
- -
- *
- /
- ()

This means that row level math, or mathematical operations between elements, can be performed in consolidations but not in custom groups. Row level math is a powerful tool in reporting, since it allows rows to be specified by mathematical operations.

Where and how data is calculated

Although the Analytical Engine might be needed to resolve banding in a custom group, the final stage of calculation is always in the data warehouse. For consolidations, all necessary data is retrieved from the data warehouse and then the consolidations are created in the Analytical Engine. Therefore, consolidations can be assigned an evaluation order to provide more and

varied analyses. For more information on evaluation order, refer to [Evaluation order, page 213](#).

SQL efficiency

For each custom group element, there is at least one SQL pass. When the custom group element is expanded in the Custom Group Editor, up to three SQL passes must be made for each custom group element.

Since the Query Engine uses a smart algorithm to combine consolidation elements and determine the minimal number of SQL passes, only one SQL pass may be needed for all the consolidation elements.

Re-using consolidation elements

You cannot use existing custom group elements to build a new custom group element. You must always build a custom group from attributes. In contrast, you can build new consolidation elements based on existing consolidation elements, which is also called recursive definition.

Hierarchical display support

Each custom group element can be viewed as a set of smaller grouping elements, which can be repeatedly broken down until the actual items are reached. This functionality, along with the four different display modes, provides flexibility and deeper analysis. For more information on viewing custom groups in a hierarchical structure, see [Custom group structure on reports, page 191](#).

Consolidations are displayed only at the element level; you cannot expand the elements.

Subtotal support

Custom groups act like attributes when subtotals are included.

By default, subtotals for consolidations are disabled. This means that while you can subtotal a consolidation on a report, you cannot subtotal an object

placed to the left of the consolidation on the report. Any subtotal which is defined across the consolidation and its higher levels is not shown on the report. For more information on consolidations and subtotals, see [*Consolidations and subtotals, page 215*](#).

DRILL MAPS

Customizing Drill Paths

Introduction

Drill maps allow you to create fully customized drill paths that are available to your users while drilling on a report. By default, the paths available are based on the system hierarchy of the project. You can create custom drill maps that can override these defaults.

This chapter describes how a drill map works, how to create a drill map, and how it can affect what you see when drilling on a report. Sections include:

- [*What is drilling?, page 225*](#)
- [*Creating custom drill maps and paths, page 228*](#)
- [*Drill map association: Objects containing the drill map, page 238*](#)

What is drilling?

After executing a report in a MicroStrategy reporting environment, a user may need to execute another report based on the original report to get more detailed or supplemental information. For example, after looking at annual

sales of a certain city, she may want to look at the monthly sales for the same city. Alternatively, after noticing that a certain item had a very high profit margin, she may want to see if that is also true for the entire category of that item. Such actions where a user creates a related report based on an existing report are referred to as drilling.

Even though a report generated as a result of drilling is related to the original report, they are, in essence, two entirely different reports. This means that the two reports can be saved or changed independent of each other. The two reports are different either because they have different templates, different filters, or both.

To drill on a report, a user can right-click an object on the report and then select the direction and destination. However, he is restricted by any settings assigned to the object's drill map. To override the default settings for filters and parent, page-by, threshold, and subtotal display on the drilled-to report, a user can drill using the Drill dialog box, which is accessed from the **Drill** option on the **Data** menu. For drilling steps, see the *MicroStrategy Developer help*. For details and examples of these settings, see [Drill path settings, page 230](#).

Drill maps and drill paths

Drill maps determine the options available to an end user when drilling on a report. When a user right-clicks an object on a report and chooses the drill option, he is displaying drill maps.

When the drill hierarchies are created, a default drill map is created. If no drill hierarchies are created, the system hierarchy is used to create the default drill map. The drill map determines what options are available to users when they drill on a report object. These different options are referred to as *drill paths*, which include the destination of the drill. The destination can be an attribute, a consolidation, a hierarchy, or a template.



All projects must have a default drill map; you cannot delete it unless you first specify a new default drill map. The default project drill map is specified in the Project Configuration Editor.

If you try to delete the only default drill map for a project, a message indicates that other objects depend on it. When you search for dependent objects, none are found. The dependent object is the project itself.

In summary, a drill map determines what drill paths are available while drilling from a report object. By default, the drill paths available to a report object reflect exactly the drill hierarchies of the project.

Default drill paths: System hierarchy drill paths

Before customizing drilling options you need to understand how the default drill paths work.

An end user can drill from any object on a report, other than a simple metric. For example, drilling down from an attribute or a hierarchy allows a user to access other child attributes in the same hierarchy. Drilling from a consolidation allows access to the attributes that make up the consolidation. Note that by default in these types, drilling changes a report by navigating through the drill hierarchies and selecting another attribute to view. The original object is replaced with the one drilled to. Drilling on a compound metric allows a user to view the metrics that compose it.

Filters and drilling

How a report's filter is changed while drilling depends on what part of the original report is selected when the drill is performed. By default, if an attribute element on the original report is selected while drilling, then that attribute element is added to the new filter created for the drill. The filter from the original report (the drilled-from report) is carried over as well.

For example, a report lists revenue by state and contains a filter for the Electronics category. A user selects Virginia when he drills to Store. The resulting report contains Electronics revenue for Virginia stores only.

You can change this default behavior for a drill path in the Drill Map Editor and for a report in Report Data Options. For descriptions of the drill path settings that affect filtering, see [Filter interaction options, page 232](#).



The right-click menu provides two ways to drill:

- If a user right-clicks a header, a filter is not added to the drill.
- If a user right-clicks an attribute element, the filter is used.

Creating custom drill maps and paths

You can override the default drill map by creating your own custom drill maps and paths. Once you begin customizing a drill map for an object, none of the drill paths of the system hierarchy are available for drilling on that object. For example, before you create a drill map for the attribute Region, the default drill map is the system hierarchy, which allows drilling up to Country and down to Call Center. You create a drill map and add a drill path down to Employee. A user cannot drill to Country or Call Center from Region unless you add these attributes to the new drill map as well.

To create a custom drill path, use the Drill Map Editor to select a destination and drill path type, and define the options, as described in the following sections:

- [The destination of the drill, page 229](#)
- [Drill path types: Drilling up, down, across, or to a template, page 229](#)
- [Drill path settings, page 230](#)
 - [Drill priority, page 231](#)
 - [Filter interaction options, page 232](#)
 - [Keeping or removing the page by and base template, page 236](#)
 - [Displaying the drilled-on object and thresholds, page 236](#)
 - [Specifying the display mode and VLDB properties, page 237](#)

You can:

- Create a new drill map as a stand-alone object, which allows you to associate multiple objects with the same drill map. To do this, access the Drill Map Editor directly. From the MicroStrategy Developer, choose New, and then select Drill Map.
- Create or edit a drill map for a particular object, access the Drill Map Editor from that object. You can access it from an attribute, custom group, consolidation, template, or report. For steps, see the *MicroStrategy Developer help*.

The destination of the drill

The destination is the object which a user will drill to in the report. This can be any of the following:

- Attribute
- Consolidation
- Hierarchy
- Template

Every time that a user selects a template drill path, a new report with the selected template is generated.

If the drill path is set to a template, every time you use this drill path a new report with the selected template is generated.

- Another drill map

Use this as a shortcut to the drill paths of the selected drill map.

If drill map is not displayed as an drill path destination, select **View Drill Map Path List** from the **View** menu.

You can create multiple destinations for each drill path type, and you can create multiple drill paths in each drill map.

Drill path types: Drilling up, down, across, or to a template

A drill path can be one of the following types:

- **Up**—The destination can be any attribute, consolidation, custom group, or hierarchy, and does not have to be related to the original object. The destination is shown as part of the **Drill Up** menu when a user right-clicks and selects **Drill** in the report.
- **Down**—This is similar to Up, except that the destination is shown as part of the **Drill Down** menu when a user right-clicks and selects **Drill**.
- **Across**—This is also similar to Up, except that the destination is shown as part of the **Other Directions** menu when a user right-clicks and selects **Drill**.

- **Template**—This allows you to replace the template of the original report template with a completely different destination template. Select the template to use as the destination template.
- **Drill Map**—Use this as a shortcut to the drill paths of another drill map, by selecting an existing drill map to use as the destination. You cannot select an attribute, consolidation, custom group, or hierarchy as the destination.
 - The destinations of those drill paths are displayed along with the destinations that you have created. For example, you select a drill map that drills up to Brand. You already have a drill path up to Subcategory. When a user selects **Drill** and **Up**, both Brand and Subcategory are displayed.



If you select a drill map or hierarchy as the destination, you cannot define any settings, such as priority or page by display, for the drill path.

Drill path settings

The settings for a drill path are described in the following sections:

- [Drill path display name, description, and set, page 231](#)
- [Drill priority, page 231](#)
- [Filter interaction options, page 232](#)
- [Keeping or removing the page by and base template, page 236](#)
- [Displaying the drilled-on object and thresholds, page 236](#)
- [Specifying the display mode and VLDB properties, page 237](#)



If a drill map or hierarchy is the destination of a drill path, you cannot define any options for that drill path.

The drill path settings available in the Drill Map Editor can be overwritten either:

- At the report level (see the *Answering Questions about Data* chapter in the *Basic Reporting Guide* or the *MicroStrategy Developer help* for descriptions of the drilling options at the report level)
- When a user drills (if he uses the Drill dialog box)

Drill path display name, description, and set

The **Drill Path Display Name** is the name displayed in the right-click drill menu on the report. You can either:

- Use the object's name, by selecting the **Use name of object** check box.
- Type a display name in the box.

You can also include a description of the drill path, to help identify it to other report designers.

You can group drill paths together in the right-click drill menu by using the same **Set Name** for them. This is valid for all drill path types. Sets cannot cross drill types, so use them to group drill maps within a single drill type, such as Up. Once you create a Set name, it appears in the drop-down list of other drill paths.

Drill priority

The **Priority** option affects how the drill path is displayed in a report:

- **Low:** The drill path is available as a right-click menu option in a MicroStrategy Developer report. In a MicroStrategy Web report, this drill path is not available as a right-click menu option but can be accessed from the More Options link.
- **Medium:** The drill path is available as a right-click menu option in both MicroStrategy Developer and Web reports.
- **High:** The drill path is used as the default drill path in both MicroStrategy Developer and Web reports. It is still available as a right-click menu option.

When a user double-clicks an object on a MicroStrategy Developer report, the default drill path is used. In Web, if an object on a grid has a default drilling option, the elements of that object appear as hyperlinks on the grid. A user can click the hyperlink to drill on the elements.

To set a drill path as the default, assign its priority to High. Only one high priority drill path can exist in each drill map.

Filter interaction options

The following settings affect how the filter is manipulated:

- Apply user filter (see [User filter and report filter options, page 232](#))
- Apply report filter (see [User filter and report filter options, page 232](#))
- Include other filter (see [Merging or ignoring existing attribute qualifications: Include other filter setting, page 232](#))
- Apply additional filter (see [Adding another filter to the drilled-to report, page 234](#))

User filter and report filter options

The **Apply user filter** option determines whether the object that is drilled from filters the drilled-to report. The **Apply report filter** option determines whether the filter on the drilled-from report is applied to the drilled-to report. These options are not mutually exclusive. The examples in the list are based on a report that displays revenue by state and contains a filter for the Electronics category. Virginia is selected when the report is drilled to store.

- Apply both. This is the default. The drilled-to report contains Electronics revenue for Virginia stores only.
- Apply neither. The drilled-to report includes revenue, by city, for all categories and all states.
- Apply the user filter only. The drilled-to report displays Virginia revenue for all categories, listed by store.
- Apply the report filter only. The resulting report shows Electronics revenue by store for all states.

Merging or ignoring existing attribute qualifications: Include other filter setting

The **Include other filter** setting is related to the report filter's advanced option. Both determine whether existing attribute qualifications are merged when the filter is evaluated. The report filter setting affects the entire report, while the Drill Map Editor setting applies only when a user drills on the report. For a description of the report filter setting, including examples, see [Merging attribute qualifications, page 151](#).

The options for the Include other filter setting are:

- To use the report filter's setting, select **Default**.
- To include other qualifications in the drilled-to report, regardless of the report filter setting, select **Yes**.
- To ignore other qualifications in the drilled-to report, regardless of the report filter setting, select **No**.



The Include other filter setting is available only if both the Apply user filter setting and Apply report filter setting are set to True.

For example, the following report contains a metric qualification for the top three revenue-producing regions. The metric qualification merges the qualifications when the filter is evaluated. This is the default setting.

Customer Region	Metrics	Revenue
Mid-Atlantic		\$ 3,510,407
Central		\$ 4,061,433
Southwest		\$ 3,552,846

Drill down on the Central region to **Customer City**. The report shown below is displayed.

Customer City	Metrics	Revenue
Chicago		\$ 290,346
Madison		\$ 89,183
Minneapolis		\$ 98,775

The top three revenue-producing cities in the Central region are selected and displayed. The qualifications were merged to produce this result, since by default the drill map uses the report filter's merge attribute setting. In this case, it is the same as setting the Include other filter option to Yes in the Drill Map Editor.

Return to the original report, edit the drill map, and change the **Include other filter** setting to **No**. Again drill down on the Central region to **Customer City**. The following report is displayed:

Customer City	Metrics	Revenue
Chicago		\$ 290,346

Only one city is displayed because the qualifications are not merged. First, the top three revenue-producing cities are identified, regardless of region. Then the drill to the Central region is applied to just those cities. Only one city, Chicago, of the three is in the Central region, so only that city is displayed on the final report.

Adding another filter to the drilled-to report

You can add another filter to the drilled-to report, to further refine the results.

For example, the following report does not contain any filters (only a portion of the report is shown):

Report details					
Report Filter: Empty Filter					
Region	Employee	Metrics	Revenue	Cost	Profit
Central	Ellerkamp Nancy		\$847,227	\$720,449	\$126,778
	Gale Loren		\$1,669,290	\$1,416,036	\$253,254
	Torrison Mary		\$1,690,350	\$1,430,865	\$259,485
	Zemlicka George		\$822,500	\$697,693	\$124,807
Mid-Atlantic	Bernstein Lawrence		\$1,060,632	\$901,702	\$158,930
	Brown Vernon		\$331,735	\$280,504	\$51,231
	Corcoran Peter		\$325,147	\$275,752	\$49,395
	Folks Adrienne		\$1,047,776	\$888,702	\$159,074
	Hollywood Robert		\$1,026,874	\$871,679	\$155,195
	Ingles Walter		\$229,439	\$194,851	\$34,588
	Smith Thomas		\$221,379	\$188,010	\$33,368
	Young Sarah		\$209,634	\$178,331	\$31,303
Northeast	De Le Torre Sandra		\$607,895	\$514,795	\$93,100
	Kelly Laura		\$2,350,720	\$1,992,726	\$357,994
	Kieferson Jack		\$584,933	\$497,463	\$87,470

Drill down on Central and Mid-Atlantic to Call Center. The default Tutorial Standard Drill Map is used, which applies the user filter (the objects selected to drill on) to the drilled-to report. The Call Centers in the Central and Mid-Atlantic regions are displayed, as shown below:

Report details					
Report Filter: Region = Central, Mid-Atlantic					
Call Center	Employee	Metrics	Revenue	Cost	Profit
Washington, DC	Bernstein Lawrence		\$1,060,632	\$901,702	\$158,930
	Folks Adrienne		\$1,047,776	\$888,702	\$159,074
	Hollywood Robert		\$1,026,874	\$871,679	\$155,195
Milwaukee	Gale Loren		\$1,669,290	\$1,416,036	\$253,254
	Torrison Mary		\$1,690,350	\$1,430,865	\$259,485
	Zemlicka George		\$822,500	\$697,693	\$124,807
Fargo	Ellerkamp Nancy		\$847,227	\$720,449	\$126,778
Charleston	Brown Vernon		\$331,735	\$280,504	\$51,231
	Corcoran Peter		\$325,147	\$275,752	\$49,395
	Ingles Walter		\$229,439	\$194,851	\$34,588
	Smith Thomas		\$221,379	\$188,010	\$33,368
	Young Sarah		\$209,634	\$178,331	\$31,303

Return to the original report, and create a new drill path to drill down from Region to Call Center. Add a filter to the drill path, by clicking ... (the browse button) in the **Apply additional filter** field. Select the Top 5 Employees by Revenue filter. Save the drill path.

Now when you drill down on Central and Mid-Atlantic to Call Center, only five employees are displayed—the top five revenue-producing employees, as shown below:

Report details					
Report Filter: (Region = Central, Mid-Atlantic) And (Rank of Revenue Top 5)					
Call Center	Employee	Metrics	Revenue	Cost	Profit
Washington, DC	Bernstein Lawrence		\$1,060,632	\$901,702	\$158,930
	Folks Adrienne		\$1,047,776	\$888,702	\$159,074
	Hollywood Robert		\$1,026,874	\$871,679	\$155,195
Milwaukee	Gale Loren		\$1,669,290	\$1,416,036	\$253,254
	Torrison Mary		\$1,690,350	\$1,430,865	\$259,485

Keeping or removing the page by and base template

You can choose to keep or remove the page by and the base template in the drilled-to report.

- When a report contains a very large set of data, it can be easier to handle that data by grouping the report data into logical subsets, and viewing only one of the subsets at a time. This grouping is called page-by. For an example, see the *Basic Reporting Guide*.

You can choose whether or not to page the drilled-to report. By default, the page by is not included on the drilled-to report.

For example, a report displays regional revenue and is paged by year. When a user drills from region to the Call Center Analysis template, the template is not paged by year. Change the Keep page by setting to True. Now when a user drills from region to the Call Center Analysis template, the template is paged by year.

- If you keep the base template, the objects on the report being drilled from are copied to the report objects of the template being drilled to.

Displaying the drilled-on object and thresholds



These settings are available only if the drill path destination is not a template.

The **Keep parent** setting determines whether or not the original object (the object selected when the user drills) appears in the drilled-to report. By default, this setting is not selected.



The Keep parent setting is overridden by the Keep parent while drilling option set in Report Data Options. The Keep parent while drilling option allows you to specify whether or not to keep the parent for all drills in the report. To specify the parent setting at the drill path level, set **Keep parent while drilling** to **Default**. For steps, see the *MicroStrategy Developer help*.

For example, a report contains Region, Employee, and Revenue. A drill path is created that drills down from Region to Call Center. The parent object is

not kept. When you drill from the Central region, the drilled-to report displays Call Center, Employee, and Revenue, as shown below:

Call Center	Employee	Metrics	Revenue
Milwaukee	Gale	Loren	\$1,669,290
	Torrison	Mary	\$1,690,350
	Zemlicka	George	\$822,500
Fargo	Ellerkamp	Nancy	\$847,227

Change the drill path to keep the parent. Now when you drill from the Central region, the Region is displayed as well:

Region	Call Center	Employee	Metrics	Revenue
Central	Milwaukee	Gale	Loren	\$1,669,290
		Torrison	Mary	\$1,690,350
		Zemlicka	George	\$822,500
	Fargo	Ellerkamp	Nancy	\$847,227

The **Keep thresholds** setting determines whether or not thresholds are displayed in the drilled-to report. By default, this setting is not selected.



The Keep thresholds setting is overridden by the Keep thresholds while drilling option set in Report Data Options. The Keep thresholds while drilling option allows you to specify whether or not to display thresholds for all drills in the report. To specify the thresholds setting at the drill path level, set **Keep thresholds while drilling** to **Default**. For steps, see the *MicroStrategy Developer help*.

Specifying the display mode and VLDB properties



These settings are available only if the drill path destination is a template.

The **Display mode** setting determines how the drilled-to report is displayed, as one of the following options:

- **Default:** Uses the original report's display setting.
- **Grid:** Displays the drilled-to report as a grid.
- **Graph:** Displays the drilled-to report as a graph.
- **Grid graph:** Displays the drilled-to report as both a grid and a graph.

The **Inherit VLDB** property determines whether or not the VLDB properties of the original report are applied to the drilled-to report.

Drill map association: Objects containing the drill map

The drill map *association* defines which grid unit uses this drill map. In other words, drilling uses the object's associated drill map. An object can have an association both on the object level and on each template/report. If an association does not exist on the template/report level, then the association on the object level is used when a user drills at that object.

If an object is already associated with a drill map, that drill map is displayed in the Drill Map Editor. Otherwise, the default drill map is based on the system hierarchy. Once you begin to modify the default, it is no longer the system hierarchy—the name changes automatically, although you can edit this preset name. When you save your changes, a new drill map is created.

You can create and edit a drill map as a stand-alone object from the MicroStrategy Developer. You can associate multiple objects with the same drill map, using the **Associate with** button. The associated objects appear in the Origin list on the interface.

You can also access the Drill Map Editor from the following:

- Attribute Editor
- Consolidation Editor
- Report Editor
- Template Editor

When you edit a drill map from another editor, the drill map is associated with the object selected in the other editor. You cannot change the drill map association (the other objects that are associated with this drill map), but you can change which drill maps are associated with the selected object. For example, if you are editing the Store State attribute and you access the Drill Map Editor, only Store State is associated with the drill map you create.

If the original object is a template or report, the children objects are also available. For example, a sales report contains Store, Store State, Item, and

the Revenue metric. You can create a drill map for each of the grid units except Revenue, because you cannot create drill maps for metrics.

Order of precedence of drill maps: Levels

When you change or customize the drill map associated with a grid unit, you can do so at several different levels:

- **Project level**—If a drill map is associated with grid units at the project level, then all of the grid units in the project will have this drill map. Therefore, when you drill on a report, the default drill paths are those specified in this drill map. This option is found in the Project Configuration Editor.
- **Grid unit level**—A drill map can be associated with individual grid units such as attributes, consolidations, and hierarchies. When the object is used in a report or template, the grid unit level drill map overrides the project level drill map.
- **Template level**—If a drill map is associated with grid units on a particular template, it overrides the project level and grid unit level drill maps. The drill paths of this drill map are available in all reports that use this template.
- **Report level**—If a drill map is associated with grid units on a report level, it overrides the drill maps defined at the project level, grid unit level, and the template level. If a grid unit is not associated with a drill map at the report level, it inherits the map from the report template. If it is not associated with a drill map through the template, then the grid unit drill map is used, and so on.

The Drill Map Editor represents these levels and inheritances when you edit a report's drill maps. If the Name field is greyed out (disabled), the selected report object inherited that drill map from the project, grid unit, or template. When you overwrite it by adding a different drill map to the object, the Name field is enabled.

For example, you create a drill map named Customer Attribute Drills for the attribute Customer. Create a report named Customer Revenue that displays Region, Customer, and the Revenue metric. When you edit the drill maps for the report and select Customer, the Name field is disabled but displays Customer Attribute Drills. No drill paths are displayed for the drill map. Because this attribute does not have a drill map on the report, it inherited the drill map from the attribute level. Because you cannot edit the attribute's drill map from the attribute on the report, the Name field is disabled and the

drill paths do not appear. When a new drill map is created for Customer on this particular report, Name is enabled, defaulting to Customer Sales Customer Drill Map.

By default, there is a project drill map containing all of the hierarchies that have been specified as drill hierarchies using the Hierarchy Editor in the project. It cannot be deleted, but it can be modified and overridden.

Drill maps associated with templates

You can also change or customize the drill map associated with a template, rather than the individual grid units on the template. The drill map then controls the drill paths only for the metric values on the template.

For example, the Basic Report contains regions, employees, and the Revenue metric. The drill map contains a drill path to a Personnel report. This drill path is associated with the template, not Region, Employee, or Revenue. If you drill on a number in the Revenue column of the Basic Report, you can drill to the Personnel report only. If you drill on the Revenue metric heading, you can drill in other directions to various hierarchies. If you drill on a region, you can drill up to Country, down to Call Center, or in other directions.

The drill paths of this drill map are available in all reports that use this template.

Removing associations

The **Remove Association** button disassociates the object from the current drill map and replaces it with its default map. Depending on the levels described above, this default map could be the template drill map, the grid unit drill map, or the project drill map.

The **Clear All** button deletes all the drill path information for the whole drill map. The object effectively has no drilling options. **Reset** reverses any changes and resets the drill map to the condition of the last save. Drill map associations are reset as well.

ADVANCED PROMPTS

Enabling User Input

Introduction

A prompt is a MicroStrategy object that allows user interaction at report run time. The prompt object is incomplete by design. The user is asked during the report resolution phase of report execution to provide an answer to complete the information. For example, the user can enter information such as the region “Northeast” or year “2007,” and the data is returned from the data warehouse. With prompts you can create reports that allow users to change the report content at run time.

Prompts are useful for asking questions about the set of data you want to see in the report. Prompts allow the report to have dynamic report definitions, which can change with each query by altering the information in the prompt dialog box. Prompts can be a part of the report definition, filter, template, custom group, or metric.

There are benefits to using prompts as a part of these objects. Prompts are useful to keep the number of reports in a project small, as they allow users to define how they would like to view things in the report environment instead of the report designer providing a separate report for each user’s information needs.

You can find the report that you would like to see more quickly because there will be fewer options. However, prompting questions asked of the user raise the complexity of just running a report, and you run the risk of confusion. This confusion can be allayed by providing good descriptions for the prompts so that the user is clear on what questions they are answering.

What is a prompt?

By using a prompt in a filter or template, you can:

- Apply filtering conditions to the report, a custom group on the report, or a metric on the report.
- Choose what objects, such as attributes and metrics, are included in the report.

Prompts allow you to determine how the report returns data from the data warehouse. Prompts save you time. Instead of building several different reports that each answer a different, single question, a set of questions can be asked just before a prompted report is executed. There are many types of prompts, and each type allows a different type of question to be asked. These prompt types are discussed individually later in this chapter.

One type of question that can be asked is about the subset of data that should be included on the report. For example, to see a report about sales in a particular geographical region, you can build three reports— Sales in West Region, Sales in Central Region, and Sales in East Region. However, using a more efficient approach, you can create one report called “Sales, prompted on Region” that asks the user for the region to include.

A different type of question to ask is what attribute should be shown on the report. For example, Sales by Day, Week, Month, Quarter, or Year can be consolidated via prompting. This report asks the user by which time unit the report data should be displayed.

Prompts in scheduled reports

Reports are executed immediately after a request to execute a report is made. However, you can schedule a report to be executed on a predetermined schedule. After a scheduled report is executed, it is placed in the History List. For more information on scheduled reports and the History List, see the

Automating Administrative Tasks chapter of the *System Administration Guide*.

A scheduled report can contain prompts; how and if the report is executed depends on the prompt definition. The following table explains the various scenarios. Notice that to ensure that prompts are used on a scheduled report, the prompt must be required and default prompt answers set.

Prompt Required	Default Answer	Result
No	No	The prompt is ignored; the report is not filtered by the prompt.
No	Yes	The prompt and default answer are ignored since the prompt is not required; the report is not filtered by the prompt.
Yes	No	The report is not executed. No answer is provided to the required prompt so MicroStrategy cannot complete the report without user interaction.
Yes	Yes	The report is executed; the prompt is answered with the default.

Types of prompts

Using the following prompt types, you can create a prompt for nearly every part of a report. It is important to remember that prompts can be used in many objects including reports, filters, metrics, and custom groups. All of these prompts will reveal themselves at report execution time, but the origin of the prompt can be within any of the objects.

- The **filter definition prompt** encompasses four different prompt types, all of which allow the user to define the filtering criteria: attributes in a hierarchy, attribute forms, attribute element lists, and metrics.
- An **object prompt** allows you to select which MicroStrategy objects to include in a report, such as attributes, metrics, custom groups and so on. Object prompts can either determine the definition of the report template or the report filter.
- A **value prompt** allows you to select a single value such as a date, a specific number, or a specific text string. The value chosen by the user is compared to metric or attribute element values, and thus determines the data viewed by the user.

Prompts can also be used as part of a function expression, and value prompts are particularly suited to provide values for function arguments. For a description of how to use prompts in function expressions, see the *Functions Reference*.

- A **level prompt** allows you to specify the level of calculation for a metric.

This chapter provides merely a basic review of all these types, with the exception of level prompts. For details, examples, and procedures, see the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*. The *MicroStrategy Developer help* also contains detailed procedures to create the various types of prompts.



You can also create prompts for use with MicroStrategy Mobile. For examples and steps, see the *Mobile Design and Administration Guide*.

Filtering data on attributes, attribute forms, attribute elements, or metrics: Filter definition prompts

These prompts are used for qualifying on the value of attribute elements and metrics. The filters affected by these types of prompts can be in the report, in a filter (which in turn may be used in the conditionality of a metric in the report), or in an element of a custom group. Background information on custom groups may be found in [About custom groups: Grouping attribute elements with banding and filtering, page 182](#).

Choose from all attributes in a hierarchy

This type of prompt is used to qualify on one or more attributes in one or more hierarchies. At least one hierarchy and all the attributes in that hierarchy are displayed. The user can then qualify on one or more of those attributes by choosing an element list or by qualifying on a particular attribute form. The choices made are included in the filtering criteria in the report.

To define this type of prompt, you can do one of the following:

- Choose a particular hierarchy.
- Use the set of hierarchies resulting from a search for hierarchies.
- List all hierarchies available in the project.

If you choose to display more than one hierarchy, you can create qualifications from all hierarchies displayed at report run time.

Qualify on an attribute

This type of prompt is used to apply conditions or qualifications to an attribute form.

One or more attributes are displayed and the user can create qualifications on an element list or an attribute form of the selected attribute.

To define an attribute qualification prompt, you can either:

- Choose a particular attribute.
- Display a partial or complete list of attributes that is the result of a search for attributes available in the project.

Choose from an attribute element list

This option is used to allow the user to choose from a list of attribute elements to be included in a filter or custom group. This list may be restricted, at prompt design time. This type of prompt can be used with any attribute in a project.

The list of elements from which the user can choose can be implemented by:

- Selecting all elements associated with an attribute
- Providing a partial list of elements by applying a filter on all of the elements associated with an attribute
- Providing a predefined list of elements from which the user can choose

Qualify on a metric

A metric qualification prompt allows a user to qualify on a metric. One or more metrics are displayed and the user can choose a metric for qualification.

The choice of metrics can be defined by:

- Specifying a single metric for run-time use
- Specifying a search object to restrict the list of metrics from which the user can choose

Example: Filter definition prompt



For procedures to create filter definition prompts, see the *MicroStrategy Developer help* or the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*.

Report requirement

You need to create a report showing sales in the West, Central, and East Regions. All other data on the report remains the same. You do not necessarily want the regions on the same report.

Solution

To meet this requirement, the easiest solution is to create a report that includes a filter prompting the user on region. When the report is executed, the prompt dialog opens, asking the user to choose the region(s) for which to return the report results.

Filtering data based on metrics, attributes, or other objects: Object prompts

Object prompts are used in reports and documents to allow the user to choose what objects will be included in the report filter or on the report template.

These are defined by specifying either a search object or a predefined list of objects from which the user can choose. An object prompt allows specification of default answers as well as maximum and minimum number of objects to be selected.

A search object defines the criteria (such as location, date, owner, and so on) for a list of objects to be generated. Searches defined in prompts are saved in the project. For example, a search object can display all metrics that are contained in a certain folder and use a particular fact.

One prompt can contain different types of objects, such as both metrics and attributes, or attributes and custom groups. This reduces the number of prompts that are needed for a report. However, if the prompt mixes metrics

with another type of object, either the metrics or the other objects are removed when the prompt is executed in a report, as described below:

- If a metric is the first object in the list, all non-metric objects are removed.
- If a non-metric object is the first object in the list, all metrics are removed.

This ensures that only attributes or metrics are included on the rows of a report. Alternatively, you can create one object prompt for the metrics and another for the other types of objects. For an example, see [Examples: Object prompts, page 247](#).

Examples: Object prompts



For procedures to create object prompts, see the *MicroStrategy Developer help* or the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*.

Report requirement

Create a report displaying item sales. At run time, the user can select whether to calculate sales at the Category or the Subcategory level.

Solution

Create an object prompt with Category and Subcategory. Create the Sales metric, using the object prompt as the target of the level (dimensionality). When the report is executed, the user is prompted to select either Category or Subcategory. Based on the user's choice, the Sales metric is calculated accordingly.

Report requirement

Create a report displaying item sales. At run time, the user can select whether to calculate sales at the level of Category, Subcategory, or Age Groups.

Solution

Create an object prompt with the Category and Subcategory attributes, as well as the Age Groups custom group. Create the Sales metric, using the object prompt as the target of the level (dimensionality). When the report is executed, the user is prompted to select Category, Subcategory, or Age Groups. Based on the user's choice, the Sales metric is calculated accordingly.

Report requirement

The sales manager frequently asks her analysts to provide similar reports with minor changes to the metrics. She always wants a metric to calculate revenue for each employee. In addition, she sometimes wants to compare the results of each employee to the revenue results of the best or the worst employee. Other times, she wants to compare the results of each employee to the average revenue of all the employees.

Solution

Instead of creating many different reports, you can provide the sales manager with the flexibility to select the analytical function she wants at the time of running the report.

In this case you can give her three functions to choose from:

- Minimum
- Maximum
- Average

When she runs the report, she can select which function to use with the Revenue metric. Your final report can then display the following objects:

- Employee
- Revenue metric
- Revenue metric that uses the analytical function selected by the user

Value prompts

Value prompts are used when the information needed at run time is a single value of a specific data type. The value chosen by the user is compared with either an attribute form or a metric. This comparison can be done in a filtering criteria or in a custom group.

The different types of Value prompts are:

- **Date** prompts for a date value.
- **Numeric** prompts for a numeric value. Numeric value prompts accept integers or decimals up to 15 digits of precision.
 -  If a user enters more than 15 digits for a numeric prompt, the data is converted to scientific notation. If precision is needed beyond 15 digits, you should use a Big Decimal value prompt instead.
- **Text** prompts for any type of text.
- **Big Decimal** prompts for a big decimal value. Big Decimal value prompts accept integers and decimals up to 38 digits of precision.
 -  Big Decimal prompts should only be used in expressions that require high precision, such as qualifying on a Big Decimal attribute ID.
- **Long** prompts for a long integer value. Long prompts accept integer numbers up to 10 digits.

Although **long** prompts are not part of the options available by default for selection, you can enable them as part of your project preferences. To enable them:

- Double-click your project to open it.
- From the **Tools** menu in MicroStrategy Developer, select **My Preferences**. The My Preferences dialog box opens.
- Expand the **General** category on the left, and select **Prompts**.
- Select **Add long prompts to the list of available value prompts**.
- Click **OK** to save your changes.

Value prompts allow specification of maximum and minimum values to be applied.

Example: Value prompt



For procedures to create value prompts, see the *MicroStrategy Developer help* or the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*.

Report requirement

Create a report showing sales since a certain date.

Solution

Prompt the user for the date since they want to see sales data. The value they choose is applied to a filter criteria for the attribute Date. The prompt here is included in the filter on a report.

Example: Currency conversion

Report requirement

An international company is based in the United States, so the data is saved in dollars in the data warehouse. However, an end user is based in Europe and wants to see revenue in Euros. Create a report that converts revenue in dollars to Euros.

Solution

The report prompts the user for a conversion rate, which is applied to a revenue metric. If a user wants to see dollars, he can enter a conversion rate of one.

The high-level steps are:

- 1 Create a value prompt for the conversion rate.



The prompt's instructions can include a website to find today's conversion rate, and also direct the user to enter a conversion rate of one to keep the revenue in US dollars.

- 2 Use the conversion prompt in a metric. The metric's formula should look like the following, assuming revenue is the target fact and Euro Conversion Rate is the name of the prompt:

`(Sum(Revenue) {~} * ?[Euro Conversion Rate])`

- 3 Use the metric in the report.

In the US, the report is executed, and 1 is entered at the currency conversion prompt. The results are shown below:

Region	Metrics	Revenue
Central		\$5,029,366
Mid-Atlantic		\$4,452,615
Northeast		\$8,554,415
Northwest		\$1,761,187
South		\$5,389,280
Southeast		\$2,239,951
Southwest		\$3,694,132
Web		\$3,902,762

In Spain, the report is executed, and a conversion rate of .72263 is entered at the currency conversion prompt. The revenue data is multiplied by that number, and the results are shown below:

Región	Indicadores	Ingresos
Atlántico Central		€3.217.593,21
Central		€3.634.370,93
Noreste		€6.181.676,59
Noroeste		€1.272.686,71
Sur		€3.894.455,62
Sureste		€1.618.655,83
Suroeste		€2.669.490,39
Web		€2.820.252,94

Defining the dimensionality of metrics: Level prompts

A level prompt allows the user to select, during report execution, the level of calculation (or dimensionality) for a metric. For example, a report shows sales totals by time and allows the user to choose whether the totals are calculated for the year, the month, the week, and so on.



For background information on metric levels, see [Level metrics: Modifying the context of data calculations, page 14](#).

When the definition of multiple metrics would differ only in level, it is useful to create a level prompt to avoid having to create multiple metrics.

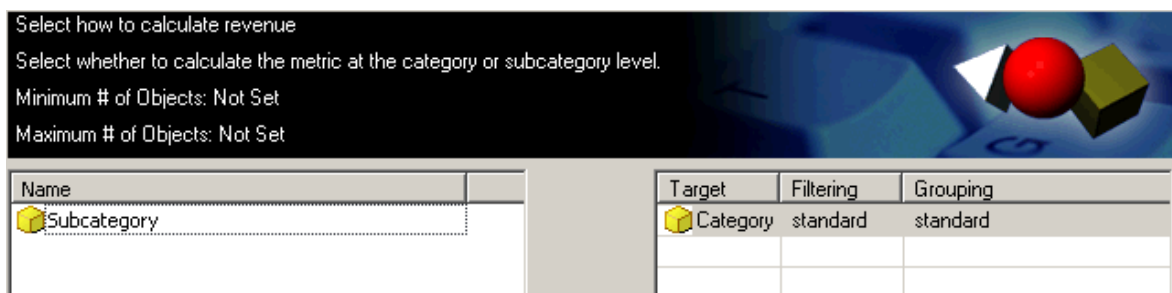
You can define a default prompt level, which is displayed when the prompt is executed. This allows users to complete report execution more quickly, since they do not need to answer the prompt, but can simply run the report using the default level. The user can select the default, or specify his own level. A default prompt level is particularly useful if a large percentage of your users will select the same level for this prompt. By default, the default level is the report level, with standard filtering and standard grouping. You can remove or edit this level, and add more levels.

 An object prompt allows prompting only on the target, while a level prompt allows prompting on all components of the metric—target, filtering, and grouping. For definitions of these components, see [Level metrics: Modifying the context of data calculations, page 14](#). For an example of an object prompt, see the *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide*.

Example: Level prompt

Report requirement

Create a report listing items with sales calculated at either the Category or the Subcategory level. The item's revenue is displayed as a percent of the Category or Subcategory revenue. At run time, the user must be able to select the Category or Subcategory level, as well as the filtering and grouping settings of the metric, as shown below:



Select how to calculate revenue

Select whether to calculate the metric at the category or subcategory level.

Minimum # of Objects: Not Set

Maximum # of Objects: Not Set

Name	Target	Filtering	Grouping
 Subcategory	 Category	standard	standard

When the prompt is answered with Category, as shown above, the report looks like the following. The Category/Subcategory Revenue amount is the same for each item; it changes only when the Category changes, as

determined by the prompt answer. The item revenue is divided by the category revenue, to yield the percentage.

Category	Item	Metrics	Revenue	Category/ Subcategory Revenue	Item as % of Cat/Subcat
Books	100 Places to Go While Still Young at Heart		\$67,993	\$2,640,094	2.58%
	Art As Experience		\$23,733	\$2,640,094	0.90%
	Reversing Diabetes		\$27,076	\$2,640,094	1.03%
	Self-Management and Efficiency		\$35,451	\$2,640,094	1.34%
Electronics	Harman Kardon Digital Surround Sound Receiver		\$745,650	\$24,391,303	3.06%
	Harman Kardon AM/FM Stereo Receiver		\$350,075	\$24,391,303	1.44%
	Harman Kardon Dolby Digital Receiver		\$514,045	\$24,391,303	2.11%
	GPX Portable CD Player with Bass Boost		\$44,523	\$24,391,303	0.18%



This report contains 360 rows, so only portions of the report have been reproduced above.

If the prompt is answered with Subcategory instead, the report looks like the following. Note that while the Revenue values are still the same, the Category/Subcategory Revenue amounts change when the Subcategory (not shown on the report) changes. The percentages are now greater, since the item revenue is compared to the subcategory revenue, not the larger category revenue.

Category	Item	Metrics	Revenue	Category/ Subcategory Revenue	Item as % of Cat/Subcat
Books	100 Places to Go While Still Young at Heart		\$67,993	\$480,173	14.16%
	Art As Experience		\$23,733	\$480,173	4.94%
	Reversing Diabetes		\$27,076	\$335,106	8.08%
	Self-Management and Efficiency		\$35,451	\$400,871	8.84%
Electronics	Harman Kardon Digital Surround Sound Receiver		\$745,650	\$3,782,832	19.71%
	Harman Kardon AM/FM Stereo Receiver		\$350,075	\$3,782,832	9.25%
	Harman Kardon Dolby Digital Receiver		\$514,045	\$3,782,832	13.59%
	GPX Portable CD Player with Bass Boost		\$44,523	\$3,782,832	1.18%



This report contains 360 rows, so only portions of the report have been reproduced above.

Solution

Create a level prompt with Category and Subcategory. Create a revenue metric called Category/Subcategory Revenue, using the level prompt as the target of the level (dimensionality). When the report is executed, the user is prompted to specify the complete level of the metric, including the target, filtering, and grouping. The Category/Subcategory Revenue metric is then calculated accordingly.



This example differs from the one in [Examples: Object prompts, page 247](#). An object prompt allows prompting only on the target, while the level prompt allows prompting on all components of the metric—target, filtering, and grouping.

Creating a level prompt

To create a level prompt

- 1 In MicroStrategy Developer, on the **File** menu, point to **New**, and then choose **Prompt**. The Prompt Generation Wizard opens.
- 2 Select the **Level prompt** option and click **Next**.

To define the prompt type to be displayed to the user

- 3 Select the attributes and/or hierarchies available to the user to determine the metric level at run time. Do one of the following:

- To choose specific objects, complete the following steps:
 - a Select **Use a pre-defined list of objects**.
 - b Click **Add**. The Select Objects dialog box opens.
 - c Select the objects from the **Available objects** list and click **>** to move them to the **Selected objects** list.



You can choose multiple objects by pressing **CTRL** or **Shift** while selecting the objects.

- d Click **OK** to return to the Prompt Generation Wizard.
- e You can add more objects by clicking **Modify**, which opens the Select Objects dialog box so you can select additional objects.

- f To remove the selected object, click **Remove**.
- g To delete all the objects in the list, click **Clear**.
- h To change the order that the objects are displayed in the prompt, select the object to move and click the **Up** or **Down** arrow.
- To use a set of objects from a search object, select **Use the results of a search object**. To display the object path (folder hierarchy) when using a search, select **Display using folder structure**, which is useful when the same object with the same name is saved in multiple folders. Do one of the following:
 - To use an existing search, click ... (the browse button). Navigate to the search object, select it, and click **OK** to return to the Prompt Generation Wizard.
 - To create a new search, click **New**. The Search for Objects dialog box opens. Create a new search.
- To allow users to choose from all the attributes and hierarchies available in the project, select **List all attributes and hierarchies (no restriction)**.

To specify a title and instructions

Think about a name and instructions carefully, with your users in mind. The title and instructions that you provide for a prompt can make the difference between users finding prompted report execution confusing and users completing rapid report execution that displays exactly the data they want to see. For examples of effective and ineffective titles and instructions, see the *Building Query Objects and Queries, for Designers* chapter in the *Basic Reporting Guide*.

- 4 Click **Next**. The General Information page opens.
- 5 Enter a **Title**, which is used as the default object name when you save the prompt, although you can change it.
- 6 Type text in the **Instructions** field, which is displayed when the prompt is run during report execution.

To allow personal answers

Personal answers allow a user to save prompt answers for this prompt, and then reuse the answers on any report that this prompt is used on. For more information on personal answers and how they can be used, see the *Building Query Objects and Queries, for Designers* chapter in the *Basic Reporting Guide*.

- 7 To determine whether personal answers can be saved for this prompt, select one of the following options from the **Personal answers allowed** drop-down list:
 - **None:** No personal answers can be saved. Every time a user sees the prompt, he must answer it manually (if it is required).
 - **Single:** Only one personal answer can be saved for this prompt. When the prompt is used again (on this report or a different one), the personal answer is displayed. A user can keep the personal answer, or add or delete selections. He can save his changes as a new personal answer, but only one personal answer can be saved for the prompt.
 - **Multiple:** Multiple personal answers can be named and saved, allowing different answers for the same prompt. When the prompt is used again (on this report or a different one), the personal answers are available. The user can select one of them, or answer the prompt manually.

To define a range within which the user's answer must fall

- 8 To restrict the user to entering values within certain ranges (optional):
 - Select the **Minimum objects** check box and enter the lowest number of objects allowed in the prompt answer.
 - Select the **Maximum objects** check box and enter the highest number of objects allowed for the prompt answer.
- 9 If you choose, you can select the **Prompt answer required** check box to require users to answer the prompt before running the report. For considerations about required and optional prompt answers, see the *Building Query Objects and Queries, for Designers* chapter in the *Basic Reporting Guide*.
- 10 Click **Next**. The Default Prompt Level Selection page opens.

To specify a default prompt level

A default prompt level can allow users to complete report execution more quickly, since they do not need to answer the prompt, but can simply run the report using the default level. The levels that you define here are displayed when the prompt is executed. The user can select the default, or

specify his own level. A default prompt level is particularly useful if a large percentage of your users will select the same level for this prompt.

By default, the default level is the report level, with standard filtering and standard grouping. You can remove or edit this level, and add more levels.

- 11 To add another level, click **Modify**. The Select Objects dialog box opens. Select the object(s) to use as the target(s) from the **Available objects** list and click > to move them to the **Selected objects** list. Click **OK** to return to the Prompt Generation Wizard. The objects are added to the default prompt answer table, with standard filtering and grouping.

The target is the attribute level at which the metric calculation groups. For more information, see [Target: The context of a calculation, page 22](#).

- 12 To change the filtering of a level, click in the **Filtering** column of the level and select the new filtering option.

Filtering governs how the report filter interacts with the metric calculation. For more information, see [Filtering: Interaction with report filters, page 30](#).

- 13 To change the grouping of a level, click in the **Grouping** column of the level and select the new grouping option.

Grouping determines how the metric aggregates. For more information, see [Grouping: How aggregation is performed, page 23](#).

- 14 To delete the selected level, click **Remove**.

- 15 To return to the default report level and delete any additional levels, click **Reset**.

- 16 By default, the metric filter is applied to the metric calculation. You can instead exclude filter attributes, so that filter attributes that are not on the report or in the level of the metric are not included in the metric calculation. For more information on this behavior, see [Level metrics: Applying the metric condition to the metric calculation, page 53](#).

- To change this behavior, click **Advanced**, set **Filter setting: uncheck to exclude attributes absent in report or level (dimensionality)**, and click **OK** to return to the Prompt Generation Wizard.

- 17 Click **Finish**. The Save As dialog box opens.

To save your prompt

- 18 Select the folder in which to save the new prompt, enter a name, and click **Save** to return to MicroStrategy Developer.

All level prompts must be added to a metric, and then the metric is added to a report.

Prompting for a user's login name: System prompts

System prompts are built-in prompts that are created when the project is created. System prompts are created in the System prompts sub folder in the Prompts folder of MicroStrategy Developer.

The User Login system prompt is a special prompt that returns the current user's user login. The user is not prompted to provide an answer to the User Login prompt, as the Intelligence Server automatically answers the prompt with the user's login when prompts are resolved. The User Login system prompt can be used in conditions in which a text value prompt is used in a report.

See the *System Administration Guide* for details on working with system prompts.

The system prompts Token 1, Token 2, Token 3, and Token 4 are provided to support using an XQuery source to authenticate users for a MicroStrategy project. For steps to report on and authenticate using XQuery sources, [Using an XQuery source to authenticate users for a MicroStrategy project, page 649](#).

Defining how a prompt is displayed in MicroStrategy Web

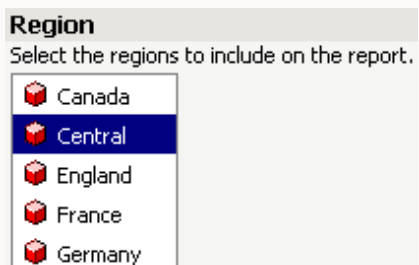
You can determine how a prompt appears when it is executed in MicroStrategy Web, by selecting a display style, and then updating the style settings associated with it. For examples of the display styles, see [Display styles for prompts in MicroStrategy Web, page 259](#). For descriptions of the

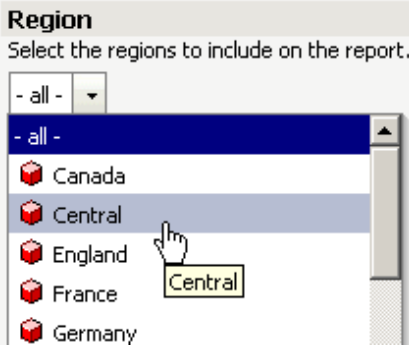
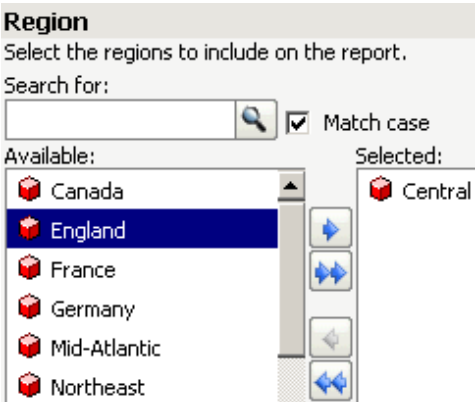
style settings, see [Defining how prompts are displayed in MicroStrategy Web: Style settings, page 262](#).

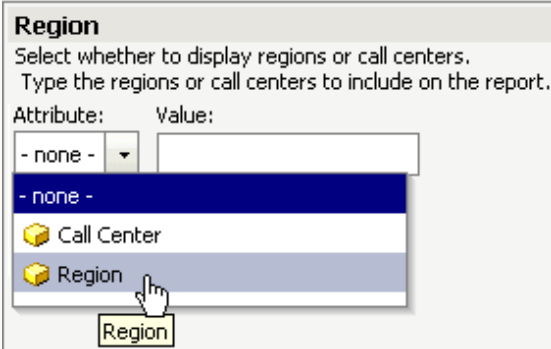
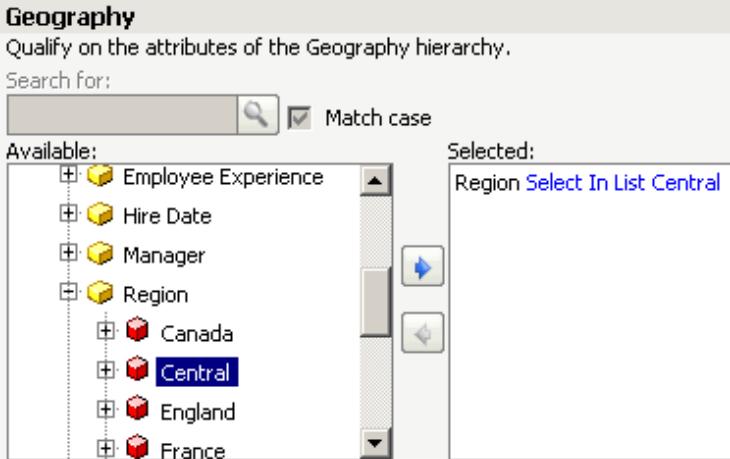
For “Choose from All Attributes in a Hierarchy”, “Qualify on an Attribute”, and “Qualify on a Metric” prompts, you can also configure how the user can define the attribute or metric qualification in the prompt when the prompt is executed in MicroStrategy Web. For details, see [Configuring how the qualification can be defined in MicroStrategy Web prompts: Expression settings, page 265](#).

Display styles for prompts in MicroStrategy Web

The type of prompt determines which display styles are available, as described in the following table:

Display Style	Available for:
Check box 	• Attribute element list prompts • Object prompts
List 	• Attribute element list prompts • Attribute qualification prompts • Metric qualification prompts • Object prompts

Display Style	Available for:
Pull down 	• Attribute element list prompts • Attribute qualification prompts • Metric qualification prompts • Object prompts
Radio button 	• Attribute element list prompts • Attribute qualification prompts • Metric qualification prompts • Object prompts
Shopping cart 	• Attribute element list prompts • Attribute qualification prompts • Hierarchy prompts • Metric qualification prompts • Object prompts Note: In MicroStrategy Web, a user clicks an object to drill to the next level. The user hovers over an object to highlight it, then double-clicks in the highlighted area to add the object to the Selected list.

Display Style	Available for:
Text box 	• Attribute qualification prompts • Metric qualification prompts • Value prompts
Tree 	• Hierarchy prompts Note: In MicroStrategy Web, a user expands the levels to navigate to an object. The user double-clicks an object to add it to the Selected list.

Custom prompt styles for MicroStrategy Web

MicroStrategy provides a wide array of prompt styles out of the box, and you can add custom prompt styles to suit your needs. You can use any custom prompt styles that your project contains. To use a custom prompt style, select the **Display Style** associated with the custom prompt style. Next, select the custom prompt style from the **Custom Style** drop-down list. For more information on custom prompt styles, including steps to create them, see the *Creating a custom prompt style for MicroStrategy Web* topic in the *MicroStrategy Developer help*.

Defining how prompts are displayed in MicroStrategy Web: Style settings

All the style settings for determining how prompts are displayed in MicroStrategy Web are described below. The type of prompt and the display style determine which style settings are available.

- **Allow modification of the logical operator:** Specifies whether the user, when the prompt is executed, can change the logical operator (AND, OR, and so on) of a qualification.

If **Allow modification** is selected, select one of the following options:

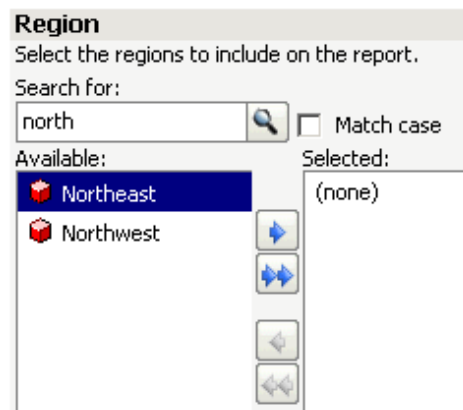
- **Use a single logical operator between all conditions:** The user can select only one logical operator to join all the conditions of the prompt qualification.
 - **Allow the user to set independent logical operators between conditions:** The user can select different logical operators between each condition of the prompt qualification.
- **Allow navigation above the search root:** Specifies whether a user, when the prompt is executed, can navigate the hierarchy path above the search root, as described below:
 - For a prompt that uses a search object in the prompt definition, the search root is the directory specified in the **Look in** field for the search.
 - For a prompt that does not use a search object but allows a search (the **Show search box** option is enabled), the search root is the project directory, for example, `\MicroStrategy Tutorial`.
 - **Allow the user to browse elements in a qualification:** Specifies whether a user, when the prompt is executed, can expand an attribute to select elements to answer the prompt.
 - **Height:** If **List height mode** is set to **Fixed**, specifies the height of the list. If **List height mode** is set to **Automatically**, this setting is unavailable.
 - **Items per column/row:** Defines the number of items to display in a column/row of the check box or radio button list.
 - **List height mode:** Defines whether the height of the list is either:
 - **Automatically** adjusted to the height of the list's contents.
 - **Fixed** at the height specified in the **Height** field.

- **List width mode:** Defines whether the width of the list is either:
 - **Automatically** adjusted to the width of the list's contents.
 - **Fixed** at the width specified in the **Width** field.
- **Orientation:** Specifies whether the check boxes or radio buttons are **Horizontal** (displayed on a single line from left to right) or **Vertical** (displayed in a single column).
- **Show search box:** Determines whether a search of possible prompt answers is allowed when the prompt is executed. Allowing searches is useful if the prompt contains many possible answers. It provides users who know what they want a quick way to find it, while those who do not can scroll through the list until they find what they want.

The Show search box option is available for:

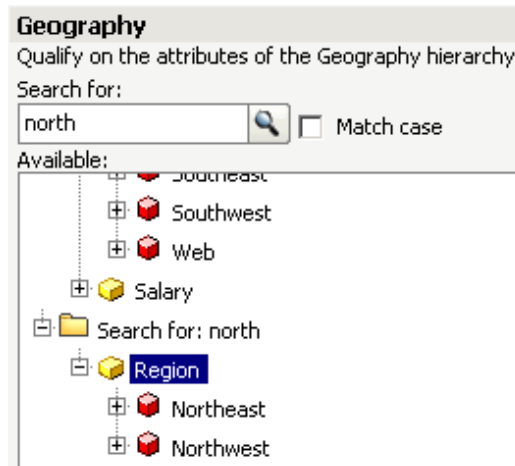
- Attribute element list prompts that list all elements of the attribute
- Attribute qualification prompts and metric qualification prompts that use search objects (for all display styles except Text Box)
- Hierarchy prompts
- Object prompts that use search objects (for all display styles)

For example, an attribute element list prompt lists regions. When the prompt is executed, a user can select from the list of regions, or type “north” in the search box to display only Northeast and Northwest in the list, as shown below:



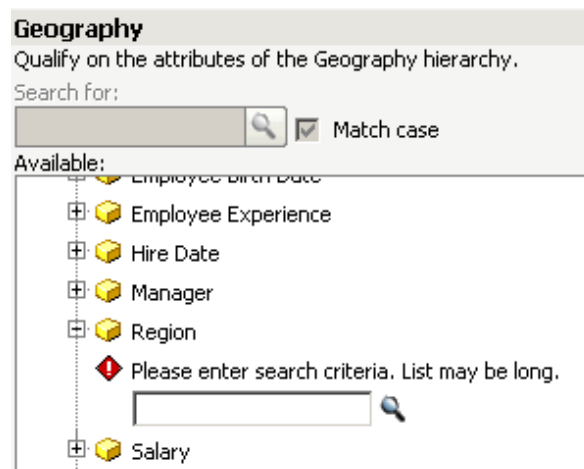
For a hierarchy prompt, the search box is not enabled until the user selects an attribute. In the example shown below, the user selected

Region, the search box was enabled, and the user typed “north”. The search results are displayed at the bottom of the hierarchy list.



 If the prompt is based on a search that contains a string, the search box is disabled when the prompt is executed.

- **Require search:** (Available only for hierarchy prompts) Determines whether the user must specify a search when he is browsing elements in a hierarchy prompt. If a search is not required, the elements are listed in the prompt. A search is useful if long lists of elements are possible. For example, a prompt displays the attributes from the Geography hierarchy. When an attribute is clicked, the attribute elements are listed (for an example, see the Tree example on [page 261](#)). If a search is required, a search box is displayed instead of the attribute elements, as shown below:



 You must enable **Show search box** for the **Require search** check box to become available.

- **Width mode:** Defines whether the width of the text box or pull-down list is either:
 - **Automatically** adjusted to the width of the list or text box contents
 - **Fixed** at the width specified in the **Width** field
- **Width:** If **List width mode** or **Width mode** is set to **Fixed**, specifies the width of the list or text box. If **List width mode** or **Width mode** is set to **Auto**, this setting is unavailable.

Configuring how the qualification can be defined in MicroStrategy Web prompts: Expression settings

For hierarchy, attribute qualification, and metric qualification prompts, you can configure in MicroStrategy Developer how the user can define the attribute or metric qualification in the prompt when the prompt is executed in MicroStrategy Web.

Expression settings for attribute and hierarchy qualifications

The expression settings for attribute and hierarchy qualifications are described below:

- **Expression type allowed:** Select the type of expression that the user can create when the prompt is executed. The options are:
 - **Select:** When the prompt is executed, the user selects an attribute, and then creates a list of attribute elements to include in or omit from the report.

For example, the following prompt, which is being executed in MicroStrategy Web, allows the user to select the regions to include in the report:

- Qualify:** When the prompt is executed, the user creates an attribute qualification to answer the prompt. The user selects an attribute, an attribute form (such as ID or Description), an operator (such as Contains or Begins with), and a value to compare the attribute element to.

For example, a MicroStrategy Web user can select Region, Description, Begins with, and north to include all northern regions in the report, as shown below:

- **Select and Qualify:** When the prompt is executed, the user chooses to either select attribute elements or create an attribute qualification, as described above.

For example, the following prompt, which is being executed in MicroStrategy Web, contains the radio buttons to select the expression type:

- **Default qualification type** (Available only if **Select and Qualify** is chosen above): Select whether the default qualification type should be **Select** or **Qualify**. The user can change from this default when the prompt is executed.
- **Default operator:** Select the default operator (such as In List or Contains) to use in the prompt's attribute qualification. When the prompt is executed, the user can accept this default operator or select a different operator, using the **Is** drop-down list as shown in the samples above.
- **Allow import elements:** By default, this check box is cleared, which indicates that files cannot be imported. Select the check box to allow a user to load a file containing a list of elements for an In List or Not in List type of qualification. The file is loaded when the prompt is executed.
- **Default operator between conditions:** Choose either **AND** or **OR** as the default operator that is used between conditions in the prompt's definition. When the prompt is executed, the user can accept this default operator or select the other operator.

For example, the following prompt, shown in MicroStrategy Web, allows multiple conditions. The user has created a condition on north and another on south. The AND operator was chosen as the default operator between conditions, so the **All selections** option is selected. This report would not return any results (a word cannot begin with both “north” and “south”), so the user can change the **Match** option to **Any selection**

(representing the OR operator). The report will then return all northern and southern regions.

 To enable multiple conditions, define the **Minimum number of qualifications** as a number larger than one.

Expression settings for metric qualifications

The expression settings for metric qualifications are described below. Examples of each follow the descriptions.

- **Default operator:** Select the default operator (such as Equals or Greater than) to use in the prompt's metric qualification. When the prompt is executed, the user can accept this default operator or select a different operator.
- **Display output level selection** check box: By default, this is selected, which indicates that the user can select an output level (metric level, report level, or a specific attribute) when the prompt is executed. Clear the check box to disable output level selection. An output level is the level at which the metric is calculated. For background information on metric levels, see [Level metrics: Applying the metric condition to the metric calculation, page 53](#).
- **Default operator between conditions:** Choose either **AND** or **OR** as the default operator that is used between conditions in the prompt's definition. When the prompt is executed, the user can accept this default operator or select the other operator.

In the following prompt, which is being executed in MicroStrategy Web, the operator is displayed in the **Is** field. The user is setting the output level to

Metric, so that the metric qualification is calculated at the level defined in the metric.

The screenshot shows a configuration window titled "Revenue" with the instruction "Create a metric qualification on revenue." The window is divided into two main sections. The left section contains a list of metrics, with "- none -" at the top and "Revenue" below it, which is highlighted with a blue background. The right section contains configuration options: "Is:" with a dropdown menu set to "Greater than or equal to", "Value:" with a text input field containing "1000000", and "At level:" with a dropdown menu set to "Metric". A list of levels is displayed below the "At level:" dropdown, with "Default", "Metric" (highlighted in blue), and "Report" visible. At the bottom of the window, there is a link that says "Choose attributes...".

DESIGNING REPORTS

Understanding data evaluation order and distributing reports to users

Introduction

A report is a MicroStrategy object that represents a request for a specific set of formatted data from the data warehouse. Reports are the focus and goal of business intelligence. They allow users to gather business insight through data analysis.

The different parts of a report include:

- Attributes and facts from the warehouse
- Filters that determine how much data is used to generate the report
- Metrics that perform calculations on the facts

This chapter builds on the information in the *Basic Reporting Guide*. This chapter focuses on the complex interaction of various report features, and their potential impact on each other and on the final results, during report execution.

Before you begin

The *Building Query Objects and Queries, for Designers* chapter of the *Basic Reporting Guide* contains fundamental information on report design. This advanced chapter builds on the concepts and procedures presented there by providing more technical details and advanced options for report design.

The reports discussed in this chapter are saved in the MicroStrategy Tutorial sample project. These reports are located in `Public Objects\Reports\MicroStrategy Platform Capabilities\Advanced Reporting Guide`.



If you create reports while following steps in this chapter, remember to save any reports you create under a different name than the sample report's name in the Tutorial project, so that you do not overwrite the sample reports in Tutorial.

Report design versus report creation

Report design is the process of building reports from basic report components in MicroStrategy Developer and Web. While report design is the most generic method for defining a report, it also requires the most in-depth knowledge of the project. In general, this method should be available only to the select group of advanced users and report designers who will design reports for others to use.



You can also design, edit, or delete multiple reports at the same time by using a Command Manager script. Command Manager is a MicroStrategy tool designed to automate certain tasks and processes. For example, you can design a number of empty reports based on the same template. For more information about Command Manager, including steps to use scripts, see the *Command Manager* chapter of the *System Administration Guide*.

Report creation is the process of building reports from existing, predesigned reports either in MicroStrategy Developer or Web. Given the wealth of reporting functionality that you can make available to your users, you have the ability to design reports that provide a wide range of options for users to create their own reports, based on the reports you design.

Report creation is different from report design in that it provides a more guided experience and does not require your users to have a thorough

understanding of the project. This allows your users to create their own reports in a controlled, user-friendly environment.

Accessing the Basic Report: Foundation of examples

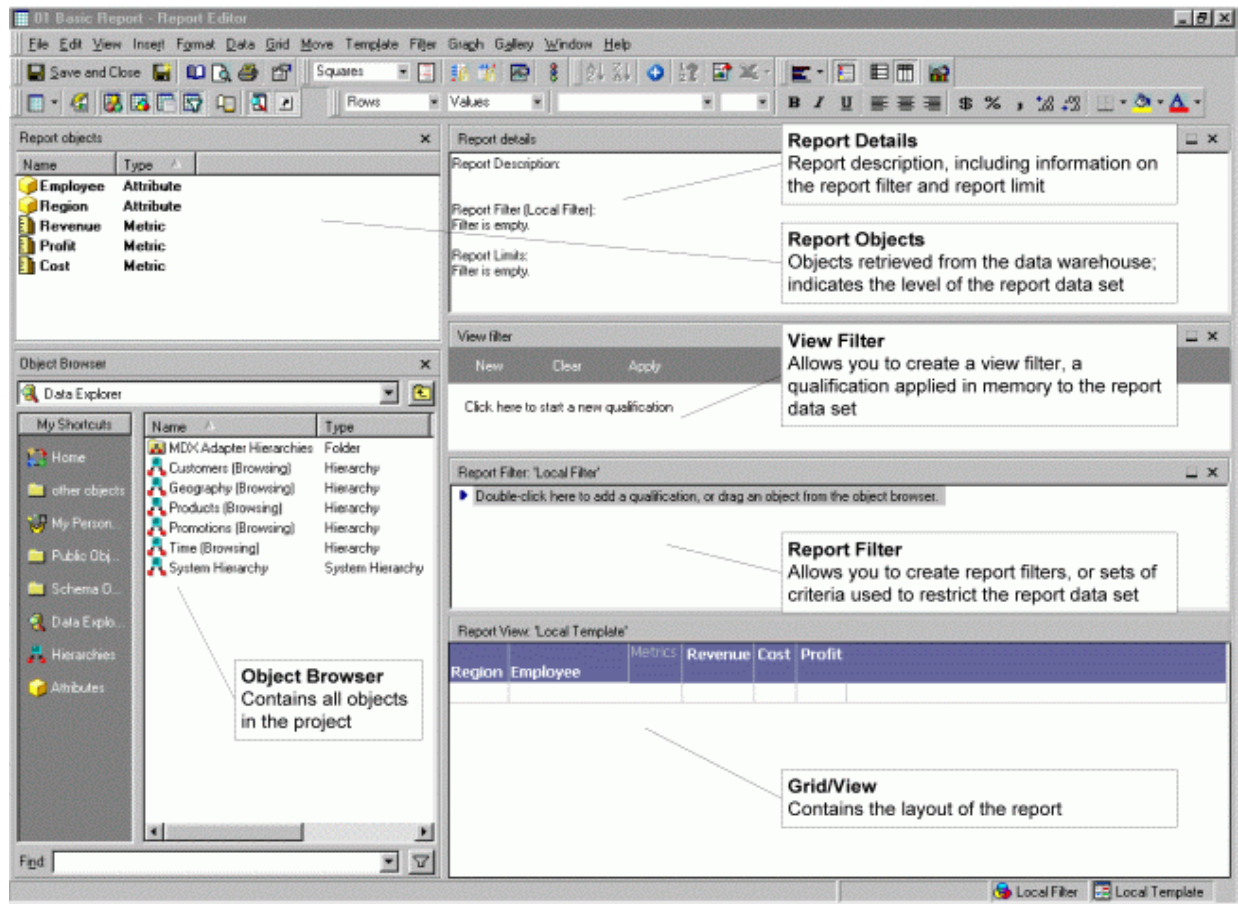
The Basic Report is used as a foundation report to create the more advanced reports used as examples in this chapter.

In MicroStrategy Developer, open the **Basic Report** from the MicroStrategy Tutorial. The report is displayed in Grid View, as shown below.

Region	Employee		Metrics	Revenue	Cost	Profit
Northeast	De Le Torre	Sandra		\$ 514,524	\$ 391,121	\$123,403
	Kelly	Laura		\$ 329,888	\$ 250,120	\$79,768
	Kieferson	Jack		\$ 389,888	\$ 294,751	\$95,137
	Sawyer	Leanne		\$ 316,786	\$ 240,110	\$76,676
	Sonder	Melanie		\$ 421,036	\$ 318,975	\$102,061
	Yager	Beth		\$ 362,742	\$ 275,208	\$87,534
Mid-Atlantic	Bernstein	Lawrence		\$ 403,122	\$ 305,730	\$97,392
	Brown	Vernon		\$ 548,862	\$ 416,009	\$132,853
	Corcoran	Peter		\$ 346,565	\$ 263,369	\$83,196
	Folks	Adrienne		\$ 418,967	\$ 317,132	\$101,835
	Hollywood	Robert		\$ 591,776	\$ 448,257	\$143,519
	Ingles	Walter		\$ 377,583	\$ 286,494	\$91,089
	Smith	Thomas		\$ 420,931	\$ 318,093	\$102,838
	Young	Sarah		\$ 305,534	\$ 231,887	\$73,647

This report calculates the metrics Revenue, Cost, and Profit by the attributes of Region and Employee. Region and Employee define the level of the report. The level of a report is the attribute level at which the metrics are calculated.

Switch to Design View. The following image displays the Basic Report in the Design View of MicroStrategy Developer. All of the panes mentioned in the next paragraphs have been opened and annotated for your use.



Select **Report Objects** from the **View** menu and notice that all the Report Objects are included in the grid. Recall that the Report Objects pane lists all of the objects for which data is retrieved from the database, as well as the derived metrics created for this report.

Filters and order of calculation on reports

About filters

For an introduction to filters, see the *Basic Reporting Guide*. This section builds on information in the *Basic Reporting Guide* by describing how filters

are handled along with other report features during report execution, to impact the resulting report display.

A *filter* is used to select the data for calculating the metrics in the report. It also restricts the attribute elements included in the report. In our example, we use the Month filter, which does not allow April, May, and December data to be included in the metric calculations. For our purposes, these months are not representative of the normal business cycle, so the filter excludes them from calculations.



The Month filter is included in the Reports\
MicroStrategy Platform Capabilities\
Ad hoc Reporting\Component Objects folder.

Report filter example

Add the Month filter to the Basic Report, in Design View. For steps, see the online help. When you re-execute the report, it looks like the following:

Region	Employee	Metrics	Revenue	Cost	Profit
Northeast	De Le Torre Sandra		\$ 354,350	\$ 269,591	\$84,759
	Kelly Laura		\$ 215,628	\$ 163,478	\$52,150
	Kieferson Jack		\$ 229,233	\$ 173,672	\$55,561
	Sawyer Leanne		\$ 198,976	\$ 150,803	\$48,173
	Sonder Melanie		\$ 286,039	\$ 216,651	\$69,388
	Yager Beth		\$ 220,460	\$ 167,350	\$53,110



If you do not want to create it yourself, this report is saved as Filter - Month Report Filter in the Tutorial.

Notice that the metrics have different values than in the Basic Report. For example, Leanne Sawyer's contribution to revenue is \$198,076. In the unfiltered report, her revenue was \$316,786. In the Basic Report, all data for all months was retrieved from the data warehouse. The Revenue metric was calculated using all months. In this filtered report, April, May, and December amounts are not considered, so this metric does not include them in its calculations.



There are several types of filters. [Chapter 3, Advanced Filters](#), discusses filters in greater detail.

A filter affects the nature of the metric calculation by restricting the information used to compute the report metrics.

The difference between report filters and report limits

A *report limit* specifies a set of criteria used to restrict the data returned in the report after the report metrics are calculated. Because it is based on the report's metric values, a limit is applied after all the metrics are calculated. The difference between a filter and a report limit is shown in the following example.

Rank metrics apply a ranking number to the metric values for a given attribute. For an example, open the Sales Rank report. As shown in the following figure, this report is the Basic Report with two additional metrics, Revenue Rank and Revenue Rank (unfiltered).

Region	Employee	Metrics	Revenue	Cost	Profit	Revenue Rank	Revenue Rank (unfiltered)
Northeast	De Le Torre Sandra		\$ 514,524	\$ 391,121	\$123,403	12	12
	Kelly Laura		\$ 329,888	\$ 250,120	\$79,768	32	32
	Kieferson Jack		\$ 389,888	\$ 294,751	\$95,137	25	25
	Sawyer Learne		\$ 316,786	\$ 240,110	\$76,676	33	33
	Sonder Melanie		\$ 421,036	\$ 318,975	\$102,061	19	19
	Yager Beth		\$ 362,742	\$ 275,208	\$87,534	28	28

These metrics rank employees based on the Revenue metric. The Revenue Rank metric uses the report filter in its calculation, while the Revenue Rank (unfiltered) metric ignores this report filter. This feature allows both filtered and unfiltered values on the same report. For example, when a filter for the Northeast region is added to the report, the calculation for Revenue Rank (the filtered metric) uses only the employees in that region. The unfiltered metric uses all the employees, regardless of region, to calculate its rank numbers. A complete example is provided in [Filtering with rank, page 277](#) below. Metric level filtering is also explained in more depth in [Filtering: Interaction with report filters, page 30](#). In the report sample above, these two metrics display the same value because the report does not contain a filter.

Sorting on rank

To make the order of ranking easier to view, sort by the rank metric. In Grid View, right-click the **Revenue Rank** column and select **Sort rows by this column**. As you can see from the following report sample, the rows are

re-arranged based on the value in the Revenue Rank column. The report data does not change, only its order on the report changes.

Region	Employee	Metrics	Revenue	Cost	Profit	Revenue Rank	Revenue Rank (unfiltered)
Web	Walker Robert		\$ 1,716,267	\$ 1,301,142	\$415,125	1	1
Southwest	Bates Michael		\$ 783,886	\$ 594,787	\$189,099	2	2
Northwest	Becker Kyle		\$ 692,441	\$ 525,800	\$166,641	3	3
Southwest	Bell Caitlin		\$ 624,961	\$ 473,529	\$151,432	4	4
Mid-Atlantic	Hollywood Robert		\$ 591,776	\$ 448,257	\$143,519	5	5
Central	Gale Loren		\$ 575,268	\$ 436,105	\$139,163	6	6
Southeast	McClain Sean		\$ 556,149	\$ 420,844	\$135,305	7	7
Mid-Atlantic	Brown Vernon		\$ 548,862	\$ 416,009	\$132,853	8	8
Southeast	Strome Fred		\$ 541,361	\$ 409,968	\$131,393	9	9
	Benner Ian		\$ 526,867	\$ 399,590	\$127,277	10	10

 This report is saved as Sort by Revenue Rank.

Filtering with rank

Switch to the Design View to add the Month filter to the sorted report. When you re-execute it, note the changed values in the Revenue Rank metric. In the following sample, the rankings that have changed are highlighted.

Region	Employee	Metrics	Revenue	Cost	Profit	Revenue Rank	Revenue Rank (unfiltered)
Web	Walker Robert		\$ 1,041,503	\$ 790,379	\$251,124	1	1
Southwest	Bates Michael		\$ 509,269	\$ 386,237	\$123,032	2	2
Northwest	Becker Kyle		\$ 443,333	\$ 336,565	\$106,768	3	3
Southwest	Bell Caitlin		\$ 415,746	\$ 314,706	\$101,040	4	4
Southeast	Benner Ian		\$ 393,866	\$ 298,309	\$95,557	5	10
	Strome Fred		\$ 393,815	\$ 298,311	\$95,504	6	9
Northeast	De Le Torre Sandra		\$ 354,350	\$ 269,591	\$84,759	7	12
Mid-Atlantic	Hollywood Robert		\$ 354,241	\$ 268,206	\$86,035	8	5
South	Pierce Charles		\$ 353,974	\$ 268,166	\$85,808	9	13
Mid-Atlantic	Brown Vernon		\$ 339,208	\$ 257,341	\$81,867	10	8
Northeast	Kelly Laura		\$ 215,628	\$ 163,478	\$52,150	32	32
	Sawyer Leanne		\$ 198,976	\$ 150,803	\$48,173	33	33
Mid-Atlantic	Young Sarah		\$ 104,813	\$ 79,671	\$25,142	34	34

 This report is saved as **Sort by Revenue Rank - Month Report Filter**.

In the previous report, Ian Benner's revenue was \$526,867, which placed him as the tenth highest revenue producer. In this new report, his revenue is calculated at \$393,866 because the report filter is applied before the metric value is determined. The revenue does not include April, May, and December. His new revenue rank is calculated as five, since the report filter affects the data used to calculate the Revenue metric. However, the Revenue Rank (unfiltered) metric still returns a ten because it is set to ignore the report filter.

Report limits with rank

Open the Sort by Revenue Rank report. Notice that the highest rank is 34 and there are 34 rows in the report. Now, add a report limit of revenue greater than \$320,000. (For steps to add a report limit, see the online help.) Re-execute the report to see the following results.

Region	Employee	Metrics				Revenue Rank	Revenue Rank (unfiltered)
			Revenue	Cost	Profit		
Web	Walker Robert		\$ 1,716,267	\$ 1,301,142	\$415,125	1	1
Southwest	Bates Michael		\$ 783,886	\$ 594,787	\$189,099	2	2
Northwest	Becker Kyle		\$ 692,441	\$ 525,800	\$166,641	3	3
Southwest	Bell Caitlin		\$ 624,961	\$ 473,529	\$151,432	4	4
Mid-Atlantic	Hollywood Robert		\$ 591,776	\$ 448,257	\$143,519	5	5
Central	Gale Loren		\$ 575,268	\$ 436,105	\$139,163	6	6
Southeast	McClain Sean		\$ 556,149	\$ 420,844	\$135,305	7	7
Mid-Atlantic	Brown Vernon		\$ 548,862	\$ 416,009	\$132,853	8	8
Southeast	Strome Fred		\$ 541,361	\$ 409,968	\$131,393	9	9
	Benner Ian		\$ 526,867	\$ 399,590	\$127,277	10	10
Central	Torrison Mary		\$ 333,377	\$ 253,507	\$79,870	31	31
Northeast	Kelly Laura		\$ 329,888	\$ 250,120	\$79,768	32	32



This report is saved as Sort by Revenue Rank - Report Limit - Revenue > 320K.

Notice that the highest rank is now 32 and there are only 32 rows on the report. The last two rows from the previous report have disappeared because the revenue in each row was less than the report limit. None of the metrics changed values because a report limit does not affect how the metrics are calculated; the limit is applied at the level of the report after the metrics are calculated.

Simultaneous report filters and limits

Both report filters and report limits can be used on the same report because they are applied at different stages of the execution cycle.

Right-click the Sort by Revenue Rank report on MicroStrategy Developer to open it in the Design View for editing. Add the Month filter as the report filter and a report limit of revenue greater than \$320,000, as described

previously. Execute the report. The results appear as displayed in the following figure.

Region	Employee	Metrics					
			Revenue	Cost	Profit	Revenue Rank	Revenue Rank (unfiltered)
Web	Walker Robert		\$ 1,041,503	\$ 790,379	\$251,124	1	1
Southwest	Bates Michael		\$ 509,269	\$ 386,237	\$123,032	2	2
Northwest	Becker Kyle		\$ 443,333	\$ 336,565	\$106,768	3	3
Southwest	Bell Caitlin		\$ 415,746	\$ 314,706	\$101,040	4	4
Southeast	Benner Ian		\$ 393,866	\$ 298,309	\$95,557	5	10
	Strome Fred		\$ 393,815	\$ 298,311	\$95,504	6	9
Northeast	De Le Torre Sandra		\$ 354,350	\$ 269,591	\$84,759	7	12
Mid-Atlantic	Hollywood Robert		\$ 354,241	\$ 268,206	\$86,035	8	5
South	Pierce Charles		\$ 353,974	\$ 268,166	\$85,808	9	13
Mid-Atlantic	Brown Vernon		\$ 339,208	\$ 257,341	\$81,867	10	8
Southwest	Schafer Rose		\$ 337,269	\$ 255,190	\$82,079	11	11
	Johnson Andrew		\$ 330,672	\$ 250,951	\$79,721	12	15
Southeast	McClain Sean		\$ 328,536	\$ 249,001	\$79,535	13	7
Central	Gale Loren		\$ 327,194	\$ 248,076	\$79,118	14	6
South	Nelson Arthur		\$ 324,212	\$ 245,341	\$78,871	15	14



This report is saved as Sort by Revenue Rank - Report Filter & Report Limit.

Notice that the report is much smaller than either the Sort by Revenue Rank - Month Report Filter report or the Sort by Revenue Rank - Report Limit - Revenue > 320K report. Only 15 rows are returned, as opposed to 34 or 32. Also notice that the Revenue, Cost, Profit, and Revenue Rank values are the same as the filtered report. However, the Revenue Rank (unfiltered) values are the same as the Revenue Rank - Limit.

The first step in creating this report is calculating metrics. The data used in the metrics is restricted by the report filter, so information from April, May, and December is not included. All the metrics are calculated using this data, except for the unfiltered metric, which ignores the report filter. Its values are calculated on the full year's worth of data.

The results after all the metric calculations are completed form the report result set. The report limit is applied to this result set. The employees with revenue less than \$320,000 (the report limit) are removed from the display before the report is presented. Because the revenue is calculated on fewer months than the Revenue Rank - Month Filter report, more employees are discarded than from the previous limit.

This means that the limit stays the same (greater than \$320,000), but the filter changes the data considered in calculating each employee's rank.

A report filter affects the data used to calculate metrics, whereas a report limit does not affect how the metrics are calculated. Report limits are applied at the level of the report after the metrics are calculated.

About metric qualifications

A *metric qualification* is a filtering condition based on the value of a metric. It contains an output level, which determines the level at which the metric is calculated and to which attributes the metric applies. Like every filter, a metric qualification changes the nature of the metric calculations, unlike a report limit, which is applied after the metrics are calculated.

Recall that the level of the Basic Report is Region and Employee—the attributes on the report. The output level of the metric qualification can remain at the report level, or it can be changed.

If the output level is the same as the report level, the results are usually the same as using a report limit. This is just a coincidence, however, because report limits and metric qualifications are calculated differently and at different times in the report execution cycle.

If the output level differs from the report level, the metrics are calculated at the output level. In the example that follows, the report level is region and employee. In the previous reports, the metrics were calculated for each employee using all brands and all products. When a metric qualification with an output level is applied to the report, the metrics are calculated with only the data that meets the metric qualification. Working through the sample report will help you better understand metric qualifications and output levels.

Whether or not the output level differs from the report level, a metric qualification affects the report result set. On the other hand, a report limit is applied after the metrics are calculated.

Example: Metric qualification

Right-click the Sort by Revenue Rank report on MicroStrategy Developer and select **Edit** to edit the report. Add a metric qualification by following the steps that follow.

To add a metric qualification

- 1 Double-click in the Report Filter pane to add a qualification.
- 2 Select **Add a Set qualification** and click **OK**. A set qualification is based on a metric or attribute relationships.

- 3 Click the browse button next to Output Level.
- 4 Select **Calculate the output for the list of attributes**. This allows you to select the output level for the metric qualification.
- 5 Select **Brand** under the Products folder and click > to add it to the **Selected objects** list.
- 6 Click **OK**.
- 7 Click the browse button next to Metric.
- 8 Select **Revenue** in the Sales Metrics folder.
- 9 Click **OK**.
- 10 Keep the Function as Metric Value, but select **Greater than** from the **Operator** drop-down list.
- 11 Do not change Value, but type **320000** in the box next to it.
- 12 Click **OK**.

Execute the report. The results are displayed in the following figure.

Region	Employee		Metrics			Revenue Rank	Revenue Rank (unfiltered)
			Revenue	Cost	Profit		
Web	Walker	Robert	\$ 1,104,263	\$ 802,431	\$301,832	1	1
Southwest	Bates	Michael	\$ 497,685	\$ 361,893	\$135,792	2	2
Northwest	Becker	Kyle	\$ 435,122	\$ 316,082	\$119,040	3	3
Southwest	Bell	Caitlin	\$ 406,314	\$ 295,437	\$110,877	4	4
Mid-Atlantic	Hollywood	Robert	\$ 388,117	\$ 282,335	\$105,782	5	5
Central	Gale	Loren	\$ 369,801	\$ 268,847	\$100,954	6	6
Southeast	McClain	Sean	\$ 362,130	\$ 262,818	\$99,312	7	7
	Strome	Fred	\$ 353,170	\$ 256,531	\$96,639	8	9
Mid-Atlantic	Brown	Vernon	\$ 350,440	\$ 254,555	\$95,885	9	8
Southeast	Berner	Ian	\$ 333,718	\$ 242,444	\$91,274	10	10



This report is saved as Sort by Revenue Rank - Report Filter - Metric Qualification at the Brand Level.

The metric values on the report are different from those calculated for the Sort by Revenue Rank report. The Sort by Revenue Rank report produces values for each employee for all products. On the other hand, the metrics on this report are calculated only on those brands with revenue greater than \$320,000 because of the metric qualification.

In the Sort by Revenue Rank report, Fred Strome's revenue rank was nine, with revenue of \$541,361. On this metric-qualified report, his revenue is \$353,170, because any brands with revenue less than \$320,000 were not included in the Revenue metric calculation. While his unfiltered revenue rank remains the same, he has moved up to eight in the revenue ranking. The unfiltered metric does not include the metric qualification, so it is calculated on all brands, and therefore, all products. In contrast, the metric qualification affects the other Rank metric, just as it affects the Revenue, Cost, and Profit metric calculations. Therefore, only brands with more than \$320,000 of revenue are included in those calculations.

You can think of a metric qualification as creating a temporary report. When the report is executed, the metric qualification first generates a temporary report in the background. In the earlier example, that report is a list of brands. The qualification is applied, so the report is trimmed to include only those brands with revenue in excess of \$320,000. This report looks like the following.

Brand	Metrics	Revenue
Sony		\$ 2,509,622
Sharp		\$ 2,253,700
Panasonic		\$ 1,722,800
Harman Kardon		\$ 908,200
Hitachi		\$ 687,800
Minolta		\$ 622,080
Hewlett Packard		\$ 532,562
RCA		\$ 528,530
Olympus		\$ 454,400
Pioneer		\$ 349,150
3Com		\$ 340,940



This report is saved as Revenue by Brand.

Then this temporary report is applied to the actual report. Metrics are calculated including only those brands listed on the temporary report—Sony, Sharp, Panasonic, and so on. In essence, this report is the same as creating a filter for the set of brands Sony, Sharp, Panasonic, and so on. However, unlike that filter, the metric qualification is dynamically calculated based on the Revenue metric at the brand level. When new revenue data is added, the values can change.



In many cases, a report limit can generate more efficient SQL than a metric qualification. A metric qualification is contained in a separate pass of SQL, generating a temporary table at the output level. When

this table is joined to the rest of the output, it limits the data included in the other metric calculations. Because it is another table, a metric qualification is a separate step in report execution. In contrast, a report limit is contained in a HAVING or WHERE clause in one of the final SQL passes. Therefore, using a report limit reduces the number of SQL passes needed to execute the report. However, since they often yield different results, do not choose a report qualification or a limit based solely on SQL efficiency.

About the report-as-filter

Report as filter allows you to create a report and use it as a filter to generate another report. It is a different way to achieve the same results as a metric qualification. Because the logic used to generate the final report is clearer, MicroStrategy recommends using it rather than the metric qualification.



In MicroStrategy Developer, select **Add a Shortcut to a Report** to access the report as filter functionality.

Report as filter example

To create the same report as the metric qualification example, open the Sort by Revenue Rank report in the Report Editor. Add a new report filter. Select **Add a Shortcut to a Report** and choose the **Revenue by Brand** report. Execute the report. Sample report results are shown in the following figure.

Region	Employee		Metrics	Revenue	Cost	Profit	Revenue Rank	Revenue Rank (unfiltered)
Web	Walker	Robert		\$ 1,104,263	\$ 802,431	\$301,832	1	1
Southwest	Bates	Michael		\$ 497,685	\$ 361,893	\$135,792	2	2
Northwest	Becker	Kyle		\$ 435,122	\$ 316,082	\$119,040	3	3
Southwest	Bell	Caitlin		\$ 406,314	\$ 295,437	\$110,877	4	4
Mid-Atlantic	Hollywood	Robert		\$ 388,117	\$ 282,335	\$105,782	5	5
Central	Gale	Loren		\$ 369,801	\$ 268,847	\$100,954	6	6
Southeast	McClain	Sean		\$ 362,130	\$ 262,818	\$99,312	7	7
	Strome	Fred		\$ 353,170	\$ 256,531	\$96,639	8	9
Mid-Atlantic	Brown	Vernon		\$ 350,440	\$ 254,555	\$95,885	9	8
Southeast	Benner	Ian		\$ 333,718	\$ 242,444	\$91,274	10	10



This report is saved as Sort by Revenue Rank - Report Filter - Report as Filter at the Brand Level.

As with the metric qualification report, the metric values differ from the unfiltered Sort by Revenue Rank report. The values shown in the earlier

figure are calculated only for the brands that are returned on the Revenue by Brand report chosen as the filter.

Freeform SQL and Query Builder reports can also support this functionality, as described in [Using Freeform SQL reports to filter other reports, page 628](#) and [Using Query Builder reports to filter other reports, page 726](#).

Defining a filter for a drilled-to report

When drilling, you can define a filter for the resulting, drilled-to report. This filter normally includes only the relevant attribute elements to the left of (or above) the object on which you are drilling.

Including only highlighted elements to the left of or above the selected element

For example, a report contains the Category and Subcategory attributes and the Revenue metric, as shown below:

Report Filter (Local Filter):
Filter is empty.

Category	Subcategory	Revenue
Books	Art & Architecture	\$859,358
	Business	\$589,246
	Literature	\$295,385
	Books - Miscellaneous	\$366,250
	Science & Technology	\$1,375,727
	Sports & Health	\$538,269

Click **Books** in the report grid and notice how all the subcategories in the Book rows are highlighted. Then right-click any of the Book subcategories and drill down to **Item**.

The new report is filtered on the Books category, as shown below. Subcategory is not included in the report filter because it is the attribute on which you drilled.

Report Filter (Local Filter):
Category = Books

Category Item		Revenue
Books	100 Places to Go While Still Young at Heart	\$135,930
	Art As Experience	\$31,738
	The Painted Word	\$17,495
	Hirschfeld on Line	\$103,318
	Adirondack Style	\$77,428
	Architecture : Form, Space, & Order	\$82,099
	50 Favorite Rooms	\$51,105
	500 Best Vacation Home Plans	\$23,239
	Blue & White Living	\$32,014

Including all highlighted elements in the filter

If you set the report's Drilling Filter option to include all highlighted attribute elements, and then perform the same drilling operation as in the example above, the drilled-to report contains the same data. However, note the report filter, which now includes Subcategory instead of Category, as shown below:

Report Filter (Local Filter):
Subcategory = Art & Architecture,Business,Literature,Books - Miscellaneous,Science & Technology or Sports & Health

Category Item		Metrics	Revenue
Books	100 Places to Go While Still Young at Heart		\$135,930
	Art As Experience		\$31,738
	The Painted Word		\$17,495
	Hirschfeld on Line		\$103,318
	Adirondack Style		\$77,428
	Architecture : Form, Space, & Order		\$82,099
	50 Favorite Rooms		\$51,105
	500 Best Vacation Home Plans		\$23,239
	Blue & White Living		\$32,014

Since Subcategory was highlighted when you drilled on it, it is included in the report filter. Category is no longer included in the report filter because the only items that appear on the new report are from the Books category.

To include all highlighted attribute elements in the drilling filter for a report

- 1 Open the report in the Report Editor.
- 2 From the **Data** menu, select **Report Data Options**. The Report Data Options dialog box opens.
- 3 Expand **General**, then select **Drilling**.
- 4 Set **Drilling Filter Options** to **Include in the filter all highlighted attribute elements**.
- 5 Click **OK**. The Report Data Options dialog closes.

The basics of report execution

Now that you have designed reports containing filters and limits, you can better understand how a report is generated. The following table describes the steps to execute a report.

Step	Description
1	The objects in Report Objects and the report filter are used to calculate all the metrics, based on the data in the data warehouse.
2	A logical result set is generated in the database or brought back to the Intelligence Server. Optimally, the result set remains in the database to increase performance.
3	If there is a report limit, it is applied at the level of the Report Objects to further restrict the result set. The report limit is based on the result of the metric calculations from step 1.
4	If there are no other functions, the report is returned to the user and displayed in the selected format.

These four steps are the data definition section of the report execution. The data definition establishes how the data is accessed and manipulated in the data warehouse.

The other functions noted in step 4 comprise the view definition, which represents how the data is viewed and manipulated in Intelligence Server. The remainder of this chapter covers manipulating the final report result set generated in step 3.

Data definition and view definition objects: Data versus display

The following tables are samples of the information stored in the data definition and the view definition.

Data Definition	
Report Filter	Criteria used to select the data to calculate the metrics in the report
Report Objects	List of objects that make up the data definition; the attributes define the level of detail of the report Note: Derived metrics are listed in Report Objects but are not part of the data definition.
Report Limits	Additional limits applied after report metrics are calculated

View Definition	
Grid	Objects contained in rows, columns, and pages
Formatting	Font, number format, grid format, and column aliases
Thresholds	Conditional formatting
View Filter	Additional filter applied in memory to the report data
Derived Metrics	Calculations based on metrics already in report and generated from the report result set
Derived Elements	Groupings of attribute elements on a report, to provide a new view of report data for analysis and formatting purposes.
Subtotals	Metric values aggregated to selected attribute levels
Sorting	Sort used to display data in grid

Report designers are generally concerned with data definition while report analysts usually focus on view definition. Report designers work on the details of reports to create a controlled context or environment for the report analysts to work in. This ensures that only reasonable queries are submitted to the database. Reasonable means that irrelevant result sets cannot be created, nor can huge amounts of data be retrieved from the warehouse. This allows a group of report designers to be trained about more advanced report functions while report analysts can manipulate reports without needing to understand the report execution details. Through privileges, you can assign different levels of functionality to different users.

For details on the view definition of reports, and other OLAP analysis features, such as allowing analysts to filter the display of data; on-the-fly

report result calculations; defining attribute elements on-the-fly; and changing the level of data aggregation and display, see the *In-memory Analytics Guide*.

Another method of limiting the data available to analyze on a report is through the use of Intelligent Cubes, which are a part of OLAP Services. For an introduction to Intelligent Cubes, see [Intelligent Cubes](#) below.

Intelligent Cubes

Intelligent Cubes are multi-dimensional cubes (sets of data) that allow you to use OLAP Services features on reports, as well as share sets of data among multiple reports. On MicroStrategy's business intelligence platform, you have two unique methods to implement Intelligent Cube Technology:

- **Personal Intelligent Cubes:** A personal Intelligent Cube is the Intelligent Cube functionality available in all pre-9.0 MicroStrategy releases. To use a personal Intelligent Cube, you can begin by creating reports in MicroStrategy as usual, and then analyze your reports with OLAP Services features such as view filters, derived metrics, and dynamic aggregation. These features are processed on the in-memory copy of data known as a personal Intelligent Cube, rather than being processed on the data warehouse.
- **Intelligent Cubes:** A shared Intelligent Cube (called an Intelligent Cube) is a set of data that can be shared as a single in-memory copy, among many different reports created by multiple users. Rather than returning data from the data warehouse for a single report, you can return sets of data from your data warehouse and save them directly to Intelligence Server memory. The reports accessing Intelligent Cubes can use all of the OLAP Services features for analysis and reporting purposes.

For information on Intelligent Cubes and the OLAP Services features they support, see the *In-memory Analytics Guide*. For information on administering shared Intelligent Cubes, see the *System Administration Guide*.

Totaling and subtotalling data on reports

The term “subtotal” is used to mean all types of totaling, including grand totals, totals, and subtotals.

About subtotals

In the context of reports, subtotals reflect data rolled up to the selected attribute levels and can be applied dynamically to any report. You can apply subtotals using one of many standard subtotal functions such as total, count, minimum, maximum, standard deviation, and others. For a list of the standard subtotal functions, see [Standard subtotal functions, page 103](#). You can create a customized user-defined subtotal using the Subtotal Editor. For more information, see [Creating your own subtotals, page 104](#).

On a report, you can apply the subtotal by position, across a level, or using group by. This Applied levels option, which is located on the Advanced Subtotals dialog box, does not affect the location of the subtotal on the report, but rather how the subtotal is calculated. To change the display location, use the **Display Options** tab in the Subtotals dialog box; for steps, see the *MicroStrategy Developer help*.

- Applying a subtotal *across a level* calculates a subtotal across the selected attributes. The subtotal is applied to particular levels—rows, columns, and pages. This means “group by attributes to the left of the selected attribute.”

For example, if you have Region and Employee, in that order, on a report (as on the Basic Report), selecting across Employee means group by Region. A subtotal for each Region, totaling the individual Employee-Region values, displays on the report. Likewise, across Region means group by none since there is nothing to the left of it on the report. The result is a grand total. However, if the report is pivoted and the order of the attributes changes, the totals also change. If Employee is pivoted to the left of Region, the across Employee subtotal means group by none.

- The *by position* option means applying the subtotal based on its location on the report. The subtotal is calculated across all attributes and hierarchies on the report. It provides the same behavior as across level, but without selecting a level. Instead, the level is selected dynamically so these subtotals change as you alter the layout of the template. The two choices for by position are All subtotals, meaning “across all attributes,” and Grand Total, meaning “across the leftmost attribute.”

For example, you can choose to subtotal on rows and/or columns. The Basic Report contains the columns Region, Employee, Revenue, Cost, and Profit. You can subtotal by both rows and columns, which provides totals at the employee and region level for each metric.



By default, the by position option is selected.

- *Group by* applies the subtotal by the selected attribute across all other attributes on the template, regardless of position. Group by effectively allows you to use both subtotal and sort by attributes that are not the furthest to the left. The Grand Total check box allows you to also add a subtotal grouped by nothing, effectively calculating a total of all attributes on the template.

If a report contains Region, Category, and Quarter and you group by Region, a Region subtotal always appears, regardless of where Category and Quarter are located with respect to Region. You can also group by multiple attributes. For example, grouping by Region-Category on that report provides a subtotal every time a new Region-Category combination occurs.



Group by works best if the report is sorted by the same attribute used to group the subtotals, regardless of position.

Subtotals by position example

Open the Subtotals report, a sample of which is displayed below. This report is based on the Basic Report, with the addition of the attribute Quarter. Also,

a view filter has been added, which includes only quarters 1 and 2 of Year 2002 and the Northeast, Central, and South regions.

Quarter	Region	Employee	Metrics	Revenue	Cost	Profit
Q1 02	Northeast	De Le Torre	Sandra	\$ 60,125	\$ 46,020	\$14,105
		Kelly	Laura	\$ 37,129	\$ 28,023	\$9,106
		Kiefferson	Jack	\$ 38,169	\$ 28,998	\$9,171
		Sawyer	Leanne	\$ 38,980	\$ 29,328	\$9,652
		Sonder	Melanie	\$ 42,125	\$ 32,049	\$10,076
		Yager	Beth	\$ 39,651	\$ 30,058	\$9,593
		Total		\$ 256,179	\$ 194,476	\$61,703
	South	Conner	Beatrice	\$ 46,307	\$ 34,987	\$11,320
		Nelson	Arthur	\$ 47,746	\$ 36,308	\$11,438
		Pierce	Charles	\$ 53,195	\$ 40,429	\$12,766
		Total		\$ 147,248	\$ 111,724	\$35,524
		Total		\$ 586,599	\$ 445,504	\$141,095
Q2 02	Northeast	De Le Torre	Sandra	\$ 66,824	\$ 50,833	\$15,991
		Kelly	Laura	\$ 45,174	\$ 34,287	\$10,887
		Kiefferson	Jack	\$ 76,518	\$ 57,508	\$19,010
		Sawyer	Leanne	\$ 51,592	\$ 39,012	\$12,580
		Sonder	Melanie	\$ 69,371	\$ 52,368	\$17,003
		Yager	Beth	\$ 62,244	\$ 47,195	\$15,049
		Total		\$ 371,723	\$ 281,203	\$90,520
	South	Conner	Beatrice	\$ 52,705	\$ 39,964	\$12,741
		Nelson	Arthur	\$ 79,624	\$ 59,820	\$19,804
		Pierce	Charles	\$ 76,334	\$ 57,755	\$18,579
		Total		\$ 208,663	\$ 157,539	\$51,124
		Total		\$ 847,940	\$ 641,025	\$206,915
		Total		\$ 1,434,539	\$ 1,086,529	\$348,010

 You can create this report yourself by starting with the Basic Report. Note that the Subtotals report has a different format than the Basic Report. The Subtotals report uses the autostyle named SmallType, while Basic Report uses Squares.

Each region is totaled for each quarter, then each quarter is totaled, and finally a grand total is calculated. The subtotals use the by position option. To view how these subtotals are set up, select **Subtotals** from the **Data** menu.

 Press **F11** to show or hide the grand total display for reports in MicroStrategy Developer.

Move Region to the left of Quarter and notice that the subtotals change. Instead of totals by region, by quarter, and then a grand total, the subtotals are calculated by quarter, by region, and then for all attributes (a grand total). This dynamic recalculation is a feature of the subtotal by position option. Return Region to its position between Quarter and Employee.

Subtotals across levels example

Begin with the Subtotals report, and change the by position subtotals to across levels.

To set subtotals across levels

- 1 Select **Data**, then **Subtotals**. The Subtotals dialog box opens.
- 2 Click **Advanced**. The Advanced Subtotals dialog box opens.
- 3 Select **Across level**. A list of report objects is displayed.
- 4 Select **Region** from the list of report objects.
- 5 Click **OK**, then **OK** again to return to the report.

The only totals now are quarterly, as displayed below.

				Metrics	Revenue	Cost	Profit
Q1 02	Northeast	De Le Torre	Sandra		\$ 60,125	\$ 46,020	\$14,105
		Kelly	Laura		\$ 37,129	\$ 28,023	\$9,106
		Kieferson	Jack		\$ 38,169	\$ 28,998	\$9,171
		Sawyer	Leanne		\$ 38,980	\$ 29,328	\$9,652
		Sonder	Melanie		\$ 42,125	\$ 32,049	\$10,076
		Yager	Beth		\$ 39,651	\$ 30,058	\$9,593
	South	Conner	Beatrice		\$ 46,307	\$ 34,987	\$11,320
		Nelson	Arthur		\$ 47,746	\$ 36,308	\$11,438
		Pierce	Charles		\$ 53,195	\$ 40,429	\$12,766
	Total				\$ 586,599	\$ 445,504	\$141,095
Q2 02	Northeast	De Le Torre	Sandra		\$ 66,824	\$ 50,833	\$15,991
		Kelly	Laura		\$ 45,174	\$ 34,287	\$10,887
		Kieferson	Jack		\$ 76,518	\$ 57,508	\$19,010
		Sawyer	Leanne		\$ 51,592	\$ 39,012	\$12,580
		Sonder	Melanie		\$ 69,371	\$ 52,368	\$17,003
		Yager	Beth		\$ 62,244	\$ 47,195	\$15,049
	South	Conner	Beatrice		\$ 52,705	\$ 39,964	\$12,741
		Nelson	Arthur		\$ 79,624	\$ 59,820	\$19,804
		Pierce	Charles		\$ 76,334	\$ 57,755	\$18,579
	Total				\$ 847,940	\$ 641,025	\$206,915

Across levels means group by attributes to the left of the selected attribute. Since the selected attribute is Region, the only attribute to the left of it is Quarter, hence the quarterly totals.

As you did with the by position example, move Region to the left. Only grand total is displayed, because now there is no attribute to the left of Region.

Return Region to its position between Quarter and Employee.

Subtotals group by example

Begin with the Subtotals report, which contains subtotals by position. Sort the report by region, by right-clicking **Region** in the grid and selecting **Sort**, then **Ascending**. Notice how the Q1 and Q2 Totals now appear at the bottom of the report.

Move Region to the right, after Employee. The employees for each region are displayed, then employee totals for each quarter, with a quarterly total, and finally a grand total. Now change the by position subtotals to group by.

To set group by subtotals

- 1 Select **Data**, then **Subtotals**. The Subtotals dialog box opens.
- 2 Click **Advanced**. The Advanced Subtotals dialog box opens.
- 3 Select **Group by**. A blank list of group by levels is displayed.
- 4 Click **Add**. The Group By Selection dialog box opens.
- 5 Select **Region** from the list of attributes on the report.
- 6 Click **OK** to return to the Advanced Subtotals dialog box. Notice that Region has been added to the list of levels.
- 7 Click **OK**, then **OK** again to return to the report.

Now the sort and the subtotals work together to provide regional totals, as shown below.

Quarter	Employee		Region	Metrics	Revenue	Cost	Profit
Q1 02	Ellerkamp	Nancy	Central		\$ 50,167	\$ 38,099	\$12,068
	Gale	Loren	Central		\$ 53,012	\$ 40,333	\$12,679
	Torrison	Mary	Central		\$ 32,620	\$ 25,040	\$7,580
	Zemlicka	George	Central		\$ 47,373	\$ 35,832	\$11,541
Q2 02	Ellerkamp	Nancy	Central		\$ 50,307	\$ 37,932	\$12,375
	Gale	Loren	Central		\$ 106,610	\$ 80,466	\$26,144
	Torrison	Mary	Central		\$ 48,515	\$ 36,750	\$11,765
	Zemlicka	George	Central		\$ 62,122	\$ 47,135	\$14,987
Total			Central		\$ 450,726	\$ 341,587	\$109,139
Q1 02	De Le Torre	Sandra	Northeast		\$ 60,125	\$ 46,020	\$14,105
	Yager	Beth	Northeast		\$ 62,244	\$ 47,195	\$15,049
Total			Northeast		\$ 627,902	\$ 475,679	\$152,223
Q1 02	Conner	Beatrice	South		\$ 46,307	\$ 34,987	\$11,320
	Nelson	Arthur	South		\$ 47,746	\$ 36,308	\$11,438
	Pierce	Charles	South		\$ 53,195	\$ 40,429	\$12,766
Q2 02	Conner	Beatrice	South		\$ 52,705	\$ 39,964	\$12,741
	Nelson	Arthur	South		\$ 79,624	\$ 59,820	\$19,804
	Pierce	Charles	South		\$ 76,334	\$ 57,755	\$18,579
Total			South		\$ 355,911	\$ 269,263	\$86,648

Custom report subtotals

By default, when you use subtotals in a report, the same subtotal function is used for all metrics in the report. The name of the subtotal is displayed in the subtotal line items that appear in the report. You can use custom report subtotals to give you more control over the characteristics of a subtotal. *Custom report subtotals* allow you to define custom subtotal line items that appear on your reports. Custom report subtotals allow you to do the following:

- Customize the subtotal name that appears in the subtotal line item
- Define different subtotal functions to be used on different metrics in the report
- Specify the level of each total
- Turn off subtotalling for specific metrics on the report

You can make the subtotal name dynamic by typing special characters in the subtotal name field as listed in the following table.

Character	Description
#A	The name of the attribute under which the subtotal appears.
#P	The name of the attribute to the left of, or above the attribute under which the subtotal appears.
#0	All the forms of the parent element.
#1	The first form of the parent element reading from left to right or from top to bottom.
#2	The second form of the parent element reading from left to right or from top to bottom.
#3	The third form of the parent element reading from left to right or from top to bottom.
#4	The fourth form of the parent element reading from left to right or from top to bottom.



An attribute form provides details that identify and describe an attribute. Examples are an ID, a description, or a name. For more information, see the *Attributes* chapter in the *Project Design Guide*.

Custom report subtotal example

Open the Subtotals report from the previous example. You can add custom report subtotals for the region and quarter by following the steps outlined below.

To add custom report subtotals

- 1 Select **Subtotals** from the **Data** menu. The Subtotals dialog box opens.
- 2 Clear the **Total** check box to remove the standard subtotals.
- 3 Click **Advanced**, then **New** to create a custom subtotal.

- 4 Type the following for the name: `Total for the #P #0`

P displays the parent attribute and 0 (the number zero, not the letter o) displays all the forms of the parent attribute. In this case, only one form exists for each.
- 5 All the metrics on the report are listed. You can select the subtotal function to use for each. Total is correct for all of our metrics.
- 6 Click **OK** to save the new subtotal.
- 7 Click **OK** to return to the Subtotals dialog box.
- 8 Select the **Total for the #P #0** check box. Notice the icon for this custom report subtotal is different from those of the prebuilt subtotals.
- 9 Click **Advanced**. On the Advanced Subtotals Options dialog box, select **Across level**, and then select the check boxes **Region** and **Employee**.
- 10 Click **OK** to return to the Subtotals dialog box.
- 11 Create another custom report subtotal, called Grand Total. Do not change the subtotal functions for any of the metrics.
- 12 Select the **Grand Total** check box.
- 13 Select **Across level** and **Quarter**.
- 14 Click **OK** to return to the Subtotals dialog box.
- 15 Click **OK**.

The report results are displayed below.

			Metrics	Revenue	Cost	Profit
Quarter	Region	Employee				
Q1 02	Northeast	De Le Torre	Sandra	\$ 60,125	\$ 46,020	\$14,105
		Kelly	Laura	\$ 37,129	\$ 28,023	\$9,106
		Kieferson	Jack	\$ 38,169	\$ 28,998	\$9,171
		Sawyer	Leanne	\$ 38,980	\$ 29,328	\$9,652
		Sonder	Melanie	\$ 42,125	\$ 32,049	\$10,076
		Yager	Beth	\$ 39,651	\$ 30,058	\$9,593
		Total for the Region Northeast			\$ 256,179	\$ 194,476
South	Conner	Beatrice	\$ 46,307	\$ 34,987	\$11,320	
	Nelson	Arthur	\$ 47,746	\$ 36,308	\$11,438	
	Pierce	Charles	\$ 53,195	\$ 40,429	\$12,766	
	Total for the Region South			\$ 147,248	\$ 111,724	\$35,524
Total for the Quarter Q1 02			\$ 586,599	\$ 445,504	\$141,095	
Q2 02	Northeast	De Le Torre	Sandra	\$ 66,824	\$ 50,833	\$15,991
		Kelly	Laura	\$ 45,174	\$ 34,287	\$10,887
	South	Pierce	Charles	\$ 76,334	\$ 57,755	\$18,579
		Total for the Region South			\$ 208,663	\$ 157,539
Total for the Quarter Q2 02			\$ 847,940	\$ 641,025	\$206,915	
Grand Total			\$ 1,434,539	\$ 1,086,529	\$348,010	



This report is saved as Custom Subtotals.

Subtotals and memory usage

The amount of memory required to calculate and store subtotals can be significant. In some cases, the size of the subtotals can surpass the size of the report result itself.

The size of the subtotals depends on the subtotalling option chosen, along with the order and the number of unique attributes. The easiest way to determine the number of subtotals being calculated is to use the Advanced Subtotals Options dialog box to check the number of result rows that are added when you select different subtotal options.

Subtotals can use a great deal of memory if the All Subtotals option in the Pages drop-down list is selected. This option calculates all possible subtotal calculations at runtime and stores the results in the report instance. MicroStrategy recommends that both users and report designers use less memory-intense options for calculating subtotals across pages, such as selecting grand totals only, or the specific subtotals that are needed.

For information on memory usage, see the *System Administration Guide*.

Sorting data on reports

Sorting allows you to order the report results to present your business information in a more informative way. For example, you can alphabetically sort country and region on a report, allowing you to quickly find a particular region. The *Basic Reporting Guide* discusses such quick sorting, which is available in the right-click menu when you select a column.

Advanced sorting allows you to create your own, more complex sorts for rows, columns, and pages. You can sort by both columns and rows at the same time. You can select the object to sort by, the sorting order (ascending or descending), the sorting criteria, and the position of the totals. The options for the sorting criteria depend on the sort object. For example, Employee can be sorted by last name, first name, Social Security Number, or the attribute ID. The sorting criteria do not have to be displayed on the report.

One type of advanced sort is multiple-key sorting, or *hierarchical sorting*, which sorts data according to multiple sorting criteria in a hierarchical manner. This means that the first criterion is the initial basis for sorting. Any ties are resolved using the second criterion, any remaining ties are resolved using the third criterion, and so on. If a tie remains after all the criteria are used, the default sort order is used as the tiebreaker. In a simple example, you can sort by ascending employee last name, then ascending employee first name. If two employees have the same last name, their first names are compared to alphabetically sort them. You can, of course, create more complex multiple-key sorting.

Sorting metrics hierarchically allows you to use group totals for sorting. The groups on the report are totaled, and the report is sorted on these totals. An example of hierarchical sorting is explained after the advanced sorting example that follows.

If a report is not sorted, the result set is displayed in the default order. If the report includes multiple attributes on the rows, the report sorts by the attribute farthest to the left. If the attributes are on the columns, the report sorts by the top attribute. The report is sorted by the attribute IDs, in ascending order, of the left or top attribute. If the report displays multiple attribute forms for a single attribute, the report still uses this default order. For more information on sorting with multiple attribute forms, see [Example: Sorting on multiple attribute forms, page 301](#).

If a project designer has created a report sort on an attribute form that is displayed on the report, that sort is used. The *Project Design Guide* discusses how to create report sorts.

A project designer can create an attribute form that is used solely for sorting, and is not displayed on the report. The report can be sorted by this attribute form by default, without having to define any advanced sorting for the report. The *VLDB Properties* chapter of the *Supplemental Reference for System Administration* discusses the Default Sort Behavior for Attribute Elements setting.

Sorting options example

Open the Advanced Sorting report, a subset of which is shown below.

Region	Employee		Quarter Metrics	Q4 03		Q3 03		Q2 03	
				Revenue	Rank	Revenue	Rank	Revenue	Rank
Central	Gale	Loren		\$ 118,093	4	\$ 36,165	2	\$ 73,570	4
	Zemlicka	George		\$ 74,852	3	\$ 40,801	3	\$ 62,429	3
	Ellerkamp	Nancy		\$ 62,581	2	\$ 54,331	4	\$ 48,054	2
	Torrison	Mary		\$ 55,606	1	\$ 35,643	1	\$ 46,788	1
Northwest	Becker	Kyle		\$ 108,545	3	\$ 67,766	3	\$ 94,331	3
	Gedot	Harriet		\$ 74,870	2	\$ 37,496	2	\$ 59,647	2
	Hall	David		\$ 47,759	1	\$ 26,625	1	\$ 49,579	1

Click **Advanced Sorting** in the **Data** menu. The rows are sorted by ascending region and descending fourth quarter 2003 revenue. The columns are sorted by the quarter ID in descending order. Return to the report and examine the sorted data. The columns are in reverse order, fourth quarter 2003 to first quarter 2002. The customized banding makes it easier to view the region separations in the rows. The regions are in alphabetical order, Central to Web. The rank metric helps you confirm that within each region, employees are sorted based on fourth quarter 2003 revenue. For example, the rank is 4, 3, 2, 1 in the Central region for Q4 03. For Q3 03, the rank is 2, 3, 4, 1.

Sorting metrics while retaining the hierarchical structure

The following procedure sorts the report by revenue, in descending order. The totals are placed at the top of each section, rather than more conventionally at the bottom.

To sort metrics hierarchically

To set up the sample report

- 1 Move **Rank** to Report Objects.
- 2 Move **Quarter** from the columns to the rows, to the left of Region.
- 3 Edit the view filter to remove Northwest and Web from the list of regions.
- 4 Add standard totals by choosing **Subtotals** from the **Data** menu, then selecting **Total** from the list of available subtotals. Click **OK** to return to the report.

To sort metrics hierarchically

- 5 Select **Advanced Sorting** from the **Data** menu. The Sorting dialog box opens.
- 6 On the **Rows** tab, click **Remove All** to delete the previous sort. At the prompt, click **Yes**.
- 7 Click **Add** to create a new sort.
- 8 Change **Sort By** to **Revenue**.
- 9 Change the **Order** to **Descending**.
- 10 Change the **Total Position** to **Top**.
- 11 Select the **Sort metrics hierarchically** check box, and choose **Total** from the drop-down list next to the check box.
- 12 Click **OK**.

The results are displayed below.

Quarter	Region	Employee	Metrics	Revenue
Total				\$ 5,170,447
Q4 02	Total			\$ 932,383
	Southeast	Total		\$ 376,461
		McClain	Sean	\$ 115,108
		Strome	Fred	\$ 108,615
		Benner	Ian	\$ 80,248
		Lynch	Sam	\$ 72,490
	Central	Total		\$ 317,558
		Gale	Loren	\$ 111,641
		Zemlicka	George	\$ 77,922
		Ellerkamp	Nancy	\$ 77,680
		Torrison	Mary	\$ 50,315
	South	Total		\$ 238,364
		Pierce	Charles	\$ 97,826
		Nelson	Arthur	\$ 80,650
		Conner	Beatrice	\$ 59,888
Q4 03	Total			\$ 927,290
	Southeast	Total		\$ 387,515
		Strome	Fred	\$ 111,800
		McClain	Sean	\$ 103,076
		Benner	Ian	\$ 88,616
		Lynch	Sam	\$ 84,023
	Central	Total		\$ 311,132
		Gale	Loren	\$ 118,093
		Zemlicka	George	\$ 74,852
		Ellerkamp	Nancy	\$ 62,581
		Torrison	Mary	\$ 55,606



This report is saved as Advanced Sorting - Hierarchical.

Notice how the report is sorted. Within the region Southeast in Q4 2002, the employees are sorted by revenue, in the order of the highest revenue producer to the lowest. Within Q4 2002, the regions are also sorted, from Southeast with \$376,461 in revenue to South with only \$238,364. The quarters are sorted, from Q4 2002 at \$932,383 to Q1 2003 at \$121, 639. The groups on the report are sorted hierarchically.

Example: Sorting on multiple attribute forms



Attribute display is an OLAP Services feature and you must have the “Use report objects window” and “Set attribute display” privileges to use this feature. For more information, see the *In-memory Analytics Guide*.

If multiple attribute forms are displayed for a single attribute, the report is still initially sorted in the default order, as set in the Attribute Editor. You

can change the sorting, as described below; sorting set for a report takes precedence over default sorting set for attribute forms. (For details on attribute forms, see the *Attributes* chapter of the *Project Design Guide*.) The attribute forms are displayed in the order in which you added the forms.

The following report contains one attribute, Item, but multiple attribute forms. The forms displayed are, from left to right, ID, Description, Long Description, and Price. The report is sorted by ID, by default.

Item			
91	Harman Kardon Digital Surround Sound Receiver	Digital Surround Sound Receiver 1000	
92	Harman Kardon AM/FM Stereo Receiver	HK3370 AM/FM Stereo Receiver	500
93	Harman Kardon Dolby Digital Receiver	AVR45 Dolby Digital Receiver	700
94	GPX Portable CD Player with Bass Boost	Portable CD Player with Bass Boost	30
95	GPX Portable CD Player with Car Kit	Portable Compact Disc Player with Car Kit	50
96	GPX CD AM/FM Cassette Recorder Karaoke Machine	CD AM/FM Cassette Recorder Karaoke Machine	100

Although the ID is displayed in the leftmost column on this sample, the order that attribute forms are displayed in the report does not affect how the report is sorted. It is placed there to allow you to more easily see how the report is sorted initially. Even if the ID was not displayed on the report, the report would be sorted by ID by default.

The same report is shown below, sorted by Price, then Description. Notice the order of the items with a price of \$50.

Item			
153	RCA Indoor TV Antenna	ANT120 Indoor TV Antenna	15
121	Microsoft Basic Mouse V1.0	Basic Mouse V1.0	25
94	GPX Portable CD Player with Bass Boost	Portable CD Player with Bass Boost	30
151	GPX 5" AM/FM Portable TV	TVP7 5" AM/FM Portable TV	50
95	GPX Portable CD Player with Car Kit	Portable Compact Disc Player with Car Kit	50
123	New Media 16 MB Compact Flash	16 MB Compact Flash	50
154	RCA Power TV Antenna	ANT200 Power TV Antenna	50
122	Microsoft Natural Keyboard Elite V2.0	Natural Keyboard Elite V2.0	60
133	PNY 32 MB Memory Card	32 MB 168 Pin SDRAM DIMM PC-100 Memory	60
134	PNY 64 MB Memory Card	32 MB 72 pin SIMM EDO 60NS Memory	69

The following procedure re-creates the example report shown above.

To set up the sample report

- 1 On the **File** menu, point to **New**, and then select **Report**. The Report Editor opens.

If the New Grid dialog box is displayed, click the **Empty Report** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Add **Item** to the report.
- 3 Create a report filter for Category = Electronics.

To add attribute forms and change their display order

- 4 Select **Attribute Display** from the **Data** menu. The Attribute Display dialog box opens.
- 5 Select **Use the following attribute forms**.
- 6 In the **Available forms** list, select **ID** and click **>** to add it to the **Displayed forms** and **Report objects forms** lists. Repeat with **Long Desc** and **Price**.
- 7 Re-arrange the forms in the **Displayed forms** list, using the up and down arrows on the right side of the Displayed forms box, until the forms are in the following order:
 - ID
 - Desc
 - Long Desc
 - Price
- 8 Click **OK** to return to the report.
- 9 Save the report and then execute it.

To sort by attribute forms

- 10 From the **Data** menu, select **Advanced Sorting**. The Sorting dialog box opens.

11 On the **Rows** tab, click **Add** to create a new sort.

12 Change **Criteria** to **Price**.

13 Click **Add** to create another sort.

14 Change **Criteria** to **Desc**.

15 Click **OK**. The report is now sorted as shown above.

This same report is shown below, now sorted by Price, then Long Description. Notice the order of the items with a price of \$50. In the previous sort, the first \$50 item was the portable TV, followed by the portable CD player. Now it is the compact flash, followed by the power TV antenna.

Item			
153	RCA Indoor TV Antenna	ANT120 Indoor TV Antenna	15
121	Microsoft Basic Mouse V1.0	Basic Mouse V1.0	25
94	GPX Portable CD Player with Bass Boost	Portable CD Player with Bass Boost	30
123	New Media 16 MB Compact Flash	16 MB Compact Flash	50
154	RCA Power TV Antenna	ANT200 Power TV Antenna	50
95	GPX Portable CD Player with Car Kit	Portable Compact Disc Player with Car Kit	50
151	GPX 5" AM/FM Portable TV	TVP7 5" AM/FM Portable TV	50
133	PNY 32 MB Memory Card	32 MB 168 Pin SDRAM DIMM PC-100 Memory	60
122	Microsoft Natural Keyboard Elite V2.0	Natural Keyboard Elite V2.0	60
134	PNY 64 MB Memory Card	32 MB 72 pin SIMM EDO 60NS Memory	69

To change the attribute form sort

- 1 From the **Data** menu, select **Advanced Sorting**. The Sorting dialog box opens.
- 2 On the **Rows** tab, click **Desc** in the **Criteria** column. From the drop-down list, choose **Long Desc**.
- 3 Click **OK**. The report is re-sorted, as shown above.

If you remove Price from the report display, but not from the report, the sort does not change. The report is still sorted by Price, then Long Description.

To remove an attribute form from the report display

- 1 In Report Objects, right-click **Item**, point to **Attribute Forms**, and then select **Price**. Price is no longer displayed on the report, but remains in the Report Objects.

If you remove Price from the report, the report can no longer be sorted in the same way. It is re-executed, and sorted by Long Description only, as shown below. Notice that the first item is now a CardBus as opposed to the TV antenna of the previous sample.

Item	
129 3Com 10/100 CardBus	10/100 CardBus
135 PNY 128MB Memory Card	128MB 168 Pin DIMM SDRAM PC-100 Memory
143 Xerox XD / XL Series Toner Cartridge	13R551 XD Series/XL Series Drum Cartridge
144 Xerox XE Series Toner Cartridge	13R553 XE60/XE80/XE90 Series Drum Cartridge
162 Sharp 13" 2-Head TV/VCR	13VT-L150 13" 2-Head TV/VCR
123 New Media 16 MB Compact Flash	16 MB Compact Flash
98 Pioneer 25-Disc Changer with Remote Control	25-Disc Changer with Remote Control
161 Sharp 25" Stereo Color TV	25L-S180 25" Stereo Color TV
160 Sharp 25" TV/VCR Combo	25VT-K100 25" TV/VCR Combo
133 PNY 32 MB Memory Card	32 MB 168 Pin SDRAM DIMM PC-100 Memory
134 PNY 64 MB Memory Card	32 MB 72 pin SIMM EDO 60NS Memory

To remove an attribute form from the report

- 1 In Report Objects, right-click **Price** and select **Remove from Report**. A message is displayed, indicating that the report must be re-executed for this manipulation to take effect.
- 2 Click **Yes**. The report is re-executed and displayed as shown above.

Sorting with null values

You can specify a value to use to replace null values when the report is sorted. For example, metric values on a report are sorted in descending order but blank values (or nulls) appear first, as shown below:

Item	Day	1/1/2007	1/2/2007
	Metrics	Revenue	Revenue
Art As Experience			\$11
Hirschfeld on Line			
Adirondack Style			
Architecture : Form, Space, & Order		\$84	
100 Places to Go While Still Young at Heart		\$46	
Ways of Seeing		\$22	
The Painted Word		\$18	\$6
Cabin Fever : Rustic Style Comes Home		\$16	
500 Best Vacation Home Plans		\$8	

You can instead specify that the null values are treated as zeros, so that the null values will then display at the bottom of the report. The null values still appear as blanks on the report, as shown below:

Item	Day	1/1/2007	1/2/2007
	Metrics	Revenue	Revenue
Architecture : Form, Space, & Order		\$84	
100 Places to Go While Still Young at Heart		\$46	
Ways of Seeing		\$22	
The Painted Word		\$18	\$6
Cabin Fever : Rustic Style Comes Home		\$16	
500 Best Vacation Home Plans		\$8	
Art As Experience			\$11
Hirschfeld on Line			
Adirondack Style			

To specify a replacement value for null values when the report is sorted

- 1 From the **Data** menu, select **Report Data Options**. The Report Data Options dialog box opens.
- 2 In the list of Categories on the left, expand **Display**, then select **Null Values**.
- 3 Clear the **Use Default** check box under Null Display settings.

- 4 Select the **Set the value to be used in place of empty values when the report data is sorted** check box.
- 5 Type the replacement value (such as 0) in the text box below the check box.
- 6 Click **OK** to save the change and return to the report.
- 7 If a prompt to re-execute the report is displayed, click **Yes**.
- 8 Save the report.

Sorting the page-by list

Pages group report data into logical subsets, and allow you to view one subset (or page) at a time. Page-by makes viewing a report easier, since users do not have to scroll through long lists of data. You can specify how the pages are displayed in the drop-down list of pages, by selecting the sorting order, the sorting criteria (for example, the attribute form on which to sort), and the position of the totals in the page list.

For example, the following report contains Region, Employee, the Revenue metric, and the Rank metric. It is paged by Quarter, so that a user can select a particular quarter to display. Notice that the Total is displayed first in the list, followed by quarters from the earliest to the most recent. Sorting the

page-by list allowed the report designer to select the order of the quarters and whether the total is displayed at the beginning or the end of the list.

Quarter: 2009 Q4					
2009 Q4			Revenue	Rank	
F	Total				
	2007 Q1		\$84,781	1	
	2007 Q2				
	2007 Q3		\$204,218	3	
	2007 Q4		\$209,771	4	
	2008 Q1		\$104,742	2	
	2008 Q2		\$603,512	1	
	2008 Q3				
	2008 Q4		\$57,971	2	
	2009 Q1		\$85,634	3	
C	2009 Q2		\$48,364	1	
	2009 Q3		\$191,968	1	
	2009 Q4				
	Conner	Beatrice	\$204,741	2	
	Nelson	Arthur	\$191,403	1	
	Pierce	Charles	\$242,298	3	
	Total		\$638,442	1	
	M	Benner	Ian	\$59,511	1
		Lynch	Sam	\$68,691	4
		McClain	Sean	\$66,503	2
Strome		Fred	\$68,076	3	
Total		\$262,781	1		
South	Walker	Robert	\$852,245	1	
	Total		\$852,245	1	
Total			\$2,548,949	1	

The following procedure describes how to recreate this example by starting with the Advanced Sorting report.

To sort a page-by list

To set up the sample report

- 1 Run the **Advanced Sorting** report.
- 2 Move **Quarter** to the page-by area.
- 3 Add standard totals by choosing **Subtotals** from the **Data** menu, then selecting **Total** from the list of available subtotals. Click **OK** to return to the report.
- 4 Click the page-by field and note that the quarters are sorted from the most recent date to the earliest date, with Total at the end of the list.

To sort the page-by list

- 5 Select **Advanced Sorting** from the **Data** menu. The Sorting dialog box opens.
- 6 On the **Rows** tab, click **Remove All** to delete the previous sort. At the prompt, click **Yes**.
- 7 Click the **Pages** tab. Notice that the attribute that the report is paged by, **Quarter**, is displayed in the **Sort By** column.
- 8 Change the **Order** to **Ascending**.
- 9 Change the **Total Position** to **Top**.
- 10 Click **OK** to return to the report.

Click the page-by field and note that the quarters are sorted from the earliest quarter to the latest quarter, with the total at the beginning of the list.

You can sort the page list of any page field on the report. For example, on the report created above, move Region to the page-by area. By default, the page list is sorted in ascending alphabetical order, from Central to Web. You can change the order to descending alphabetical order, or sort and order by the attribute ID rather than description. The following report sample shows the region page list sorted by ID, in descending order.

Quarter: 2008 Q4		Region: Northwest	
Employee		Metrics	
Becker	Kyle	Northwest	
Gedot	Harriet	Web	
Hall	David	Northwest	
Total		South	
		Central	
		Southeast	
		\$176,215	1

Formatting a report

You can change the general presentation formats and formatting details of a report to suit your requirements and preferences. You can set various formatting options for row and column headers, as well as for the actual

report data. You also can set borders and patterns. For formatting basics, see the *Report Formatting* chapter of the *Basic Reporting Guide*.

Formatting report cell data

The Format Cells dialog box consists of the following tabs:

- **Number:** Allows you to select the number formatting options, such as decimal spaces, currency symbol, time format, zip code format, and so on. If none of the built-in number formats meet your needs, you can create your own custom number formats using number format symbols. For more details on custom formatting, see [Custom formats, page 310](#).
- **Alignment:** Determines how the contents of the section are aligned when the formatting is applied. You can select horizontal and vertical alignment, as well as select if you would like to wrap the text or not.
- **Font:** Defines the text font for the selected section. You can select the font name, font style, size, color, and effects.
- **Border:** Defines how the borders are displayed for the selected section.
- **Background:** Defines how to fill the cell background. You can choose whether the background is a solid color, uses gradient colors (a two-color combination), uses a pattern, or is transparent to allow what is behind the cell to show.
- **Chart:** Applies a background pattern and color for a metric when it is displayed as a series in a graph report.



By default, the chart color that you define for a metric overrides any default color schemes for the graph report, although you can disable this metric formatting. For steps, see [Defining a graph color for metrics in MicroStrategy Developer, page 437](#).

Custom formats

Custom formats allow you to create your own formats for data in a report. You can format text, numbers, and date and time using custom formats. Once you create a custom format, you can use it in other metrics and report objects as well. Each custom format can have up to four optional sections, one each for:

- Positive numbers

- Negative numbers
- Zero values
- Text

You can specify these sections, separated by semicolons in the order listed above. If you specify only two sections, the first is used for positive numbers and zeros, and the second is used for negative numbers. If you specify only one section, all numbers use the same format.

Numeric data formatting

You can format fractions or numbers with decimal points by including appropriate digit placeholders in the custom format. This is explained in detail in the following table:



Numeric formatting strings supported in MicroStrategy can have a different effect when applied to the Big Decimal data type. For example, if the numeric format for a Big Decimal contains only number signs (#) to the left of the decimal point, numbers less than one are displayed beginning with a zero rather than a decimal point. The format `#.00` will display the number 0.43 as 0.43. Other data types display the same number in the same format as .43. For numeric formatting descriptions and examples when using the Big Decimal data type, see the *Data Types* appendix in the *Project Design Guide*.

Symbol	Description
0 (zero)	Digit placeholder. • If the number contains fewer digits than the placeholders contained in the format, the number is padded with zeros. For example, the format code <code>00000</code> will display the number 12 as 00012. • If there are more digits to the right of the decimal point than the placeholders in the format, the decimal portion is rounded to the number of places specified by the placeholders. • If there are more digits to the left of the decimal point than the placeholders in the format, the extra digits are retained. • If the format contains zeros to the left of the decimal point, numbers less than one are displayed with a zero to the left of the decimal point.

Symbol	Description
#	Digit placeholder. - This digit placeholder displays only significant digits and does not display insignificant zeros. For example, the format code ##.## will display the number 0025.630 as 25.63. - If there are more digits to the right of the decimal point than the placeholders in the format, the decimal portion is rounded to the number of places specified by the placeholders. - If there are more digits to the left of the decimal point than the placeholders in the format, the extra digits are retained. - If the format contains only number signs (#) to the left of the decimal point, numbers less than one are displayed beginning with a decimal point. The format #.00 will display the number 0.43 as .43.
?	Digit placeholder. - This digit placeholder adds spaces for insignificant zeros on either side of the decimal point so that decimal points align when formatted with a fixed-width font. - You can also use ? for fractions that have varying numbers of digits.
%	This symbol displays the number as a percentage, by multiplying the number by 100 and appending the % character.
, (comma)	Thousands separator. - If the format contains commas separated by #'s or 0's, commas separate the thousands. Note that the actual thousands separator used depends on the session locale. - A comma following a placeholder scales the number by a thousand. For example, using 0, scales the number by 1000, so that 10,000 displays as 10.
. (period)	Decimal separator. Note that the actual decimal separator used depends on the session locale.
E+, E-, e+, e-	Scientific notation. - If the format contains a scientific notation symbol to the left of a 0 or # placeholder, the number is displayed in scientific notation and an E or e is added. - The number of 0 and # placeholders to the right of the decimal determines the number of digits in the exponent. - E- and e- place a minus sign by negative exponents. E+ and e+ place a minus sign by negative exponents and a plus sign by positive exponents.

Character/text data

You can include formats for text and character data as mentioned in the following table:

Symbol	Description
"text"	Displays the text inside the quotation marks. Even if the text is a valid formatting symbol, it is treated as literal text if it appears within quotes. Use quotation marks around any character that is not a formatting symbol, including a space, the dollar sign (\$), minus sign (-), slash (/), exclamation mark (!), ampersand (&), tilde (~), curly brackets ({}), equals sign (=), less than and greater than signs (< >), and the caret (^). This ensures that the text appears correctly in both MicroStrategy Developer and MicroStrategy Web.
:	In a date/time format, the colon (:) does not need to be enclosed in double quotation marks (" "). However, to display it in a numeric format, it must be enclosed in quotes. For example, if you have an integer that must be displayed as 12:34:56, the correct format is "##":"##":"##".
*(asterisk)	This symbol repeats the next character until the width of the column is filled. Only one asterisk can be used in each format section.
_(underline)	This symbol skips the width of the next character. For example, to make negative numbers surrounded by parentheses align with positive numbers, you can include the format _) for positive numbers to skip the width of a parenthesis.

Date and time

The format codes for formatting days, months, years and time in a report are given in the following table:

Symbol	Description
m	Month number. Displays the month as digits without leading zeros, such as 1. Can also represent minutes when used with h or hh formats.
mm	Month number. Displays the month as digits with leading zeros, as in 01. Can also represent minutes when used with the h or hh formats.
mmm	Month abbreviation, such as Jan.
mmm	Month name, such as January.
d	Day number. Displays the day as digits with no leading zero, such as 1.
dd	Day number. Displays the day as digits with leading zeros, as in 01.

Symbol	Description
ddd	Day abbreviation, such as Sun.
dddd	Day name, such as Sunday.
yy	Year number. Displays the year as a two-digit number, such as 00.
yyyy	Year number. Displays the year as a four-digit number, such as 2003.
h	Hour number. Displays the hour as a number without leading zeros, such as 1. If the format contains an AM or PM format, the hour is based on a 12-hour clock; otherwise, it is based on a 24-hour clock.
hh	Hour number. Displays the hour as a number with leading zeros, as in 01. If the format contains an AM or PM format, the hour is based on a 12-hour clock; otherwise, it is based on a 24-hour clock.
m	Minute number. Displays the minute as a number without leading zeros, such as 0. The m format must appear immediately after the h or hh symbol; otherwise it is interpreted as month.
mm	Minute number. Displays the minute as a number with leading zeros, such as 00. The mm format must appear immediately after the h or hh symbol; otherwise it is interpreted as month.
s	Second number. Displays the second as a number without leading zeros, such as 0.
ss	Second number. Displays the second as a number with leading zeros, such as 00.
AM/PM am/pmA/ P a/p	12-hour time. Displays time using a 12-hour clock. Displays AM, am, A, or a to display time between midnight and noon; displays PM, pm, P, or p to display time between noon and midnight.
[h]	Displays the total number of hours.
[m]	Displays the total number of minutes.
[s]	Displays the total number of seconds.

Color

You can change the color of data in your report using custom formatting. The following table lists the format for color codes:

Symbol	Description
[Black]	Displays cell text in black.
[Blue]	Displays cell text in blue.
[Cyan]	Displays cell text in cyan.
[Green]	Displays cell text in green.
[Magenta]	Displays cell text in magenta.
[Red]	Displays cell text in red.
[White]	Displays cell text in white.
[Yellow]	Displays cell text in yellow.

Currency

You can include the following currency symbols in a number format. Keep the ALT key pressed and type the ANSI code of the currency. The ANSI code should be followed by the format code for the number.



To type ANSI code for the currency symbol, turn on NUM LOCK and use the numeric keypad. As you type the ANSI code, the Custom box appears blank. The currency symbol is displayed only when you finish typing the code.

Hold the ALT key down and type this code	To display
0162	¢
0163	£
0165	¥
0128	€

Conditional symbols

You can apply conditional formatting to monitor the data in your report.

Symbol	Description
[conditional value]	Designates a different condition for each section. For example, data in a column has values ranging from 200 to 800 and you want the text "Poor" to be displayed in black for values less than 400, the text "Good" to be displayed in Red for values greater than 600, and the text "Average" to be displayed in blue for values ranging between 400 and 600. You can use the following code for these conditions: [<400] [Black] "Poor"; [>600] [Red] "Good"; [Blue] "Average" In this example, [<400] and [>600] are the conditional values.

Custom number formatting examples

The following table lists examples of custom number formats. It includes the formatting symbols, the report data, and how that data is displayed after using the formatting.

Format	Cell data	Display
###	250.436 0.43	250.44 .43
#.0#	250.436 125	250.44 125.0
???.???	123.43, 45.90, 345.809	With aligned decimals
###0"CR";#;###0"DR"; 0	2567 -4567 0	2,567CR 4,567DR 0
#,###	1500	1,500
0,	10,000	10
"Sales="0.0	123.45	Sales=123.5
"X="0.0;"x="-0.0	-12.34	x=-12.3
"Cust. No. " 0000	1234	Cust. No. 1234
m-d-yy	2/3/03	2-3-03
mm dd yy	2/3/03	02 03 03
mmm d, yy	2/3/03	Feb 3, 03
mmm d, yyyy	2/3/03	February 3, 2003

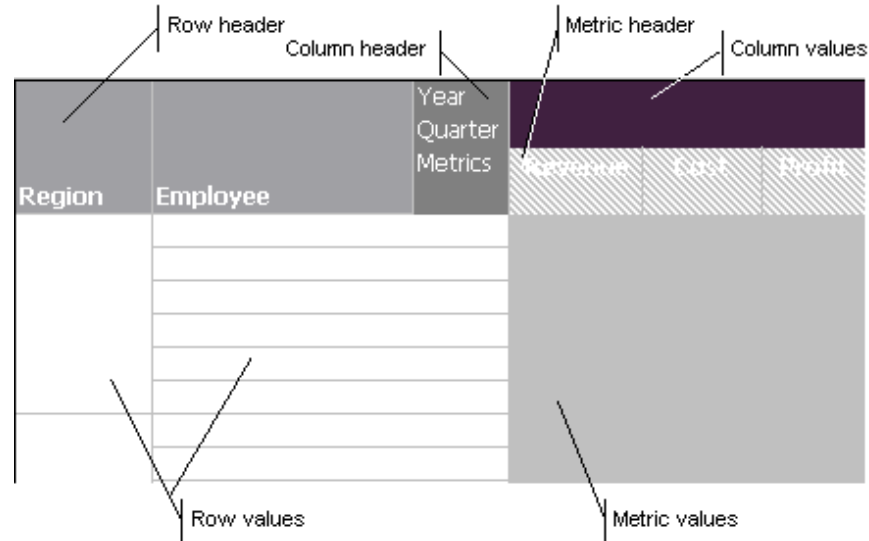
Format	Cell data	Display
d mmmm yyyy	2/3/03	3 February 2003
hh"h" mm"m"	1:32 AM	01h 32m
h.mm AM/PM	14:56	2.56 PM
#?/?	1.25	1 1/4
#?/8	1.25	1 2/8
ALT+0163 #.##	250.45	£ 250.45
###%	.08 2.8	8% 280%
\$* #,##0.00;\$* -#,##0.00	5632.567 -12.34	\$ 5,632.57 \$ -12.34
0*-	250.45	250.45----
*-0	250.45	----250.45
000-00-0000	345126789	345-12-6789
0.00E+00	10000	1.00E+04
##0.0E+0	10000	10.0E+03
0.00E+00	0.0001	1.00E-04
##0.0E-0	0.0001	100.0E-6
0.0E-00	0.0001	1.0E-04

Understanding how formatting impacts report display

Every report contains several different formatting layers, allowing you to retain control of how a report looks when it is pivoted or manipulated. You can ensure that the formatting continues to highlight the information that needs attention. There are two basic formatting layers—zones and grid units. Examples of zones are the rows headers and metric values of a report, while grid units are the values of a particular attribute or metric. The other formatting layers, such as thresholds and subtotals, can be thought of as extensions of these two basic types.

Zone formatting

The following diagram illustrates the basic formatting zones of a report. Each zone is formatted differently so that you can easily distinguish among them.



When data is manipulated in a report that is formatted by zone, the new location of the object determines what formatting is applied to it. For example, if you pivot Region from rows to columns in the preceding example, the background of the text changes from light grey to dark grey. It is now part of the column header, as shown below. The formatting of a zone does not move with the data.

Employee	Region			
	Year			
	Quarter			
	Metrics	Revenue	Cost	Profit

Grid unit formatting

Grid units are the individual attributes, metrics, and consolidations that make up a report. Unlike zone formatting, grid unit formatting is attached to the object and moves with it when the object is pivoted. For example, the following report is the same as the previous examples, except that Region has

been formatted, at the unit level. The header, Region, is now black on light grey and the values (Northeast and Mid-Atlantic) are now black on white.

Region	Employee	Year Quarter Metrics			
			Revenue	Cost	Profit
Northeast					
Mid-Atlantic					

When Region is pivoted to the column area, as in the zone formatting example, the formatting accompanies the attribute. Compare the following example with the pivot example in [Zone formatting, page 318](#).

Employee	Region	Northeast					
	Year Quarter Metrics	Revenue	Cost	Profit	Revenue	Cost	Profit

Formatting subtotals

Subtotal formatting can be applied to either zone or grid unit formatting. If the formatting is applied at the zone level, the formatting stays with that zone. If the formatting is applied at the grid unit level, when the unit is pivoted, the formatting moves with the unit.

In the following example, notice that the row subtotal formatting overwrites the column subtotal formatting.

Diagram illustrating row and column subtotal formatting. The table structure shows a grid of cells. A column subtotal header is shown in a light gray cell, and a column subtotal value is shown in an orange cell. A row subtotal header is shown in a dark blue cell, and a row subtotal value is shown in a light gray cell. The row subtotal formatting overwrites the column subtotal formatting.

Formatting thresholds

Thresholds allow conditional formatting for metric values. It is similar to unit formatting because it is data-driven. For example, the following report has a threshold set for revenue less than \$400,000.

Employee		Metrics	Revenue	Cost	Profit
De Le Torre	Sandra		\$ 514,524	\$ 391,121	\$123,403
Kelly	Laura		\$ 329,888	\$ 250,120	\$79,768
Kieferson	Jack		\$ 389,888	\$ 294,751	\$95,137
Sawyer	Leanne		\$ 316,786	\$ 240,110	\$76,676
Sonder	Melanie		\$ 421,036	\$ 318,975	\$102,061
Yager	Beth		\$ 362,742	\$ 275,208	\$87,534

Besides the basic formatting options such as font and alignment, the cell contents can be replaced with any of the following when the condition is met:

- Replacement text, which is text, such as Good Sales

- A replacement image. The destination of the image can be set using any of the following:
 - A path relative to the HTML Document directory, which is a relative path from the document directory where the image is stored (for example, `images\img.jpg`)
 - A path on the network, which is a path on the local area network, in a UNC format (`\\machine_name\shared_folder\img.jpg`)
 - A URL to an image file (`http://www.microstrategy.com/images/img.jpg`)
 - An absolute path (`c:\images\img.jpg`)



Note the following:

- If you specify the location of the image as a directory rather than a URL, you must confirm that you can access the directory over Web and MicroStrategy Developer. If not, the image will not be displayed because it will not be available over the network. This problem of network access can be avoided by referencing a URL.
- If you specify the location of the threshold image in UNC format, you cannot view threshold images when you view the report in PDF format. This is because the Internet user account does not have permissions to a file on the network. Similarly, when Intelligence Server is running on a system account, it does not have access to XSLs and HTML template files if the document directory is in a UNC format. In such cases you cannot view threshold images when you view a report in PDF format.
- A symbol chosen from a pre-defined list. In Web, these symbols are represented by an image file resembling the symbol used in MicroStrategy Developer.

For steps to format thresholds and alerts, click **Help**.

Thresholds can be applied to graph reports, as well. Only background formatting is displayed, on the series of certain graph types and on the data markers of other graph types. For a list of graph types, examples, and a procedure to enable the threshold display on a graph report, see [Displaying thresholds on graph reports, page 446](#).

Formatting consolidations and custom groups

For an introduction to custom groups and consolidations, see [Chapter 4, Custom Groups and Consolidations](#).

Formatting of consolidations and custom groups is also similar to grid unit formatting. They are both groupings of attribute elements, allowing you to qualify a report on a row-by-row basis. For custom groups, an element can be expanded into individual items. When this occurs, you can also format the item's headers and values.

Consolidation and custom group formatting is performed in the Consolidation Editor and the Custom Group Editor. You can also format consolidations as a grid unit in the Report Editor, as described in [Grid unit formatting, page 318](#). The grid unit formatting takes precedence over the consolidation/custom group formatting. The interaction between the different formatting types is explained in [Order of layers](#) below.

Order of layers

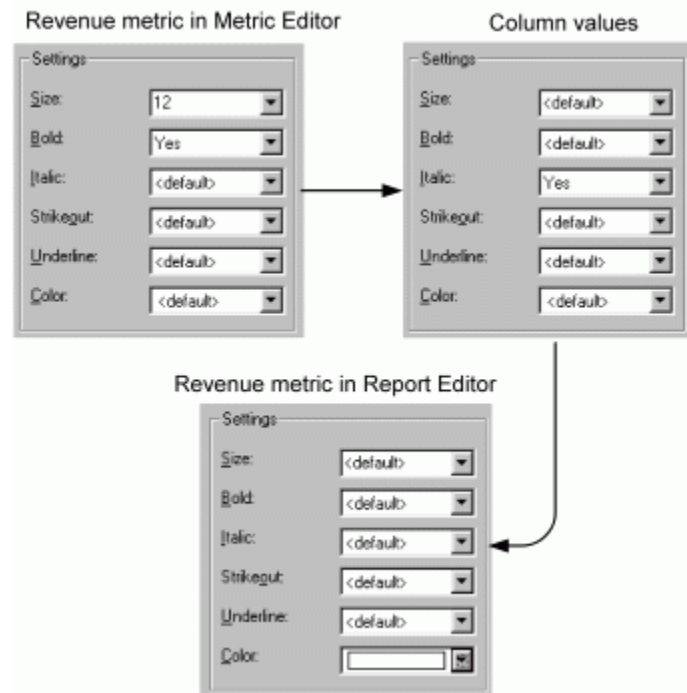
With the different types of formatting, it is important that the interaction between them is clearly defined so that display and order of processing is correct. Each succeeding layer overwrites the formatting of all its preceding layers, as shown below.

This section provides a list of all formatting types, as well as a table of each type of formatting and what formatting it overwrites.

The Basic Report, shown below, contains the Revenue metric. Use the Metric Editor to change the metric header to a bold, 12-point font. Wherever this metric is used, this header font is applied. Execute the report. The Revenue metric header appears the same as the other metric headers, because other formatting layers already set in the report overwrite the metric level formatting.

Italicize the column values and change the other font settings to default. Change the Revenue metric header to a white font. Since the **Format** menu in

the Report Editor is used for this change, the new format applies to the current report only. The formatting dialogs for each are shown below.



The completed report looks like the following:

Region	Employee	Metrics	<i>Revenue</i>	<i>Cost</i>	<i>Profit</i>
Northeast	De Le Torre Sandra		\$ 514,524	\$ 391,121	\$123,403
	Kelly Laura		\$ 329,888	\$ 250,120	\$79,768
	Kieferson Jack		\$ 389,888	\$ 294,751	\$95,137
	Sawyer Leanne		\$ 316,786	\$ 240,110	\$76,676
	Sonder Melanie		\$ 421,036	\$ 318,975	\$102,061
	Yager Beth		\$ 362,742	\$ 275,208	\$87,534

The final formatting for the Revenue metric header is a combination of the formats of the metric level header (set in the Report Editor), the column values, and the report metric.

The following list describes all the formatting layers in the order that they are applied, starting with the first layer to be applied:

- Metric** specifies a format for the particular metric, regardless of the report it is on. This formatting, which can be applied to headers and values, is performed in the Metric Editor. Metric level formatting is overwritten by axis and all metrics formatting. Setting those layers to default allows the metric level formatting to display.

To format metric values at the metric level, the all metrics values formatting must be default. To use it for metric headers, set the axis headers formatting to default.

- 2 **Axis** formatting affects all the units of the axis. This zone formatting is overwritten by grid unit formatting. The axis formatting layers are located under the **Rows** and **Columns** options on the **Format** menu.
- 3 **Metrics label** formatting applies to the text “Metrics” that is displayed on the report. It is based on the axis header formatting. For example, if the metrics are displayed on the columns of the report, then the formatting for column headers is applied to the label “Metrics”.
- 4 **All metrics** formatting has two options:
 - **All metrics headers** formats the metric names. It overwrites the axis formatting.
 - **All metrics values** formats the data zone, or where the metric values are displayed. It overwrites metric level formatting.

The **Format** menu contains the **All Metrics** option.

- 5 **Report metric** formats an individual metric on a particular report. It does not change the formatting of the metric in other reports. Report metric formatting overwrites metric level and all metrics formatting. To format a metric at the report level, select the metric on the **Format** menu.
- 6 **Grid unit** allows you to format an individual report item, such as an attribute. It overwrites axis formatting. Every grid unit is listed on the **Format** menu.
- 7 **Banding** enables row or column grouping by color to enhance readability. Banding formats are applied before subtotal formatting to allow subtotals to take priority. Select **Grid**, then **Options** to create banding.
- 8 **Column subtotals** formatting is overwritten by row subtotals when column and row subtotals intersect. Subtotal formatting can be applied as either zone or grid unit formatting:
 - In grid unit formatting, the formatting moves with the subtotals. Every grid unit is listed on the **Format** menu.
 - In zone formatting, formatting is based on location. To format subtotals as a zone, select **Columns** from the **Format** menu, then choose **Subtotal Headers** or **Values** in the drop-down menu.

The subtotal headers lie on the values zone. Formatting is applied on subtotal headers in the following order, starting with the first layer to be applied:

- a Axis values
- b Axis subtotal header
- c Template unit values
- d Template unit subtotal header

- 9 Row subtotals** formatting takes precedence over column subtotals when the two intersect. As with column subtotals, it can be applied as either zone or grid unit formatting.

The subtotal headers lie on the values zone. Formatting is applied on subtotal headers in the following order, starting with the first layer to be applied:

- a Axis values
- b Axis subtotal header
- c Template unit values
- d Template unit subtotal header

- 10 Custom group** formatting allows you to format each custom group element differently, independent of the report format. Set this format in the Custom Group Editor.

- 11 Consolidation** elements can also be formatted individually. This formatting applies to any report the consolidation is placed on. To format a consolidation, use the Consolidation Editor.

- 12 Report border** creates a border around the whole report.



To set a report border, right-click the report but not a report object. Select **Formatting**, then **Grid Borders**.

- 13 Threshold** is the last layer applied so it overwrites all other layers.

The following table contains a matrix of each formatting layer and the layers that overwrite it.

This layer...	Is overwritten by these layers...
Metric object - headers	Axis headers, grid unit headers, all metrics headers, report metric headers, column subtotal headers, row subtotal headers
Metric object - values	Axis values, grid unit values, all metrics values, report metric values, banding, column subtotal values, row subtotal values, custom group values, consolidation values, thresholds
Axis - headers	Grid unit headers, all metrics headers, report metric headers, column subtotal headers, row subtotal headers
Axis - values	Grid unit values, all metrics values, report metric values, banding, column subtotal values, row subtotal values, custom group values, consolidation values, thresholds
All metrics - headers	Report metric headers, either the column subtotal headers (if the metrics are in the columns) or row subtotal headers (if the metrics are in the rows)
All metrics - values	Report metric values, banding, column subtotals, row subtotals, custom group values, consolidation values, thresholds
Report metric	Banding, column subtotals, row subtotals, custom group values, consolidation values, thresholds
Grid unit - headers	Column subtotal headers, row subtotal headers
Grid unit - values	Banding, column subtotal values, row subtotal values, custom group values, consolidation values, thresholds
Banding	Column subtotals, row subtotals, custom group values, consolidation values, thresholds
Column subtotals	Row subtotals, thresholds
Row subtotals	Custom group values, consolidation values, thresholds
Report border	None
Custom group - headers	Consolidation headers, thresholds
Custom group - values	Consolidation values, thresholds
Custom group child - headers	Consolidation headers, thresholds
Custom group child - values	Consolidation values, thresholds
Consolidation - headers	None
Consolidation - values	Thresholds
Thresholds	None

Order of layers for metric headers displayed under the parent subtotal headers

The metric headers on a report may be displayed under the parent subtotal header (the parent subtotal is the attribute). For example, if both the attributes and the metrics are placed in the columns, the metric is displayed below the attribute subtotal, as shown below:

	Year	2007	2008	2009	Total
	Metrics	Profit	Profit	Profit	Profit
		\$1,304,141	\$1,740,085	\$2,249,397	\$5,293,624

If both the attributes and the metrics are placed in the rows, the metric is displayed to the right of the attribute subtotal, as shown below:

Year	Metrics	
2007	Profit	\$1,304,141
2008	Profit	\$1,740,085
2009	Profit	\$2,249,397
Total	Profit	\$5,293,624

When the metric headers are displayed under the parent subtotal header, the formatting is applied to the metric header in the following order:

- 1 Grid unit formatting of the template unit object for its parent subtotal element, with all regular rules for grid formatting applied
- 2 Metric header
- 3 Column or row subtotal formatting

Therefore, the metric header formatting in the subtotal column or row is inherited from the parent subtotal header formatting only if the subtotal formatting is not set to default.

For example, the Year subtotal header and values have been formatted to display white text on a black background, as shown below. The metric header (the word Profit) in the subtotal column is formatted the same as the Year subtotal header.

	Year	2007	2008	2009	Total
	Metrics	Profit	Profit	Profit	Profit
		\$1,304,141	\$1,740,085	\$2,249,397	\$5,293,624

If the Year subtotal header formatting is set to default instead, then the metric header formatting in the subtotal column is inherited from the metric header. This is shown below, where the Year subtotal header and metric header are displayed in blue text on a grey background, just as the other headers are.

	Year	2007	2008	2009	Total
	Metrics	Profit	Profit	Profit	Profit
		\$1,304,141	\$1,740,085	\$2,249,397	\$5,293,624

The same formatting order applies when the report contains the attributes and the metrics in the rows. When the subtotal header formatting is set to default, then the metric header formatting is inherited from the metric header. In the example below, the metric header is italicized, but the subtotal header is set to default. Thus the metric header in the subtotal row (the word Profit in the Total row) inherits the formatting from the metric header and is italicized.

Year	Metrics	
2007	<i>Profit</i>	\$1,304,141
2008	<i>Profit</i>	\$1,740,085
2009	<i>Profit</i>	\$2,249,397
Total	<i>Profit</i>	\$5,293,624

Change the formatting of the Year subtotal header to bold text. The metric header in the subtotal row (the word Profit in the Total row) now displays in a bold italicized font. The italics are inherited from the metric header, since all the font options of the subtotal header are set to default.

Year	Metrics	
2007	<i>Profit</i>	\$1,304,141
2008	<i>Profit</i>	\$1,740,085
2009	<i>Profit</i>	\$2,249,397
Total	<i>Profit</i>	\$5,293,624

Autostyles: Report formatting styles

Autostyles provide predefined formatting styles, allowing you to standardize formatting among reports. Each autostyle is a collection of all the formatting layers, allowing you to format the different report sections. However, not every layer must be configured to create an autostyle. Default formatting

values are supplied by the `guiprop.pds` file. For more information on defaults, see [Appendix B, Formatting Default Values](#). Each formatting layer contains all the formatting options:

- Font
- Alignment
- Border
- Pattern



HTML cannot display patterns as the background of a cell. Therefore, the patterns do not display in Web reports.

Notice that autostyles do not include number formatting. Numbers are usually formatted at a different level, such as the metric level. Retaining the report-level format allows your selected number format to remain.

Preconfigured autostyles are included in MicroStrategy Developer and MicroStrategy Web, but you can create your own as well. If you do, design at the lowest layer possible, not at the grid unit level. If a formatted grid unit does not appear on a report, that formatting also does not appear on the report.

To share your own autostyles to users, place them in the Autostyles folder under the Public Objects folder. Other MicroStrategy Developer and Web users on the system can apply any autostyle saved in that location.



After an autostyle is placed in the folder, Web users must log out and then log in to refresh the autostyles list.

Find and replace report, template, and metric formatting

Find and replace allows you to globally change reports, templates, and metric formatting. You can replace autostyles, Report Data Options, metric formatting, and graph fonts for a set of reports, templates, and metrics.

This feature is helpful when a report design changes. For example, suppose the formatting of all your inventory reports must be changed. Create a new autostyle with the changes, search for only inventory reports, and substitute the new autostyle. Find and replace autostyles allows you to quickly switch the formatting of the selected reports.

The feature also is useful when standardizing the appearance of a set of metrics across the project. However, changes to metric formatting do not overwrite the format of such metrics set by individual reports or templates.

For steps and a description of the Find and Replace interface, click **Help**.



Note the following:

- Before you can use the Find and Replace interface, you must have the Use Find and Replace privilege. For more information on privileges, see the *Supplemental Reference for System Administration*.
- Replacing autostyles or Report Data Options invalidates the report caches. A new cache is created when the report is run for the first time after the replacement.
- A report (or template) records the ID of the style when the style is applied, but it is possible that the style was changed after it was applied. In this case, the report is using the Custom style because the definition has changed. When Find and Replace is run, the report using the target autostyle, even if it has been changed and is displayed in the Report Editor as Custom, will be updated with the new autostyle.

Displaying attribute and attribute form headers in a grid

You can format the display of column or row headings (headers) for attributes and attribute forms in a grid. For example, you can choose to have a header containing the attribute form name automatically displayed above each attribute form shown in the grid, or have a single header automatically displayed for each attribute in the grid, with each header containing only the attribute name. The following images depict examples of each way in which you can choose to have attribute and attribute form headers displayed.

You can have a header containing the attribute name automatically displayed for each attribute in the grid.

Customer		Revenue	
Aaby	Alen	7796	\$3,104
Aadland	Miko	1874	\$3,814
Aadland	Warner	3771	\$2,378
Aadland	Constant	4432	\$4,508
Aafedt	Wendy	7923	\$1,064
Aagesen	Bink	1930	\$2,580
Aalgaard	Kenney	3345	\$1,551
Aamodt	Stacy	7632	\$786
Aarestad	Benjamine	2306	\$1,857
Aarnink	Marlan	7570	\$1,657

You can have a header containing the attribute name and attribute form name automatically displayed above each attribute form shown in the grid.

Customer Last Name	Customer First Name	Customer ID	Revenue
Aaby	Alen	7796	\$3,104
Aadland	Miko	1874	\$3,814
Aadland	Warner	3771	\$2,378
Aadland	Constant	4432	\$4,508
Aafedt	Wendy	7923	\$1,064
Aagesen	Bink	1930	\$2,580
Aalgaard	Kenney	3345	\$1,551
Aamodt	Stacy	7632	\$786
Aarestad	Benjamine	2306	\$1,857
Aarnink	Marlan	7570	\$1,657

You can have a header containing the attribute form name automatically displayed above each attribute form shown in the grid.

Last Name	First Name	ID	Revenue
Aaby	Alen	7796	\$3,104
Aadland	Miko	1874	\$3,814
Aadland	Warner	3771	\$2,378
Aadland	Constant	4432	\$4,508
Aafedt	Wendy	7923	\$1,064
Aagesen	Bink	1930	\$2,580
Aalgaard	Kenney	3345	\$1,551
Aamodt	Stacy	7632	\$786
Aarestad	Benjamine	2306	\$1,857
Aarnink	Marlan	7570	\$1,657

You can choose to have a header automatically displayed for each attribute form in the grid. Only the header for the first attribute form for each attribute

includes the attribute name. All other headers contain the attribute form name only.

Customer Last Name	First Name	ID	Revenue
Aaby	Alen	7796	\$3,104
Aadland	Miko	1874	\$3,814
Aadland	Warner	3771	\$2,378
Aadland	Constant	4432	\$4,508
Aafedt	Wendy	7923	\$1,064
Aagesen	Bink	1930	\$2,580
Aalgaard	Kenney	3345	\$1,551
Aamodt	Stacy	7632	\$786
Aarestad	Benjamine	2306	\$1,857
Aarnink	Marlan	7570	\$1,657

You can choose to automatically display either headers for each attribute or each attribute form depending on the number of attribute forms visible in the grid for each attribute.

Region	Customer Last Name	Customer First Name	Customer ID	Revenue
Central	Aadland	Warner	3771	\$2,317
	Aadland	Constant	4432	\$4,033
	Aagesen	Bink	1930	\$2,580
	Aamodt	Stacy	7632	\$710
	Aaron	Ferrell	7455	\$4,695
Mid-Atlantic	Aalgaard	Kenney	3345	\$1,416
	Aarestad	Benjamine	2306	\$1,857
Northeast	Aadland	Miko	1874	\$3,747
	Aaronsen	Maxwell	1	\$1,162

Detailed descriptions of each scenario are below, as well as steps to have attribute and attribute form headers automatically displayed in a grid in each of the ways described above.

- You can have a single header automatically displayed for each attribute in the grid, with each header containing only the attribute name. The grid in the first image above contains the Customer attribute, for which the Last Name, First Name, and ID attribute forms are each shown in a separate column. In this example, the designer of the grid has chosen to have the attribute name automatically displayed in the header of each attribute in the grid, so the header for the Customer attribute is displayed as Customer.
- You can have a header automatically displayed for each attribute form in the grid, with each header consisting of the attribute name followed by the attribute form name. For example, in the second image above, the headers for the Last Name, First Name, and ID attribute forms are

displayed as Customer Last Name, Customer First Name, and Customer ID, respectively.

- You can have a header automatically displayed for each attribute form in the grid, with each header consisting of only the attribute form name. In the third image above, the headers for the Last Name, First Name, and ID attribute forms are displayed as Last Name, First Name, and ID, respectively.
- You can have a header automatically displayed for each attribute form in the grid and include the attribute name only in the header for the first attribute form for each attribute. In the fourth image above, the Last Name attribute form is the first attribute form displayed for the Customer attribute, and its header is therefore displayed as Customer Last Name. The remaining attribute forms are displayed using the attribute form name only: First Name and ID, respectively.
- You can choose to automatically display either headers for each attribute or each attribute form depending on the amount of attribute forms visible in the grid for each attribute. If only one attribute form is shown in the grid for an attribute, the attribute is displayed with a header containing the attribute's name. If more than one of the attribute's forms are visible in the grid, each attribute form is displayed with a header containing the attribute name followed by the attribute form name.

In the fifth image above, both the Region and Customer attributes have been added to the grid. Because the Region attribute is displayed using a single attribute form, it is displayed with a header containing the attribute name, Region. On the other hand, three attribute forms are displayed for the Customer attribute, so a header is displayed for each attribute form: Customer Last Name, Customer First Name, and Customer ID, respectively.

Prerequisites

- This procedure assumes that you have already created the report whose attributes and attribute form headers you want to display.
- The Show Attribute Form Names option in the Grid Display preferences in Web must be set to Read from Report. For steps to modify user preferences in Web, see the *MicroStrategy Web Help*.

To display attribute and attribute headers in a grid

- 1 In MicroStrategy Web, click the name of the report to run it.

- 2 From the **Tools** menu, select **Report Options**. The Report Options dialog box opens.
- 3 On the **General** tab, from the **Show attribute form names** drop-down list, select one of the following (images and detailed descriptions of each option are above):
 - To have the attribute name automatically displayed in the header of each attribute in the grid, select **Off** (default). No attribute form names are included in the grid, as shown in the first example image above.
 - To have a header automatically displayed for each attribute form in the grid, with each header consisting of the attribute name followed by the attribute form name, select **On**. For an example, see the second image in the section above.
 - To have a header automatically displayed for each attribute form in the grid, with each header consisting of only the attribute form name, select **Form name only**. For an example, see the third image in the section above.
 - To have a header automatically displayed for each attribute form in the grid and have the attribute name included only in the header for the first attribute form for each attribute, select **Show attribute name once**. The remaining attribute forms are displayed using the attribute form name only, as shown in the fourth example image above.
 - To automatically display either headers for each attribute or each attribute form depending on the number of attribute forms visible in the grid for each attribute, select **Automatic**. If only one attribute form is shown in the grid for an attribute, the attribute is displayed with a header containing the attribute's name. If more than one of the attribute's forms are visible in the grid, each attribute form is displayed with a header containing the attribute name followed by the attribute form name. See the fifth image above for an example.
- 4 Click **OK** to apply your changes.

Determining which attribute elements are displayed: Attribute joins

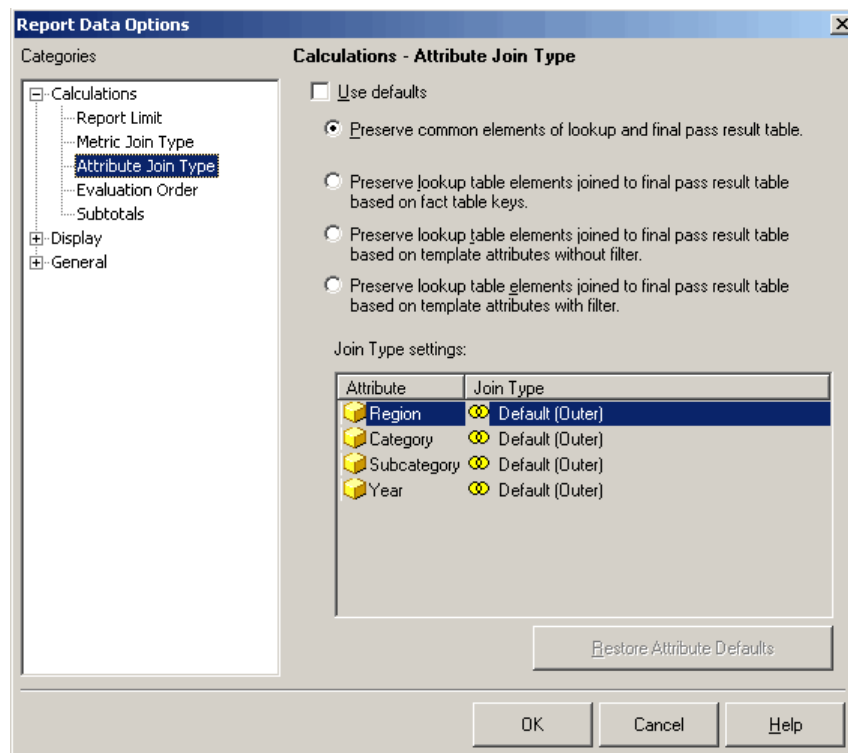
For business attribute data, the most common types of joins are inner joins and outer joins. An inner join includes in the data calculation only the data common to all the attribute elements in the join, while an outer join includes all of the data in all of the attribute elements.



Understanding the details of attribute joins requires that you have a basic understanding of SQL and of the tables in your data warehouse.

You can create an outer join on data coming from your data warehouse tables, for certain attributes on a report. You can also choose how the data warehouse table and final SQL pass result table are joined, and then determine the type of join, inner or outer, between specific attributes.

The image below shows the attribute join type setting in the Report Data Options dialog box. Each join type is described with an example in the next section, [Understanding attribute join types, page 336](#).



Understanding attribute join types

Each attribute join type affects what attribute elements are displayed on the report. You can display only those attribute elements that appear in the related lookup and fact tables in your data warehouse; or you can display all attribute elements that exist in the lookup tables in your data warehouse, whether or not there is corresponding data in the related fact tables. A detailed description of each attribute join type is provided in this section.

Examples for each attribute join type are also provided. The examples are based on the following simple example data in a sample data warehouse.

Store table (lookup table)

Store ID	Store Name
1	East
2	Central
3	South
6	North

Sales table (fact table)

Store ID	Year	Dollar Sales
1	2005	1000
2	2005	2000
3	2005	5000
1	2006	4000
2	2006	6000
3	2006	7000
4	2006	3000
5	2006	1500

The Sales table has data for Store IDs 4 and 5, but the Store table does not have any entry for these two stores. Notice that the North Store (Store ID 6) does not have any entries in the Sales table. This sample data is used in the examples below to show how the different attribute join types work.

For example, your report contains the attribute Store and a Sales metric defined as Sum(Sales), which is the sum of the data in the Sales fact table. In your data warehouse, it is possible that a store does not have any sales at all. However, you want to show all store names in the final report, even the names of those stores that do not have sales. To do this, you cannot rely on the stores in the sales fact table. Instead, you must make sure that all the stores from the lookup table are included in the final report. In SQL terms, the SQL Engine must perform a left outer join from the lookup table to the fact table.

Preserve common elements of lookup and final pass result table

This attribute join type is the default setting. It causes the SQL Engine to calculate for only the attribute elements that exist in both the lookup and fact tables. This join type lets you display only those attribute elements that are common to the lookup tables and the fact tables in your data warehouse.



You cannot change the join types of individual attributes on the report by using this setting. The attributes and their join types are displayed for information only.

Preserve lookup table elements joined to final pass result table based on fact table keys

This attribute join type causes the SQL Engine to calculate for only the attribute elements that are joined to the fact table based on fact table keys.

Sometimes the level of the fact table is not the same as the level of the report. For example, a report contains the attributes Store and Month and the Sales metric which is based on the Sales fact table. The report level is Store and Month, but the fact table level is Store, Day, and Item. Different methods to keep all the store and month attribute elements so they are displayed on the report are described below:

- Perform a left outer join first to keep all attribute elements at the Store, Day, and Item level, then aggregate the data to the Store and Month level.
- Perform the data aggregation first, then perform a left outer join to bring in all attribute elements.

This attribute join type uses the first approach. Working with the example above, the SQL Engine makes two SQL passes:

- **Pass 1:** LOOKUP_STORE cross join LOOKUP_DAY cross join LOOKUP_ITEM; inserted into temporary table TT1
- **Pass 2:** TT1 left outer join Fact_Table on (store, day, item)

The advantage of this approach is that you can perform a left outer join and aggregation in the same pass (pass 2). The disadvantage is that because you perform a Cartesian join with the lookup tables at a much lower level (pass 1), the result of the Cartesian joined table (TT1) can be very large.

Preserve lookup table elements joined to final pass result table based on template attributes without filter

This attribute join type causes the SQL Engine to calculate for all attribute elements and ignores all related filtering conditions.

Sometimes the level of the fact table is not the same as the level of the report. For example, a report contains the attributes Store and Month and the Sales metric which is based on the Sales fact table. The report level is Store and Month, but the fact table level is Store, Day, and Item. Different methods to keep all the store and month attribute elements so they are displayed on the report are described below:

- Perform a left outer join first to keep all attribute elements at the Store, Day, and Item level, then aggregate the data to the Store and Month level.
- Perform the data aggregation first, then perform a left outer join to bring in all attribute elements.

This attribute join type uses the second approach. The SQL Engine makes three SQL passes:

- **Pass 1:** Aggregate the Fact_Table to TT1 at Store and Month. This is actually the final pass of a normal report when this setting is not enabled.
- **Pass 2:** LOOKUP_STORE cross join LOOKUP_MONTH; inserted into temporary table TT2
- **Pass 3:** TT2 left outer join TT1 on (store, month)

This approach requires one more pass than the previous join type setting (**Preserve lookup table elements joined to final pass result table based on fact table keys**), but the cross join table (TT2) is usually smaller.

Preserve lookup table elements joined to final pass result table based on template attributes with filter

This attribute join type causes the SQL Engine to calculate for all attribute elements and applies all related filtering conditions.

This attribute join type is similar to the previous join type (**Preserve lookup table elements joined to final pass result table based on template**

attributes without filter). The only difference is that the report filter is applied in the final pass (Pass 3). For example, a report contains Store, Month, and Sum(Sales) with a filter of Year = 2005. You want to display every store in every month in 2005, regardless of whether there are sales. However, you do not want to show any months from other years, but only the 12 months in the year 2005. This attribute join type provides this data.

Selecting an attribute join type

You can determine attribute join types in the following ways:

- Use a VLDB property setting to determine attribute join types for a given database instance, report, or report template. See details about the Preserve All Lookup Table Elements VLDB property in [Preserve All Lookup Table Elements, page 869](#).
- Use the Report Data Options dialog box to individually select attributes on the template that need to display attribute elements. When you change from the default attribute join type setting (**Preserve common elements of lookup and final pass result table**), it is assumed that you want to keep ALL elements of the attributes in their lookup tables. However, sometimes you want such a setting to affect only some of the attributes on a report. For example, for a report containing Store, Month, and Sum(Sales), you may want to show all the store names, even though they have no sales, but not necessarily all the months in the LOOKUP_MONTH table. To do this, use the Report Data Options dialog box as described in the procedure below.



The four attribute join types available in the Report Data Options dialog box are the same as those in the VLDB Properties dialog box. The system reads them from the same location.

To view and change attribute join types

- 1 Open a grid report.
- 2 From the **Data** menu, select **Report Data Options**. The Report Data Options dialog box opens.
- 3 Under **Categories**, expand **Calculations**, and then select **Attribute Join Type**. The Calculations - Attribute Join Type appears on the right side of

the interface. This subcategory lists all attributes on the report, along with each attribute's join type, as shown in the image on [page 335](#).



If you have a long list of attributes, you can sort them by attribute name or by join type, by clicking the **Attribute** or **Join Type** column headers.

- 4 Depending on your goals for joining attribute data, select one of the following join types. Details and examples for each join type are provided in [Understanding attribute join types, page 336](#).
 - **Preserve common elements of lookup and final pass result table:** (Default setting) This join type causes the SQL Engine to calculate for only the attribute elements that exist in both the lookup and fact tables. This join type lets you display all attribute elements that exist in the lookup tables in your data warehouse, whether or not there is a corresponding fact value in the data warehouse. You cannot change the join types of individual attributes with this setting. The attributes and their join types are displayed for information only.
 - **Preserve lookup table elements joined to final pass result table based on fact table keys:** This join type causes the SQL Engine to calculate for only the elements that are joined to the fact table based on fact table keys.
 - **Preserve lookup table elements joined to final pass result table based on template attributes without filter:** This join type causes the SQL Engine to calculate for all attribute elements and ignores all related filtering conditions.
 - **Preserve lookup table elements joined to final pass result table based on template attributes with filter:** This join type causes the SQL Engine to calculate for all attribute elements and applies all related filtering conditions.
- 5 You can change the join type of a specific attribute, if you select any of the **Preserve lookup table elements** join types. In the **Join Type settings** table, click the **Join Type** for the attribute and select the join type from the drop-down list. The join type options are:
 - **Default:** This option sets the attribute to use the join type set for that individual attribute using the Attribute Editor. If a join type was not set at the attribute level, the attribute uses the join type set at the project level.
 - **Inner:** This option displays only the attribute elements common to all data warehouse tables from which data is being gathered for this attribute.

- **Outer:** This option displays all of the attribute elements from all data warehouse tables from which data is being gathered for this attribute.



If the report contains an object prompt on the report grid, the report's attributes are not yet defined and therefore cannot be displayed in the Join Type settings table. To assign join settings to the attributes included in the object prompt, see [Join Type settings for an object prompt, page 341](#).

- 6 When you are finished making changes, click **OK**. The report is re-executed against your data warehouse, and the newly calculated results are displayed.

Join Type settings for an object prompt

If a report contains an object prompt on the report grid, the report's attributes are not yet defined and therefore cannot be displayed in the Join Type settings table. You can assign join settings to the attributes included in the object prompt by answering the prompt in SQL View. This adds the attributes to the report, so that you can assign join type settings.

For example, the following report contains an object prompt, called Choose Attributes, on the rows:

	Metrics	Revenue
Choose Attributes		

When you access the Report Data Options dialog box to select join types for the attributes, no attributes are available for modification, since no attributes have been added to the report.

Running the report in SQL View and answering the prompt with all the available attributes adds them to the report. Now when you access the Report Data Options dialog box, all the attributes are displayed and available for modification. When you save the report as a prompted report, the user can select his own answers to the prompt and populate the report as he wants. The attribute join types that you assigned are saved and used.

To assign attribute join types to attributes in an object prompt

- 1 View the SQL of the report, by selecting **SQL View** from the **View** menu. The prompt is displayed, without executing the report.
 If the Report Data Options dialog box is still displayed, click **OK** to return to the report, and then change the report view.
- 2 Choose all of the attributes that are available in the prompt.
- 3 Click **Finish**.
- 4 From the **View** menu, select **Design View**. Notice that all the attributes are included in the report.
- 5 You can use the Report Data Options dialog box to assign join types for every attribute, as described in [To view and change attribute join types, page 339](#).
- 6 Save the report as prompted, rather than static.

When the report is re-executed, the prompt is displayed so that the user can select his own prompt answers. Each attribute that is selected uses the specified join setting.

Evaluation order of objects on reports

The order in which data is calculated affects the report results. By using evaluation order settings, you can control the order in which compound smart metrics, consolidations, derived metrics, derived elements, report limits, and subtotals are calculated and resolved for a given report.

Evaluation order is the order in which different kinds of calculations are performed by the Analytical Engine. The following simple example illustrates how evaluation order can influence the report results.

States Consolidation	Revenue	Cost	Revenue/Cost
New York	\$15	\$10	1.5
Virginia	\$20	\$15	1.33
New York + Virginia	\$35	\$25	X

In the above example, two calculations are involved, the States Consolidation and the compound smart metric Revenue/Cost.

The following conditions apply:

- If the States Consolidation is calculated first, X represents the Revenue/Cost value of the last row, and the result is $35/25$ or 1.4.
- If Revenue/Cost is calculated first, X represents the Revenue/Cost value of the last column, and the result is $1.5 + 1.33$ or 2.83.

By default, the compound smart metric is evaluated before the consolidation, yielding a result of 2.83. The next section, [Default evaluation order, page 343](#), provides more detail.

Default evaluation order

The default order of calculation is as follows:

- 1 Compound smart metrics
- 2 Consolidations, which are evaluated by their relative position on the report template:
 - Rows, from left to right
 - Columns, from top to bottom

 For examples of how evaluation order affects consolidations, see the following:

 - [Evaluation order, page 213](#), discusses multiple consolidations on a single report.
 - [Consolidation and view evaluation order example, page 349](#), demonstrates the interaction of a consolidation, subtotals, and the view definition with the evaluation order.
- 3 Report limits
- 4 Subtotals

Compound metrics that are not the direct aggregations of other metrics can be used in the evaluation order by setting the **Allow Smart Metrics** option of the Metric Editor to **Yes**.



Page-by and sorting are determined last, to arrange the positions of the calculation results. Their evaluation order cannot be changed.

Defining an evaluation order

You can specify the evaluation order by assigning a calculation a positive number to indicate the order in which it is to be calculated. When handling calculations, MicroStrategy first performs those to which default rules of order apply, then those that have been assigned a number. Use the Report Data Options dialog box to specify the evaluation order. The setting is found under **Calculations**, then **Evaluation Order**.

The default evaluation order described in [Default evaluation order, page 343](#) permits you to reorder consolidations only. Disabling this setting allows you to alter the evaluation order for the following objects:

- Compound smart metrics
- Consolidations
- Derived metrics
- Derived elements
- Metric limits
- Subtotals



A compound metric that has not been identified as smart cannot be part of a specified evaluation order; it is always calculated first, as discussed in [Default evaluation order, page 343](#).

Evaluation order in data definition and view definition

The data evaluation order settings control the order of evaluation for consolidations, report limits, compound smart metrics, and subtotals. Once the view definition becomes different from the data definition, the view

evaluation order is activated in the Report Data Options: Evaluation Order dialog box. It becomes available when:

- Objects are on the Report Objects but not on the grid
- A view filter is defined
- A derived metric is used



These actions must include or affect objects on which you can set the evaluation order. For example, a consolidation on the Report Objects but not on the grid will activate the view evaluation order. Since a derived metric is, by definition, a compound smart metric, it always activates the view evaluation order.

A separate view order is necessary because any manipulation of the view that does not change the SQL is performed after the base report is completed. Therefore, objects in the data evaluation order settings are evaluated first, and then those in the view evaluation order are evaluated. The following table describes where objects are evaluated.

Object	Evaluation Order
Consolidation	Data
Derived metric	View
Derived element	View
Report limit	Data
Compound smart metric	Data or View: • Set automatically to View • Can be changed manually
Subtotals	View

Data vs. view evaluation order example

Consider the following report, where Revenue Rank is a smart metric ranking the Revenue metric. (To create this metric, apply the rank function to the Revenue metric. On the Subtotals/Aggregation tab of the Metric

Editor, select the **Allow Smart Metric** check box. For more details, see the *MicroStrategy Developer help*.) Eight rows of data are returned.

Region	Metrics	Revenue	Revenue Rank
Central		\$5,029,366	6
Mid-Atlantic		\$4,452,615	5
Northeast		\$8,554,415	8
Northwest		\$1,761,187	1
South		\$5,389,280	7
Southeast		\$2,239,951	2
Southwest		\$3,694,132	3
Web		\$3,902,762	4

To restrict the report, create a metric limit of Revenue greater than \$3,000,000, as described below.

To create a metric limit

- 1 From the **Data** menu, choose **Report Data Options**. The Report Data Options dialog box opens.
- 2 Under **Categories**, expand **Calculations** and then select **Report Limit**. Calculations – Report Limit appears on the right-hand side of the dialog box.
- 3 Click **Modify**. The Report Limit Editor opens.
- 4 Double-click in the **Limit Definition** area of the editor. The Report Limit Qualification dialog box opens.
- 5 In the **Metric** box, type **Revenue** and click **OK**.
- 6 Select the **Revenue** metric from the list and click **OK**.
- 7 In the **Function** box, select **Metric Value**.
- 8 From the **Operator** box, select **Greater than**.
- 9 From the **Value** box, select **Value**.
- 10 Type **3000000** in the field.
- 11 Click **OK**. You are returned to the Report Limit Editor.

12 Click **Save and Close**. You are returned to the Report Data Options dialog box.

13 Click **OK**. You are returned to the Report Editor.

Only those regions with a Revenue greater than \$3,000,000 appear on the report. Only six rows of data are displayed, as shown below:

Region	Metrics	Revenue	Revenue Rank
Central		\$5,029,366	6
Mid-Atlantic		\$4,452,615	5
Northeast		\$8,554,415	8
South		\$5,389,280	7
Southwest		\$3,694,132	3
Web		\$3,902,762	4

The values for Revenue and Revenue Rank have not changed, only the amount of data that is displayed has. By default, Revenue Rank, as a smart metric, is calculated before the metric limit. The metric limit then removes the regions with the lowest revenue, which have the Revenue Rank of 1 and 2.

Next, create a derived metric, which causes the report view to differ from the data definition.

To create a derived metric

- 1** Select **New Metric** from the **Insert** menu. The Input Metric Formula dialog box opens.
- 2** Double-click **Revenue** in the list of Report Objects. It appears in the metric definition on the right.
- 3** Type **/2**, which also appears in the metric definition.
- 4** By default, the name of the derived metric is New Metric. Change it by typing **Revenue/2** over New Metric in the text field above the toolbar.
- 5** Click **OK** to return to the report.
- 6** Format the derived metric:
 - a** From the **Format** menu, point to **Revenue/2**, then select **Values**. The Format Cells dialog box opens.

- b On the Number tab, select **Currency** in the **Category** list.
- c Set **Decimal Places** to **0** (zero).
- d Click **OK** to return to the report.

The report results are displayed below. Notice that Revenue Rank is no longer missing 1 and 2, but is instead calculated based only on the six rows displayed. This results in a ranking from 1-6, instead of 3-8.

	Metrics	Revenue	Revenue/2	Revenue Rank
Region				
Central		\$5,029,366	\$2,514,683	4
Mid-Atlantic		\$4,452,615	\$2,226,308	3
Northeast		\$8,554,415	\$4,277,207	6
South		\$5,389,280	\$2,694,640	5
Southwest		\$3,694,132	\$1,847,066	1
Web		\$3,902,762	\$1,951,381	2

Once a derived metric is added to a report, the view evaluation order is activated. As a smart metric, the Revenue Rank metric is automatically moved to the view evaluation order, while the metric limit stays in the data evaluation order. Recall that, by default, objects in the view are calculated after objects in the data definition are computed.

You can specify whether compound smart metrics are evaluated in the data or the view definition. For instance, in this example, you can force Revenue Rank to be calculated before the metric limit, to include all the regions, even those no longer displayed on the report, in the ranking.

To move a compound smart metric from the view to the data evaluation order

- 1 From the **Data** menu, choose **Report Data Options**. The Report Data Options dialog box opens.
- 2 Under Categories, expand **Calculations** and then select **Evaluation Order**. Calculations – Evaluation Order appears on the right side of the dialog box.
- 3 Clear the **Use default evaluation order** check box.

Notice that both the Revenue Rank metric and the derived metric Revenue/2 are calculated in the view evaluation order (the metrics are displayed in the View evaluation order list).

- 4 Click **Advanced**. The Advanced dialog box opens.
- 5 Click in the **Metric Calculation** column of the **Revenue Rank** row, and select **Evaluate in the dataset** from the drop-down menu.
- 6 Click **OK** to return to the Evaluation Order dialog box.
Revenue Rank is now displayed in the Dataset evaluation order list.
- 7 Click **OK** to return to the report.

As shown below, the range of Revenue Rank is now 3-8, because the metric includes all the regions, even if they are not displayed on the report.

	Metrics	Revenue	Revenue/2	Revenue Rank
Region				
Central		\$5,029,366	\$2,514,683	6
Mid-Atlantic		\$4,452,615	\$2,226,308	5
Northeast		\$8,554,415	\$4,277,207	8
South		\$5,389,280	\$2,694,640	7
Southwest		\$3,694,132	\$1,847,066	3
Web		\$3,902,762	\$1,951,381	4

Consolidation and view evaluation order example



The following example demonstrates how the default evaluation order and view evaluation order affect consolidations. For another example on consolidations, see [Evaluation order, page 213](#). That example presents a report containing two consolidations but does not discuss the view evaluation order.

You want to compare the revenues for the Years 2002 and 2003 by Category and Subcategory. First, create a consolidation called Years to calculate the difference. The consolidation elements are defined below.

- 2002: Year = 2002
- 2003: Year = 2003
- Percentage Difference: (2002 - 2003) / 2003

Build a report with Category and Subcategory as the rows. Place the Years consolidation and the Revenue metric on the columns, then enable subtotals. When you execute the report, the following is displayed.

		Years	2002	2003	Percentage
Category	Subcategory	Metrics	Revenue	Revenue	Difference
Books	Art & Architecture		\$ 7,849	\$ 6,136	27.92%
	Business		\$ 7,610	\$ 7,487	1.64%
	Literature		\$ 7,103	\$ 6,602	7.59%
	Books - Miscellaneous		\$ 1,181	\$ 1,000	18.10%
	Science & Technology		\$ 6,093	\$ 5,955	2.32%
	Sports & Health		\$ 3,393	\$ 3,812	(10.99%)
	Total		\$ 33,229	\$ 30,992	46.57%
Electronics	Audio Equipment		\$ 852,990	\$ 843,270	1.15%
	Cameras		\$ 2,237,510	\$ 1,858,920	20.37%
	Computers		\$ 253,213	\$ 219,647	15.28%
	Electronics - Miscellaneous		\$ 486,748	\$ 469,214	3.74%
	TV's		\$ 707,093	\$ 612,099	15.52%
	Video Equipment		\$ 1,415,430	\$ 1,345,520	5.20%
	Total		\$ 5,952,984	\$ 5,348,670	61.25%
Total			\$ 5,986,213	\$ 5,379,662	107.83%



Due to space constraints, this report sample contains a report filter to include the Books and Electronics categories only.

Notice that the Percentage Difference is calculated correctly for each Subcategory, as shown in the following example:

$$(\$7,849 - \$6,136) = \$1,713 / \$6,136 = 27.92\%$$

The totals for the Revenue column are also correct. However, the totals for the Percentage Difference column are wrong:

$$\$33,229 - \$30,992 = \$2,237 / \$30,992 = 7.22\%$$

The report calculates the Percentage Difference as 46.57%. The default evaluation order calculates the consolidation before the total. The total is determined by adding all the values in the column, and not by the correct method applicable here of calculating the formula $[(2002 - 2003)/2003]$ across the rows. To calculate the total across the rows, change the evaluation order to calculate the total before the consolidation.

For steps, see [To move a compound smart metric from the view to the data evaluation order, page 348](#). Set Total to 1 and Years to 2.

The report is redisplayed as shown below:

Category	Subcategory	Years Metrics	2002 Revenue	2003 Revenue	Percentage Difference Revenue
Books	Art & Architecture		\$ 7,849	\$ 6,136	27.92%
	Business		\$ 7,610	\$ 7,487	1.64%
	Literature		\$ 7,103	\$ 6,602	7.59%
	Books - Miscellaneous		\$ 1,181	\$ 1,000	18.10%
	Science & Technology		\$ 6,093	\$ 5,955	2.32%
	Sports & Health		\$ 3,393	\$ 3,812	(10.99%)
	Total		\$ 33,229	\$ 30,992	7.22%
Electronics	Audio Equipment		\$ 852,990	\$ 843,270	1.15%
	Cameras		\$ 2,237,510	\$ 1,858,920	20.37%
	Computers		\$ 253,213	\$ 219,647	15.28%
	Electronics - Miscellaneous		\$ 486,748	\$ 469,214	3.74%
	TV's		\$ 707,093	\$ 612,099	15.52%
	Video Equipment		\$ 1,415,430	\$ 1,345,520	5.20%
	Total		\$ 5,952,984	\$ 5,348,670	11.30%
Total			\$ 5,986,213	\$ 5,379,662	11.27%

While this solution works for this particular instance of the report, what happens when you move Subcategory to the Report Objects? You cannot do it because you receive an error that the manipulation is incompatible with the evaluation order settings.

Moving Subcategory to the Report Objects activates the view evaluation order. When this occurs, the Years consolidation stays in the data evaluation order and the subtotals move to the view evaluation order. Since the data definition is evaluated before the view, the consolidation is calculated before the subtotals. However, you set the subtotals to be calculated first, to produce the correct results for the report. This conflict causes error.

To produce the correct results, you must either delete Subcategory from the report completely and not move it to the Report Objects, or create a second report without Subcategory.

Adding features for users

You can provide several report manipulation capabilities for report analysts, by either enabling the features or creating and adding capabilities to an existing report. These options provide an interactive reporting experience to your users. Once your report is created, execute and look at your new report to be sure it is visually pleasing, and that users can achieve the analysis workflows that are meaningful to them.

Analysts can apply a number of formatting characteristics to a report that they have executed. However, the report's designer can apply a broader range of formatting options to a report so that users can more effectively understand the data displayed on the report, or to simply achieve a more pleasing and easy-to-read look.

You can format the “No data returned” message that appears in a report or document when no data is returned from your data warehouse. For steps, see the *Report Services Document Creation Guide*.

For steps to add features for users, see the *Basic Reporting Guide*.

Distributing reports to a user community

MicroStrategy makes distributing reports to a wide range of users easier. You can use object templates to design high-level reports that can be customized for the end-user, or include features like prompts, drill maps, and derived metric creation to empower users to find answers to their business questions independently. Designing flexible reports also reduces the need to create new reports for every business question.

This section includes the following information:

- Object templates and shortcuts to filters and templates allow you to re-use common components of an attribute, report, or other object. For guidelines on re-using and maintaining objects, and steps to create object templates and shortcuts, see [Re-using objects via shortcuts and object templates, page 352](#)
- After you have created your reports, you can make the reports accessible to users. For steps, see [Making reports accessible, page 361](#).
- You can control which reports are cached. For steps and background information on report caches, see [Caching, page 361](#).

Re-using objects via shortcuts and object templates

Re-using objects can make distributing reports to users much easier. For example, you may have 50 reports that are all based on the same filter. If each report uses its own embedded filter, then that change must be performed in each of the 50 reports. However, if the filter is built as a stand-alone object and added as a shortcut to the reports, the filter can be

changed once, and the change is automatically applied to all of the reports in which the filter is used. By creating the filter as a stand-alone object and including the filter as a shortcut in reports, you avoid having to manually change each of the reports that the filter is used in.

When planning to update objects, you can search for the objects that contain them or other objects they contain. This type of search can identify which reports, documents, and so on will be affected by changes to the object. You can, for example, search for all templates that contain the Profit metric by right-clicking the Profit metric and selecting **Search for Dependents**. The Search dialog box is displayed with a list of the Profit metric's dependents. As another example, you can search for all metrics that are used by the Store Sales template by right-clicking the Store Sales template and selecting **Search for Components**.

Objects that can be reused include filters, report templates, custom groups, and consolidations. For information on how to take advantage of re-usable objects and steps to create them, see the following links:

- [Shortcut to a template, page 353](#)
- [Shortcut to a filter, page 357](#)
- [Object templates, page 357](#)

Shortcut to a template

A template specifies the information to retrieve from the data warehouse and how it is displayed in the report grid. Information on the report template includes metric default values, banding options, join type setting, and data sorting options. A template does not include filter information. You can create a stand-alone template in the Template Editor.

When you add a new template to a report, you can replace the default report template with a shortcut to the new template. Creating a shortcut to a template allows you to use a stand-alone template with any number of reports. Any changes made to the stand-alone template are then propagated to all reports that use that template.

You can determine whether a report's template is a shortcut to a template by looking at the template's name in the Grid definition. If a shortcut icon appears in the title bar, the report's template is a stand-alone template.

For steps to add a stand-alone template to a report, see [Adding a stand-alone template to a report, page 354](#).

When you modify a shortcut template in a report, you can either choose to save those changes to this report only, or save them to all reports that share the same template.

If you save the changes to this report only, changes made to the template in this report do not impact other reports. If you save the changes to all reports that use this template, changes made to the template data definition in this report are propagated to all other reports that contain a shortcut to the same template. For example, if you add a metric to a report with a shortcut template, and then save the changes to the shortcut instead of only to the local copy of the report, all other reports that contain the same shortcut template display the new metric in Report Objects. For steps to save the report, see [Saving a report with a shortcut to a stand-alone template, page 355](#).

To avoid unexpected impacts on reports when making template changes, conduct a quick impact analysis: see [Modifying a template and validating changes, page 356](#).

Adding a stand-alone template to a report

The following procedure describes how to apply a previously created, stand-alone template to a report during report creation.

Prerequisite

- This procedure assumes that you have already created a stand-alone template. For steps, see the *MicroStrategy Developer help*.

To add a template to a report

- 1 In Web or Developer, open the report in Design Mode.
- 2 Use the Object Browser to search for and locate the template that you want to add to the report.
 - If the Object Browser is not displayed, open it by selecting **Object Browser** from the **View** menu.

- 3 Right-click the template and select one of the following (the options are described in more detail above):
 - To connect a stand-alone template to this report, select **Replace with shortcut to template**. Going forward, changes to the stand-alone template are propagated to the report, and vice versa.
 - To embed the template in the report, select **Replace with copy of template**. Going forward, changes to the stand-alone template are not propagated to the report, and vice versa.

The template's components appear in the report's Report View pane. If the template is a shortcut, the template's name is also displayed. If the report does not use a shortcut template, the template is named Local Template.

- 4 Save the report. If you added the template as a shortcut, you must save the report with the shortcut to retain that functionality. See [To save a report with a shortcut to a stand-alone template, page 355](#) for steps.

Saving a report with a shortcut to a stand-alone template

When you save changes to a report that contains a stand-alone template, you can choose to either save the changes to that report only (called a local copy), or save the changes to all reports that use the same template (called a template shortcut). Steps are below to save a report's change to a template shortcut so that those changes are automatically propagated to all reports that also use the stand-alone template they are all connected to.

Reports with shortcuts to stand-alone templates and filters allow you to share report caches. For an introduction to report caches, see the *Caching* chapter in the *System Administration Guide*.

To save a report with a shortcut to a stand-alone template

- 1 Save the report with a shortcut to a stand-alone template:
 - a Click **Save and Close** on the toolbar of the Report Editor or Report Viewer. The Save Report As dialog box opens.
 - b Select the folder in which to save the report, then type the name of the new report.
 - c Click **Save**. The Advanced Save Options dialog box opens.

- If the report contains prompts, the Save Options dialog box opens after you have named the report. Click **Advanced** on the Save Options dialog box. For a prompted report, see the *MicroStrategy Developer help*.
- 2 Select one of the following options for saving the report:
 - **Create a local copy of the template.** Changes that you make to the template are not propagated to the original object, and vice versa. This is also called a local or embedded template.
 - **Retain the shortcuts to the template.** Creating shortcuts allows you to take advantage of the benefits of object reuse. Changes that you make to the template are propagated to the original object, and vice versa. Use this option to share report caches.
 - 3 To use the selections that you made above as the defaults for saving reports in future, select the **Remember answers the next time I am prompted** check box.
 - 4 Click **OK** to save the report.

Modifying a template and validating changes

Changing a template may cause unexpected issues with some of the reports that depend on the template. For example, a report has a shortcut to a template, which contains Country, Region, Metric 1, and Metric 2. The view filter is set to “Metric 1 > 20.” Metric 1 is then removed from the stand-alone template but not from the report view. When that report is executed, an error occurs because the view filter can no longer be evaluated (Metric 1 no longer exists).

You can check for issues like the example above by validating your template changes. You validate template changes by conducting a quick impact analysis with the Template Dependency Validator, before saving any changes to a template. When a template is modified and saved in MicroStrategy Developer, the Template Dependency Validator lists:

- Reports that depend on the template
- Reports that will not run if the change is completed

To resolve any issues, you can do the following in the Template Dependency Validator:

- Cancel the change and re-evaluate.

- Open each dependent report and remove the dependencies, then make your changes to the template.

For the previous example, you might remove the view filter from the view definition of the dependent report.

Shortcut to a filter

When you add a filter to a report, you have the option of adding it as an embedded filter which applies to that report only, or you can add a shortcut to an existing filter.

An embedded filter is generated when a copy of an existing filter is added to a report or when the filter is created on the fly from within the report. Changes made to an embedded filter affect only the report in which it is contained because the filter exists only within that report.

In contrast, a shortcut to a filter stands alone as a filter and can be used in multiple reports. When the stand-alone filter is changed, the changes are propagated to all other reports that contain a shortcut to the same filter. For more information and steps on how to create an embedded filter or a shortcut to a filter, see [Shortcut vs. embedded filters, page 177](#).

Object templates

An object template is a predefined structure you can use to create a new object. An object template provides a foundation you can use to create reports, metrics, filters, or other objects. For example, you may want to create many filters that contain the current month as part of the filter's definition. You can create a filter object template that contains the current month and use the filter object template to automatically include the current month condition in every new filter you create. Using an object template, you only have to define that filtering condition once.

Another example is a need to build multiple reports containing the attribute Day and the metrics Revenue, Cost, and Profit. To reduce the time spent creating these similar reports, define a report with these objects and save it in the Object Templates folder, thus creating a report object template. Multiple reports can now be based on the new report object template, thus incorporating the template's attributes and metrics quickly and easily.

You can create empty object templates that define default formatting on a project level for new reports, templates, metrics, and documents. For a list of

settings you can define for each type of empty object template, see [Empty object templates, page 358](#).

You can create object templates for the following objects:

- Consolidations
- Custom groups
- Documents
- Filters
- Metrics
- Reports
- Templates (A template is the part of a report that specifies the information to retrieve from the data warehouse and how it is displayed in the report grid. It does not include filter information. For more information on what a template includes, see [Shortcut to a template, page 353](#).)

You can determine whether users can choose from a list of available object templates when creating a new object, or if they must use a default object template that you define. For steps, see [Making object templates available to users, page 360](#).



If an object template is altered, the change is not propagated to previously defined objects.

To be used as an object template, the object must be saved in the Object Templates folder. This is the only difference between an object (such as a report) and an object template (such as a report object template).

Empty object templates

Empty object templates define default formatting and other default settings on a project level for new reports, templates, metrics, and documents. They do not contain a definition.

For example, your project has a series of reports that must be exported in Excel format to a particular location. A specific Excel macro must be run after the report is exported. You can create a user-defined empty report object template, called Excel Export Settings, with these specifications. When the Excel Export Settings report object template is used to create a new report, the new report contains the correct Excel information.

As another example, a project requires all currency values to include cents for precision and to distinguish negative values in red font. To meet these conditions, create an empty metric object template named Currency Formatting and set these formats. When a user creates a new metric that returns currency values, he selects Currency Formatting in the New Metric dialog box. The formatting for red negative values and cents is included in the new metric.

There are two kinds of empty object templates:

- **Empty object templates** provided by MicroStrategy. These are displayed in the New Object dialog box as Empty Document, Empty Metric, Empty Report, and Empty Template.

You can choose whether or not to display the empty object template for a particular object type. For steps, see [Making object templates available to users, page 360](#).

- **User-defined empty object templates.** These help you control new objects created in your project, by specifying a default set of formatting options that should be applied to each new object.

For example, you can create an empty metric template with specific currency formatting or an empty report object template set to display data by default in outline mode. The user-defined empty object template contains only formatting options. User-defined empty metric templates do not have formulas and user-defined empty report templates do not include attributes, metrics, or other grid items in the report objects.

Empty object templates are saved in the Object Templates folder.

The settings available in a user-defined empty object template vary with the type of object, as described below:

- An **empty metric object template** does not contain a formula but can contain the following settings:
 - Formatting settings
 - VLDB properties
 - Formula join type
 - Metric join type
 - Metric column settings
- An **empty report object template** does not contain any grid units, such as attributes, metrics, consolidations, and so on. An empty report contains:

- Export options
 - Filters
 - Formatting
 - Grid options
 - Report data options
 - VLDB properties
 - Other settings such as merging header cells and automatic column width
- An **empty template object template** is similar to an empty report object template.

Making object templates available to users

You can determine whether users can choose from a list of available object templates when creating a new object. By default, the New Object dialog box displays when users create a new object and contains a list of available object templates users can use to define the new object.

If you choose to hide the New Object dialog box for a type of object, each new object of that type will be created using the default object template. If you have not defined a default object template, the object will be created using MicroStrategy's empty object template. Default object templates include Documents, Metrics, Reports, and Templates.

For steps to define these settings, see the *MicroStrategy Developer help*.

If you choose to display the New Object dialog box, you can choose to hide MicroStrategy's empty object template. The empty object template is displayed in the New Object dialog box as Empty Document, Empty Metric, Empty Report, or Empty Template. If you hide the empty object template, users cannot create new objects with the empty object template.

If you disable the empty object template for a particular object type, and the only object template is the project-level default object template, all new objects must use the formatting and other settings defined on the default object template. You can use this feature to ensure that all reports in a project adhere to specific formatting standards.

For example, you create a report object template with specific formatting and save the report object template as Default Format. The project does not have

any other report object templates. You set Default Format as the default object template for reports and do not show the Empty Report object template. If a user has enabled object templates for reports, when he creates a new report, his only choice in the New Grid dialog box is Default Format. If object templates for reports have been disabled, the New Grid dialog box is not displayed and Default Format is automatically used.

For steps to enable and disable object templates, see the *MicroStrategy Developer help*.

Making reports accessible

To distribute reports, you place the reports in a specific folder so that other users can access them. Reports saved in the Reports folder under the Public Objects folder can be viewed by other users on the system. A Web user can navigate to Shared Reports and run those reports by clicking them. MicroStrategy Developer users can navigate to reports in the Public Objects\Reports folder and execute them by double-clicking them.

To save reports that are frequently used to create new reports, place the reports in the Reports folder under Object Templates. They are displayed as templates on which to base a new report, when a MicroStrategy Web user clicks **Create New Reports** or a MicroStrategy Developer user selects **New**, then **Reports**.

The reports and folders described above are not special in any way except that they are available for other users to access.

To view the Object Templates folder, select **Preferences** from the **Tools** menu. Expand **Developer**, and then select **Browsing**. Select **Display Hidden Objects**, and click **OK** until you are returned to MicroStrategy Developer. For more information on object templates, see [Object templates, page 357](#).

Caching

In general, a cache holds recently accessed values to increase performance. Frequently-requested reports are often cached, providing faster execution because the cached reports do not have to access the data warehouse.

Default settings for the report caches for all reports in a project are commonly configured by your system administrator. You can override the default settings for specific reports. For example, you may want to enable caching for a specific report, even if other reports in your system are not cached, because it puts a large load on the data warehouse. Or, you may want to disable caching for a specific report, even if other reports in your system are cached, because the data used in the report changes so often that the cached version of the report is rarely valid.

You can also select a specific schedule for invalidating the cache. Invalidating a report cache indicates to Intelligence Server that this cache should not be used. For example, you can schedule the cache to be invalidated every time new data is loaded into the data warehouse. In this case, the first time the report is executed after the data load, a new cache is created, with the most up-to-date data.

For reports, the following caches are used:

- The **report cache** contains pre-processed report data and is stored on disk. Report caches are created automatically.
- **Intelligent Cubes** are similar to report caches in that they provide a set of data that reports can quickly retrieve without having to re-query a data source. Unlike report caches, Intelligent Cubes are created manually. Intelligent Cubes also allow reports to utilize various OLAP Services features. For an introduction to Intelligent Cubes, see [Intelligent Cubes, page 288](#).

For detailed information about all types of caches, see the *System Administration Guide*.

To enable or disable caching for a specific report

- 1 In MicroStrategy Developer, open the report in the Report Editor.
- 2 From the **Data** menu, select **Report caching options**. The Report Caching Options dialog box opens.
- 3 Select whether to enable or disable caching of this report:
 - To enable caching for this report even if caching is disabled at the project level, select the option called **Enabled: This report should be cached**.

- To disable caching for this report even if caching is enabled at the project level, select the option called **Disabled: This report should not be cached**.
 - To use the default project-level setting for caching, select the **Use default project-level behavior** option. This indicates that the caching settings configured at the project level in the Project Configuration Editor apply to this specific report or document as well.
- 4 Click **OK**. The Report Caching Options dialog box closes and your changes are saved.

Exporting large reports as delimited text files

The bulk export feature allows you to select a report and export it to a delimited text file. Using this feature, you can retrieve data from large result sets without having to load the result sets in memory. As the data is brought back incrementally, no limit is set on the size of the result set to be retrieved.

While you can schedule a bulk export report for execution with Distribution Services, you cannot directly execute a bulk export report in MicroStrategy Developer. When you right-click a bulk export report, the **Run Report** option is not available. You can, however, view the report in Design View and SQL View on MicroStrategy Developer. (All the other views are disabled.)



You can convert a bulk export report back to a standard report, to be able to execute the report in MicroStrategy Developer. For steps, see [To convert a bulk export report to a standard report, page 365](#).

Prerequisites

- To use the Bulk Export feature, you must have a Distribution Services license.
- To execute a bulk export report from MicroStrategy Web, you must have the Web Subscribe To Bulk Export privilege.
- To define a report as a bulk export report in MicroStrategy Developer, you must have the Use Bulk Export Editor privilege.

To configure a report for bulk export

- 1 In MicroStrategy Developer, open a report in the Report Editor.

- 2 From the **Data** menu, select **Configure Bulk Export**. The Bulk Export Setup dialog box is displayed.
- 3 Click the **General** tab.
- 4 By default, the **Bulk export database instance**, which is the database instance for the database containing the report data, is defined as the data warehouse for the project. You can specify another database instance in this field.
- 5 Select the **Default delimiter for the exported text file**, which is the character that separates the data in the text file. If the character that you want is not available, select **Other**, then type the character.
- 6 Click the **Advanced** tab.
- 7 Type the **Maximum time for SQL execution to build table**, in seconds. If this time limit is reached before the table is built, the export is cancelled with a timeout error.
- 8 Type the **Maximum time for exporting data to file**, in seconds. If this time limit is reached before the file is completely written, the export is cancelled with a timeout error.
- 9 Type the **Number of rows retrieved in each step**, which is the number of database rows that the bulk export retrieves at a time.
- 10 Type the **Table space**, which is the database table space in which the temporary table for the bulk export is created.
- 11 Type the **Table prefix**, which is the table prefix for the bulk export temporary table.
- 12 Click the **SQL Statements** tab.
- 13 Specify the SQL statements to be executed after the table is populated, but before the data is exported to a file. For instance, if you need to format the warehouse data in a particular way, you can run a SQL script to do so.
- 14 Click **OK**. The Bulk Export Setup dialog box closes and the report is now specified as a bulk export report.
- 15 Save the report.

After you save a report as a bulk export report, its icon in MicroStrategy Developer changes as shown in the following figure.



Note the following:

- If you open a bulk export report in SQL View, certain column names are replaced by column aliases. You can view the column names by changing the default VLDB properties for metrics. To do this, from the Project Configuration Editor, select **Database instances** in the Project Definition category, then click **VLDB Properties**, and then select the **Default To Metric Name** option from the Metrics folder. Select the **Use the metric name as the default metric column alias** option, and save the changes. Restart the MicroStrategy Intelligence Server. For detailed information on VLDB Properties, see the *Supplemental Reference for System Administration*.
- If you have selected multiple reports, including bulk export reports, and choose the **Run Report** option from the right-click menu, all the bulk export reports are filtered out and are opened in the Report Editor; the other reports are executed as usual.
- Bulk export reports have the same limitations and restrictions as data mart reports. In particular, bulk export reports cannot be used in HTML documents or Report Services documents. For information on HTML documents, see the *MicroStrategy Developer help*. For information on Report Services documents, see the *Report Services Document Creation Guide*.
- Large bulk export reports may require that a substantial amount of data be transferred between Intelligence Server and the data source. You can improve the performance of this process by transferring the data in multiple, incremental transfers of data. To do this, enable the Incremental Data Transfer VLDB property, in the Query Optimizations VLDB properties group. For steps to set VLDB properties, see the *VLDB Properties* chapter of the *Supplemental Reference for System Administration*. This property is available at both the project and report levels.

To convert a bulk export report to a standard report

- 1 In MicroStrategy Developer, open the bulk export report in the Report Editor.

- 2 From the **Data** menu, select **Remove Bulk Export Definition**.
- 3 Save the report. It is no longer a bulk export report, and its MicroStrategy Developer icon changes to the standard report icon.

GRAPHING

Introduction

This chapter discusses how to view, format, and work with MicroStrategy graph reports. It focuses primarily on how you can use MicroStrategy graphs to better analyze and format your enterprise business data.

Previous chapters of this guide and the *Basic Reporting Guide* provided information on how to perform a variety of tasks with MicroStrategy reports. This chapter assumes that you have a fundamental understanding of how to design MicroStrategy reports and add objects such as attributes, metrics, and filters to them.

Specifically, this chapter includes the following:

- [*Graphing best practices and troubleshooting, page 368*](#)
- [*Viewing graph reports, page 371*](#)
- [*Adding an object to a graph report, page 372*](#)
- [*Choosing an appropriate graph style, page 374*](#)
- [*Moving objects on a graph report, page 378*](#)

- [Graph style usage and examples, page 381](#)
- [Advanced graph style usage and examples, page 408](#)
- [Formatting a graph report, page 434](#)
- [Choosing a graph image format for HTML documents and Web, page 503](#)
- [Exporting and sharing graphs, page 504](#)
- [Using interactive graph visualizations in documents: Widgets, page 507](#)



Note the following:

- The extent to which you can format a graph, add and manipulate data, and so on depends on the permissions granted to you by your administrator. This chapter assumes you have Developer privileges, and can therefore format various aspects of a report grid and graph.
- It is recommended you review the list of graphing best practices in [Graphing best practices and troubleshooting, page 368](#) before designing a graph in MicroStrategy.

Graphing best practices and troubleshooting

Before proceeding to other sections of this chapter, review the following list of best practices for designing an effective graph in MicroStrategy, as well as information on how to troubleshoot specific graphing issues.

The information below applies to every graph style available in MicroStrategy:

- You can select **Grid Graph View** to see the report grid and corresponding graph side-by-side. This allows you to add and replace objects on the grid and immediately see how its corresponding graph changes.



In MicroStrategy Web, Grid Graph View is referred to as Grid and Graph mode.

- As an alternative to using Grid Graph View to move objects on a graph, you can also use the categories and series drop zones. This allows you to move objects between the categories and series for the graph and immediately see how these changes affect the graph. To show or hide the

series and categories drop zones in MicroStrategy Developer, click the **Show Drop Zones** option  in the toolbar.

While you can use the categories and series drop zones in both MicroStrategy Developer and Web, the effects of the changes you make are only shown immediately in MicroStrategy Developer. In Web, you must execute the graph report to see the changes.

- Your report grid must contain at least one attribute and one metric to be displayed as a graph.
- As you design your graph, try to position objects such as attributes and metrics on the report grid (or the categories and series) in a variety of ways until the graph presents the data in the desired manner. This is often a necessary step in designing an effective graph because it helps you determine the way you wish to display the graph data. For example, view a report in Grid Graph mode, and try placing an attribute or two on the grid's columns and the metrics in the rows. Then, swap the rows and columns to see how the graph report changes.



For more information on how the positioning of report objects affects how a graph is generated, see [Moving objects on a graph report, page 378](#).

- Each graph style requires a minimum or specific number of attributes and metrics on the report grid. For example, a Pie chart requires only one attribute and one metric on the report grid, although more attributes and metrics can be included. On the other hand, a Hi-Lo-Open-Close Stock graph requires a specific number of report objects: one attribute and four metrics.



For more information on the minimum requirements for each graph style, see [Choosing an appropriate graph style, page 374](#).

- Consider the order of the attributes, metrics, and other objects on your report grid. For example, if you have three metrics in the columns of the grid, a dual-axis graph may display the first metric using the Y-1 axis and the second and third metrics as the Y-2 axis.



For an example of this, see [Vertical Bar: Dual-Axis Clustered graph example, page 385](#).

- Dual-axis graphs can improve the display of graph reports that include metrics that return different types of data. For example, a report includes a Profit metric that returns currency information in the thousands and millions. The same report also includes a Profit Margin metric that returns percentages such as 25%. These two metrics would benefit from being on different axes that use different ranges to display values. For information on using dual-axis graphs, see [Scaling a graph, page 489](#)

and [Defining a graph to use a dual Y-axis in MicroStrategy Developer, page 494](#).

- The way in which report data is sorted on your report grid directly affects how the data is displayed on the graph report. For example, the bars within a Bar graph can be displayed in descending order instead of ascending order, or they can be sorted based on a different metric in the report. As a test, view a report in Grid Graph mode, sort a metric on the grid, and notice how the graph changes accordingly.
- If you are creating a graph report to view in MicroStrategy Web, it is recommended that you choose the JPEG or PNG image format before you save the graph report. If your graph report includes color gradients, you must use one of these formats to display the graph correctly in MicroStrategy Web. For more information on configuring image preferences for graph reports in MicroStrategy Developer for use in MicroStrategy Web, see [Choosing a graph image format for HTML documents and Web, page 503](#).
- If the graph is included in a Report Services document, you can provide some interactive functionality to the graph by doing the following. In the document, add a Slider selector under the graph. Then, configure the slider to control the graph. This allows you and other users to navigate to different sections of the graph's results. For information about creating selectors, see the *Report Services Document Creation Guide*.
- Use graph formatting effects such as bevels (three-dimensional surface patterns), gradients (combinations of two colors), curved lines, and transparency to make your graph series and other objects stand out and look sharper.
 - To apply gradients, bevel effects, and transparency effects, right-click your graph's series and access the Graph Preferences dialog box. For these steps, see [Manually formatting graph series and objects, page 451](#).
 - To apply a curved effect to a Line or Area graph, from the **Graph** menu, select **Preferences**. Expand **Options**, and select the **General** category. Select the **Enable curved lines** or **Enable curved areas** check box, respectively.
- Apply a glossy, rounded effect to your graph's series. The effect is displayed in Graph View and Grid Graph View in MicroStrategy Developer and Editable, Interactive, and View Mode in MicroStrategy Web. To do so, from the **Graph** menu, select **Preferences**. Expand

Options, and select the **General** category. Select the **Apply rounded effects to all series** check box.



In Flash Mode in MicroStrategy Web only, a more distinct rounded and glossy effect is applied to the graph series when you select this check box.

- Use font anti-aliasing to display more clearly and distinctly the text that appears on or near your graph report. To do so, from the **Graph** menu, select **Preferences**. Expand **Options**, and select the **General** category. Select the **Use font anti-aliasing** check box. This formatting effect is especially noticeable when the graph report is viewed in Flash Mode in MicroStrategy Web.

Troubleshooting issues with graphs

The following table lists solutions to common issues you may have while working with graph reports.

Graph Issue	Solution
All of your graph data does not appear on one page and you want it to	Change the number of categories and series. From the Graph menu, select Preferences . Adjust the Maximum number of categories and Maximum number of series settings as desired. For example, if you are reporting on 15 different departments and the Department attribute is in the rows of your grid, set the Maximum number of categories to 15 to ensure all of the department data is shown on one page.
Cannot view graph data for one attribute element at a time	Move the corresponding attribute from the report grid to the Page-by field. This allows you to produce a different, more refined graph each time you select a new element in the Page-by field. However, keep in mind that the placement of report objects on the grid determines whether a certain graph style can be generated.
Cannot move objects such as titles or the graph legend on your graph	Enable manual layout mode. For more information about how to enable manual layout mode, see Pros and cons of manual and automatic graph layout, page 484 .

Viewing graph reports

You can view any MicroStrategy report as a graph by selecting **Graph View** from the **View** menu in MicroStrategy Developer or clicking **Graph** in the MicroStrategy Web toolbar while viewing a report.

You can select **Grid Graph View** to see the report grid and corresponding graph side-by-side. This allows you to add and replace objects on the grid and immediately see how its corresponding graph changes.



Reports saved while being viewed in Graph View are displayed in Graph View when re-opened. Likewise, reports saved while viewed in Grid Graph View are displayed in Grid Graph View.

Adding an object to a graph report

As you work with a MicroStrategy graph report, you may need to analyze data that is not currently on the report. In this case, you must add objects to the grid report so that the graph report is changed accordingly.

You can add data to a report by using the Object Browser to add attributes, metrics, consolidations, and other objects.

For example, you have a grid report that provides data about the yearly cost of medical supplies. Your district manager wants to view this data graphically. Moreover, she is interested in seeing this data at the month level, as well as the year level already available on the report.

Since your manager wants to view the report at the month level, you might choose to add the Month attribute to your cost of medical supplies report. The procedure below guides you through adding an object to a graph report.



Some graph styles accommodate more report data than others, as explained in [Graph style usage and examples, page 381](#) and [Advanced graph style usage and examples, page 408](#). Therefore, in some cases, adding report objects to a report may limit the graph styles you can use to present data effectively. Look at the table in [Choosing an appropriate graph style, page 374](#) to select a graph style that can accommodate the number of objects on your report. Experiment with different graph styles until you find one that works best for your specific data.

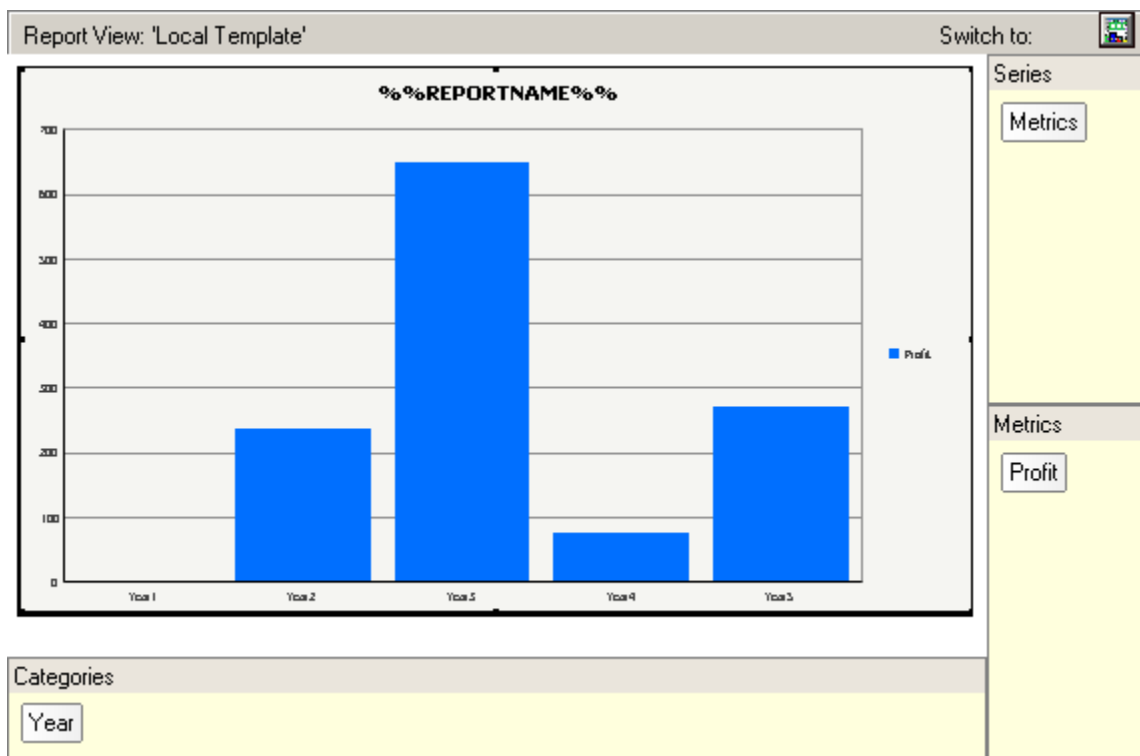
To add an object to a graph report

- 1 If the report is not already in Design view, from the **View** menu, select **Design View**.

- 2 From the **Object Browser**, navigate to the attributes and metrics to add to the graph report. Add at least one attribute and one metric to the **Report View** pane, as shown below.

Report View: 'Local Template'		Switch to:	
	Metrics	Profit	
Year			

- 3 In the **Report View** pane, click **Switch to** when the button displays the graph icon, which is shown in the image above. The graph report is displayed as a graph with random data to show how the graph report may appear once it is executed in Graph View.
- 4 On the Graph toolbar, click **Show Drop Zones** (). This displays areas to define the categories and series for your graph report, as shown below.



- 5 From the **Object Browser**, continue to add attributes, metrics, and other objects to the categories and series of the graph report.
- 6 Once finished, from the **View** menu, select **Graph View** to view your graph.

Choosing an appropriate graph style

You can display your report graph using a variety of different graph styles, depending on the structure and number of objects on your report and your personal preferences. The structure and number of objects on your report must factor into your decision because some graph styles cannot be displayed unless a certain number of attributes or metrics appear on the report grid.

Minimum object requirements for each graph style

The table below provides information about the minimum object requirements for displaying a graph in each graph style.



Although the table below lists the minimum requirements to display a graph in each graph style, some graphs require specific types of attributes and metrics to be useful. For this information as it relates to each graph style, see [Graph style usage and examples, page 381](#) and [Advanced graph style usage and examples, page 408](#).

Graph Style	Minimum Requirements (unless otherwise noted)
Area	One attribute, one metric
Bar	One attribute, one metric
Line	One attribute, one metric
Pie	One attribute, one metric
Scatter: XY	One attribute, two metrics Note: If there are more than two metrics, the graph may not be useful because Scatter plots are used to plot two values at a time. Any additional metrics are not included on the display of the Scatter plot.
Scatter: XYZ	One attribute, three metrics Note: If there are more than three metrics, the graph may not be useful because Scatter plots are used to plot three values at a time. Any additional metrics are not included on the display of the Scatter plot.
Polar	One attribute, two metrics
Radar	One attribute, one metric
Bubble	One attribute, three metrics Note: If there are more than three metrics, the graph may not be useful because Bubble charts are used to plot two values (a first and second metric) at a time. The size of each bubble represents the third metric. Any additional metrics are not included on the display of the Bubble chart.

Graph Style	Minimum Requirements (unless otherwise noted)
Stock:Hi-Lo	One attribute, two metrics Note: To ensure that this type of graph displays data correctly, you must include specific types of metrics. For more information, see Using Stock graphs effectively, page 412 .
Stock: Hi-Lo-Open	One attribute, three metrics Note: To ensure that this type of graph displays data correctly, you must include specific types of metrics. For more information, see Using Stock graphs effectively, page 412 .
Stock: Hi-Lo-Open-Close	One attribute, four metrics Note: To ensure that this type of graph displays data correctly, you must include specific types of metrics. For more information, see Using Stock graphs effectively, page 412 .
Histogram	One attribute, one metric
3D: Risers	One attribute, one metric
3D: Floating	One attribute, one metric
3D: Connect Group	One attribute, three metrics
3D: Connect Series	One attribute, one metric
3D Surface	One attribute, one metric
3D: Scatter	One attribute, three metrics
Gauge	One attribute, one metric
Funnel	One attribute, one metric
Pareto	One attribute, one metric
Pareto: Percent	One attribute, two metrics Note: To ensure that this type of graph displays data correctly, you must include specific types of metrics. For more information, see Pareto Percent example, page 426 .
Boxplot	One attribute, exactly five metrics Note: To ensure all five sections of the Boxplot (minimum, maximum, median, lower quartile, upper quartile) appear correctly on the graph, the five metrics must be defined in a specific way, as explained in Using Boxplot graphs effectively, page 430 .
Gantt	One attribute, two metrics Note: To ensure that this type of graph displays data correctly, you must include specific types of metrics. For more information, see Gantt chart example, page 431 .

Once you determine the graph style in which you want to display your graph, follow the steps below to apply the style to your graph report.

To apply a graph style to a graph report

- 1 View the report in Graph View. For steps to view a report as a graph, see [Viewing graph reports, page 371](#).
- 2 In MicroStrategy Web, on the **Graph** toolbar, from the first drop-down list, select a graph type.
- 3 From the next drop-down list, select a graph style. This drop-down list displays the different graph styles within each graph type. For example, if you select the Pie chart graph type, you can choose from several different Pie chart graph styles, including Pie:Ring or Pie:Multiple Proportional Ring.
- 4 Click **OK** to apply the graph style to your graph:
 - If your report contains sufficient data to be displayed in the selected graph style, your graph is generated in the new style.
 - If your report does not contain sufficient data to be displayed in the selected graph style, a dialog box notifies you of this and your graph is not updated. Refer to the table in [Choosing an appropriate graph style, page 374](#) to display your report and try using a different graph style.

Placement of report objects and choice of graph styles

Depending on the graph style you select, if you place attributes and metrics on your grid in certain ways, the graph cannot be generated. When this occurs, the system notifies you that the report contains insufficient data to be displayed in the selected graph style.

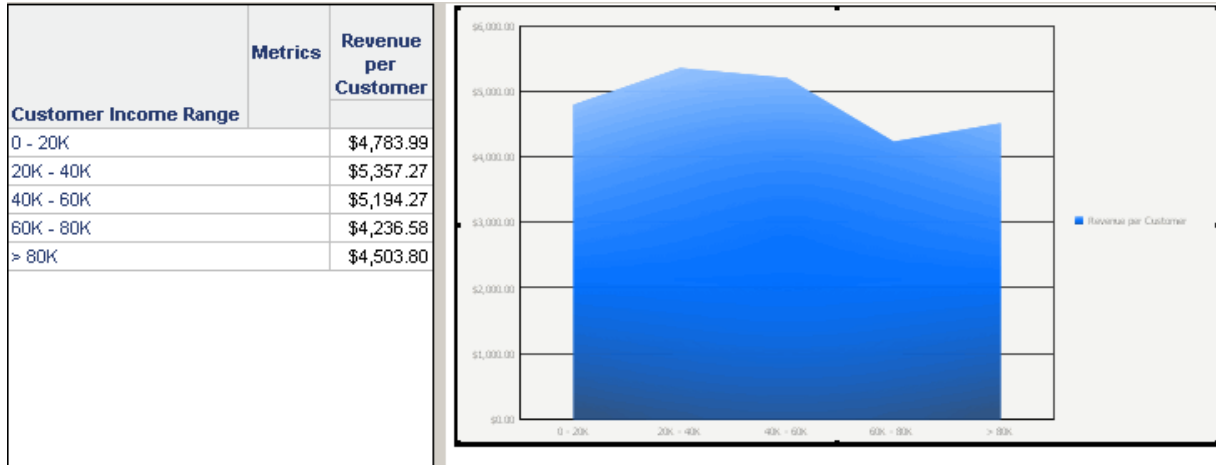
Check that you have chosen the right graph style for the number of attributes and metrics you have. For information about the minimum requirements of each graph style, see the table in [Minimum object requirements for each graph style, page 374](#). Once you have determined that the graph style is an appropriate choice, experiment with different ways to position the attributes and metrics on the grid until the graph is generated in the selected style.

Consider the following example. The grid report for the graph below contains the following:

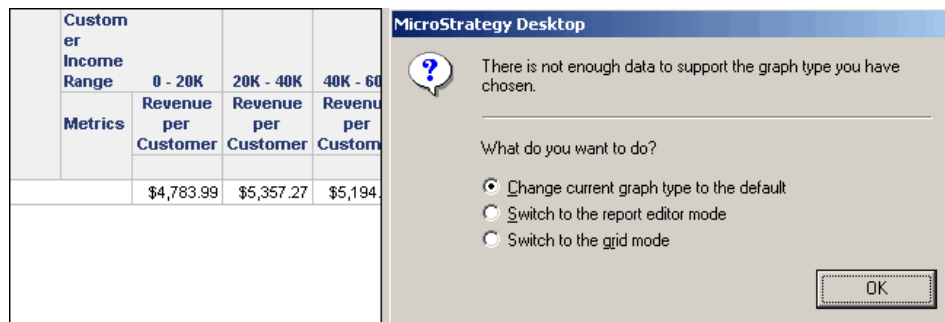
- An attribute (Customer Income Range) in the grid rows

- A metric (Revenue per Customer) in the grid columns

In the image below, the grid report is displayed as a graph using the Area graph style.



Next, move the attribute to the columns. When you view the resulting graph, you encounter an error, as shown in the image below.



This occurs because the placement of the attributes on the grid columns does not accommodate the display format of the Area graph style. The error also occurs if you move the attribute to the columns and the metric to the rows.

As you design your graph, keep in mind that the location of the report objects on your grid determines whether your report data can be displayed using a style of graph.

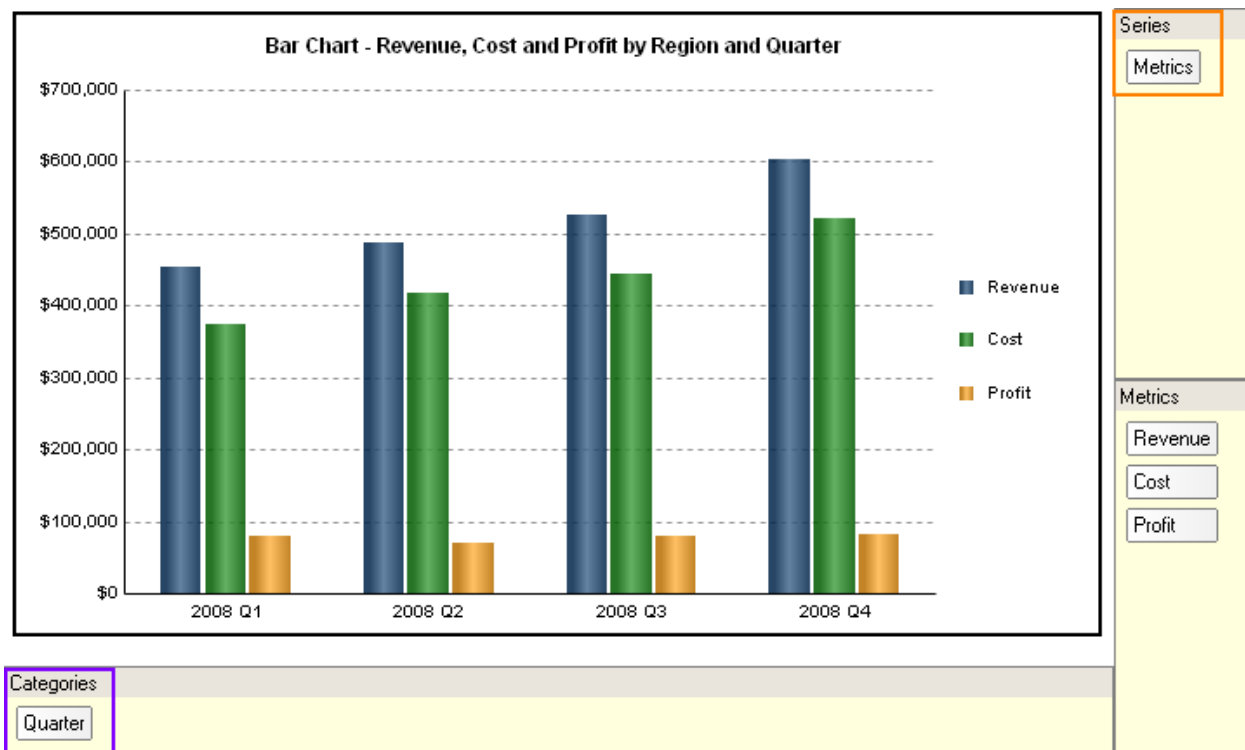
Moving objects on a graph report

A graph report's layout depends on the location of report objects on the grid of the report. For example, a grid report with its metrics in the rows and attributes in the columns produces a different graph than a grid report with the same metrics in the columns and attributes in the rows.

Graph reports consist of categories and series. A summary of these elements of graphing is provided in the table below.

Graphing Term	Definition/How it appears in reports
Categories	- Data along the X-axis of a graph report - Correspond to the rows of a grid report - Usually represent attributes
Series	- Data along the Y-axis of a graph report - Correspond to the columns of a grid report - Displayed in legend of graph report - Usually represent metrics

Consider the following example. The Bar graph shown below displays revenue, cost, and profit by region and quarter.

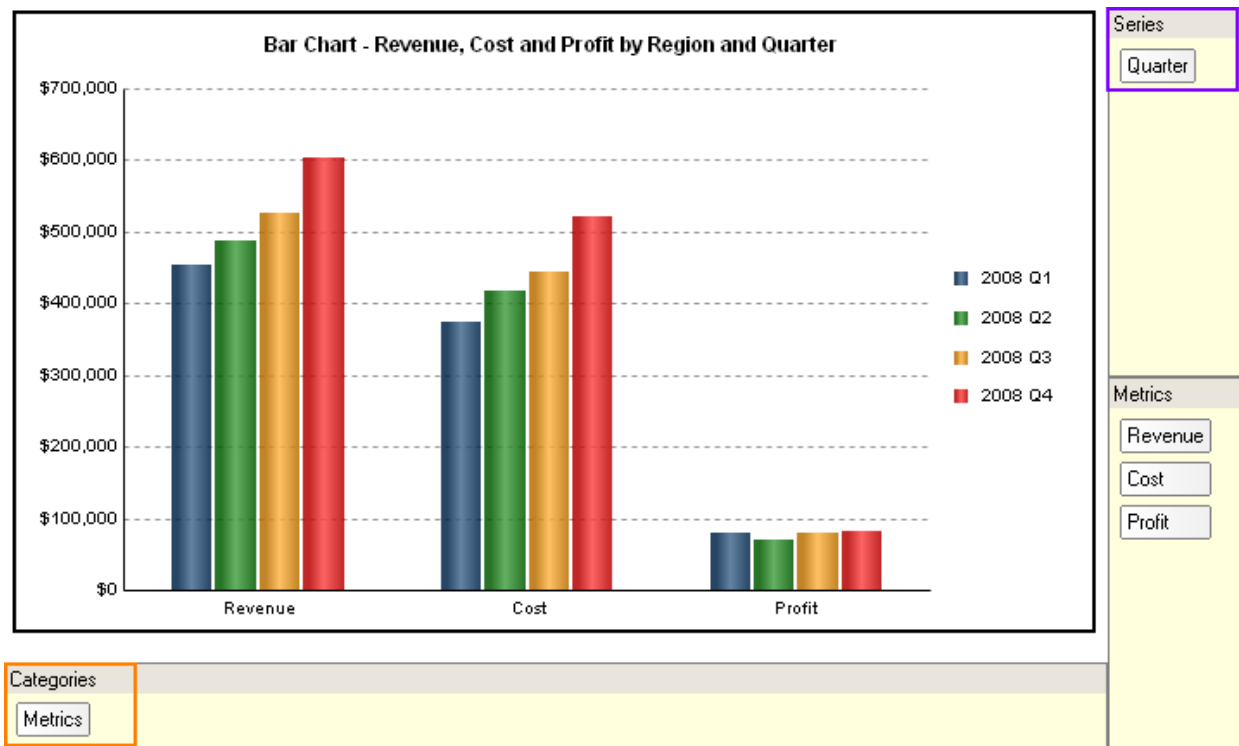




This graph report, named Bar Chart - Revenue, Cost and Profit by Region and Quarter, is available in the MicroStrategy Tutorial project.

Notice in the graph report shown above that the attribute Quarter is included in the categories, while the metrics Revenue, Cost, and Profit are included in the series. This displays a cluster of three bars for the Revenue, Cost, and Profit metrics. Each cluster represents the metric values for a particular quarter. This view of data displays how revenue, cost, and profit affect each other for a given quarter.

You can change the focus of the graph report simply by changing the series and categories. For example, the same graph report is used to display the graph below.



Notice that the only change that has been made is that the Quarter attribute is now a part of the series, while the metrics Revenue, Cost, and Profit are part of the categories. Each cluster of bars now represents a single metric over time. This view of data displays how revenue, profit, and cost have increased or decreased over four quarters.

As shown in the previous examples, how you situate report objects such as attributes and metrics on a grid directly affects how data is displayed on your

graph, which in turn can change the analytical focus of the graph. It is helpful to keep this in mind as you design reports you intend to display as graphs.



To display or hide the series and categories drop zones in MicroStrategy Developer, click **Show Drop Zones**  in the toolbar.

It is also important to remember that each graph type has its own data requirements, and changing the location of objects for one graph type can have a vastly different effect than it does for another graph type. For information on the requirements for each graph style, see [Minimum object requirements for each graph style, page 374](#).

Displaying graph data without scrollbars

On some graph reports, the entire set of data may not fit onto one page of the graph.

- If the labels along the X-axis extend beyond one page, a horizontal scroll bar appears at the bottom of the graph for you to scroll through the rest of the data.
- If the graphic elements of your graph, such as bars or lines, extend beyond one page, a vertical scroll bar appears to the right of the graph for you to scroll through the rest of the data. For example, if you have many attributes in the columns of your grid, not all of them can be displayed on one page.

To fit the data on a single graph page, without scroll bars, use the following procedure to specify the number of categories and series in your graph.

To change the number of categories and series in a graph

- 1 View the report in Graph View. For steps to view a report as a graph, see [Viewing graph reports, page 371](#).
- 2 In MicroStrategy Web, from the **Format** menu, select **Graph**. The Format:Graph dialog box opens.
- 3 Click **General**.
- 4 In the Maximum area, type the maximum number of categories and series in the Categories and Series fields respectively. For example, if you have 10 attributes in the columns of the grid and not all of them are

appearing on one graph page, you can specify 10 in the Series field. This ensures that all 10 of the attributes appear on one page of the graph.

- 5 Click **OK**.
- 6 Save the report to save you graph preferences.

Graph style usage and examples

This section describes the graph styles available in MicroStrategy Developer and MicroStrategy Web. It provides a general description and images of these different graph styles, and also offers insight on the type of data best presented with each style, as well as tips to create the graphs in MicroStrategy.



A separate section, [Advanced graph style usage and examples, page 408](#), discusses the Advanced graph styles available in MicroStrategy, including Gantt charts and Boxplot graphs.

This chapter provides information about the following graph styles:

- [Area, page 382](#)
- [Bar, page 384](#)
- [Line, page 392](#)
- [Pie chart, page 395](#)
- [Scatter plot, page 398](#)
- [Polar chart, page 400](#)
- [Radar chart, page 402](#)
- [Bubble chart, page 405](#)
- [Combination, page 407](#)
- [Custom, page 408](#)

How you position report objects on a grid report determines whether a graph can be generated in the selected style. For more information, including an example, see [Placement of report objects and choice of graph styles, page 376](#). This is something to keep in mind as you experiment with different graph styles and ways to position report objects on a grid.

Area

An Area graph emphasizes changes in data values by connecting various data points along a line, and then filling in the portion of the graph beneath the line. Therefore, Area graphs can often resemble a mountain range of high peaks and low valleys, with each peak and valley representing a different data value.

Area graph types

You can create several types of Area graphs in MicroStrategy, including the following:

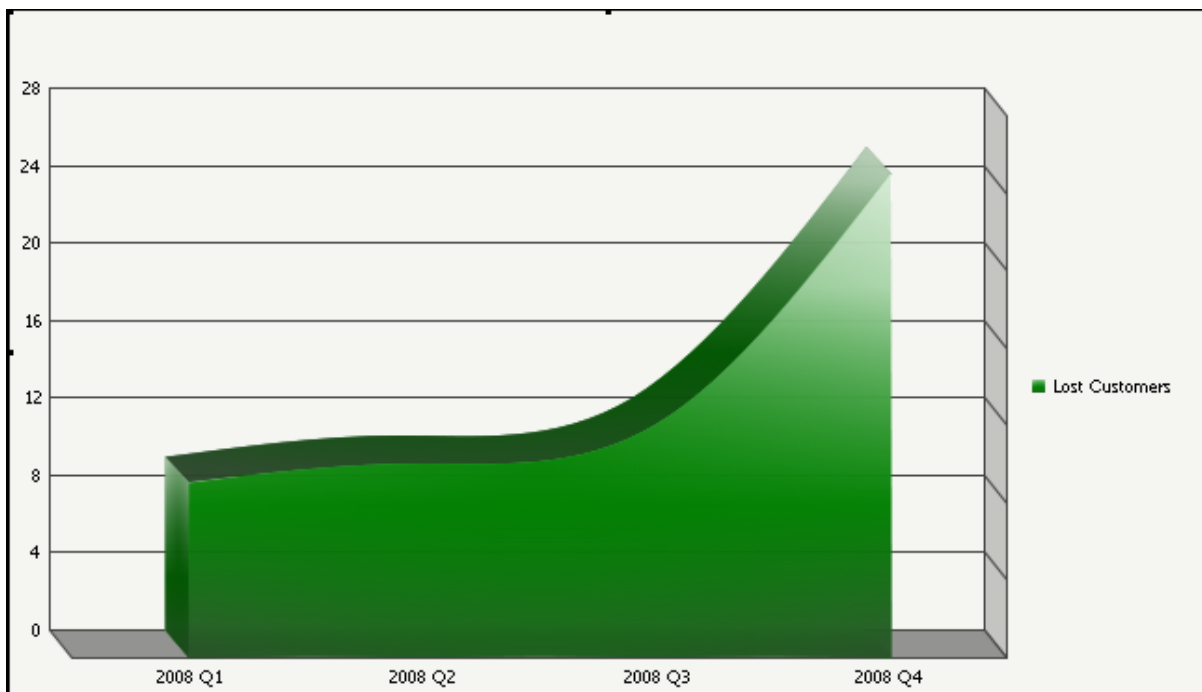
- Horizontal Area or Vertical Area styles:
 - Absolute
 - Stacked
 - Bi-polar Absolute
 - Bi-polar Stacked
 - Dual-axis Absolute
 - Dual-axis Stacked
 - Percent
- 3D Area styles:
 - Series
 - Group

Vertical Area: Absolute graph example

One of the types of Area graphs you can create in MicroStrategy is a Vertical Area: Absolute graph. You can think of this type of graph as a vertical Line graph that has the area beneath it filled in a certain color.

For example, the following graph provides information about how many customers a company lost per quarter. The top edge of the graph represents the various data points, in this case, the number of customers lost. The X-axis represents the different quarters. In the graph, notice the sharp increase in lost customers between Q3 2008 and Q4 2008.

Going into Q3, the company had lost only a total of 12 customers. During Q4 alone, however, the company saw that number grow to a total of 25 customers, as shown by the highest point on the far right of the graph.



Using Area graphs effectively

- Overall, the Area graph is a flexible graph format because you can create one from a variety of attributes, metrics, and other report objects. Its minimum requirements are that one attribute and one metric be present on the report grid. However, as explained in [Moving objects on a graph report, page 378](#), the location of these objects on your graph report determines whether and how the grid report can be generated as an Area graph.
- You can apply a curved effect to the lines of the graph, as shown in the image above. To do so, from the **Graph** menu, select **Preferences**. Within the **General** category, select the **Enable curved areas** check box.
- For detailed information about specific formatting, scaling, and other options for Area graphs in MicroStrategy, see the *MicroStrategy Developer help*. (From within MicroStrategy Developer, press F1.)

Bar

Bar graphs, also known as column graphs, are one of the most commonly used graphs because they are easy to use, flexible, and allow you to easily compare a wide range of data values.

Bar graph types

You can create many types of Bar graphs in MicroStrategy, including the following:

- Horizontal Bar and Vertical Bar styles:
 - Absolute
 - Clustered
 - Stacked
 - Dual-axis Absolute
 - Dual-axis Clustered
 - Dual-axis Stacked
 - Bi-polar Absolute
 - Bi-polar Clustered
 - Bi-polar Stacked
 - Percent
- 3D Bar styles:
 - Riser
 - Floating

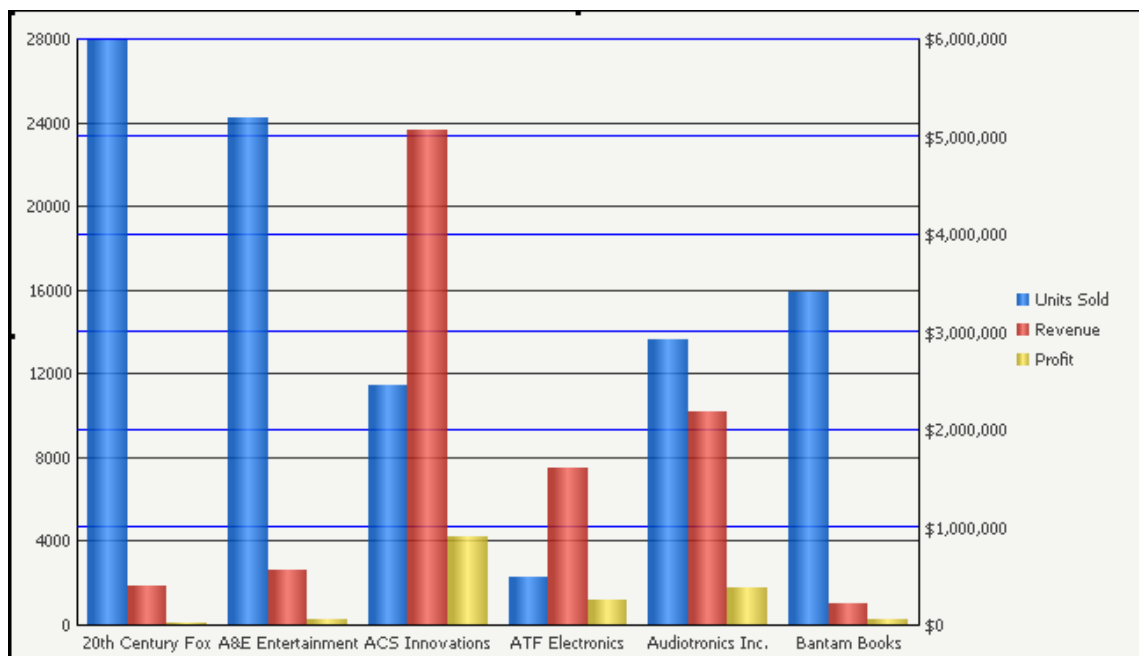
MicroStrategy also includes Budgeting graph types that provide special Bar graph formatting of the values on a graph:

- Horizontal Lipstick and Vertical Lipstick, see [Lipstick and Overlapping Bar graphs, page 387](#).
- Horizontal Overlapping Bars and Vertical Overlapping Bars, see [Lipstick and Overlapping Bar graphs, page 387](#).

Vertical Bar: Dual-Axis Clustered graph example

One type of Bar graph you can create in MicroStrategy is a Vertical Bar: Dual-Axis Clustered graph. This traditional vertical Bar graph generally plots data within individual categories, with each bar representing a different measurement.

For example, the graph below allows you to analyze the revenue and profit generated by different suppliers, as well as the number of units each sold. The X-axis is divided by supplier (ACS Innovations, for example), the Y1-axis on the left represents the number of units sold, and the Y2-axis on the right represents dollar amounts.



Notice that 20th Century Fox sold the most units, as represented by the tallest blue (the leftmost column for each category) column on the graph. The suppliers' smaller red (the middle column for each category) and yellow (the right-most column for each category) columns represent revenue and profit respectively. You can determine how much revenue and profit ACS Innovations generated by identifying the top of each column and comparing it against the dollar amounts on the right.

In the grid report, notice that the Units Sold metric is the first object on the grid's columns. This is important because the Units Sold metric's placement on the grid determines that the Y1-axis on the left side of the graph will represent the number of units sold.

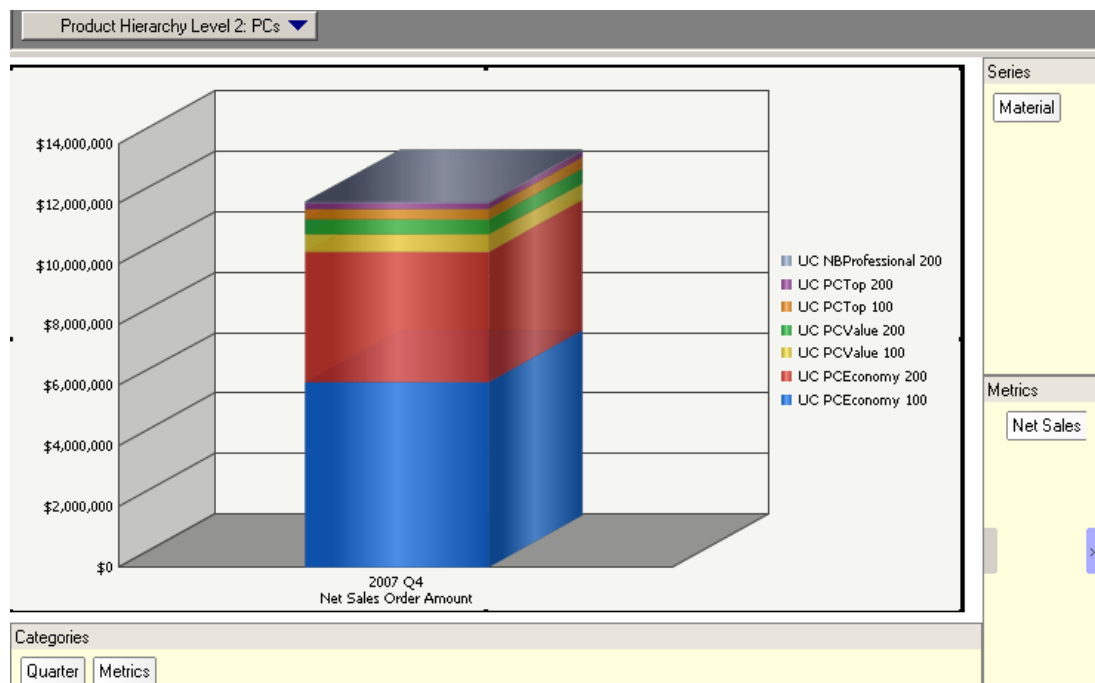
Revenue and profit are measured in the same way, in that both are calculated by currency. Therefore, it is convenient for these metrics to reside to the right of the Units Sold metric in the columns. This way, a Y-2 axis on the right side of the graph can measure both Revenue and Profit along the same axis.

As shown in the graph above, you can allow some graph types to have two numeric axes (Y1 and Y2). This allows you to chart some of the series on one axis and some of the series on another axis. This is most helpful when the series use different value formatting. For example, the Units Sold data is displayed as a general count while Revenue and Profit data are displayed in a currency format.

Vertical Bar: Stacked graph example

Another type of Bar graph you can create in MicroStrategy is a Vertical Bar: Stacked graph. This vertical Bar graph plots data within one column, with individual sections of the column representing a different measurement.

For example, the graph below provides information about the amount of net sales generated for each material (in this case, servers) in 2007 Q4. On the grid report, notice that the Material attribute is placed within the series and Product Hierarchy Level 2 attribute is placed within the Page-by field; having this attribute in the page-by field allows you to switch between different graph data for each specific material. For instance, since PCs is selected in the Page-by field, the graph below reveals net sales data about the different PCs purchased by customers in Q4 2007.

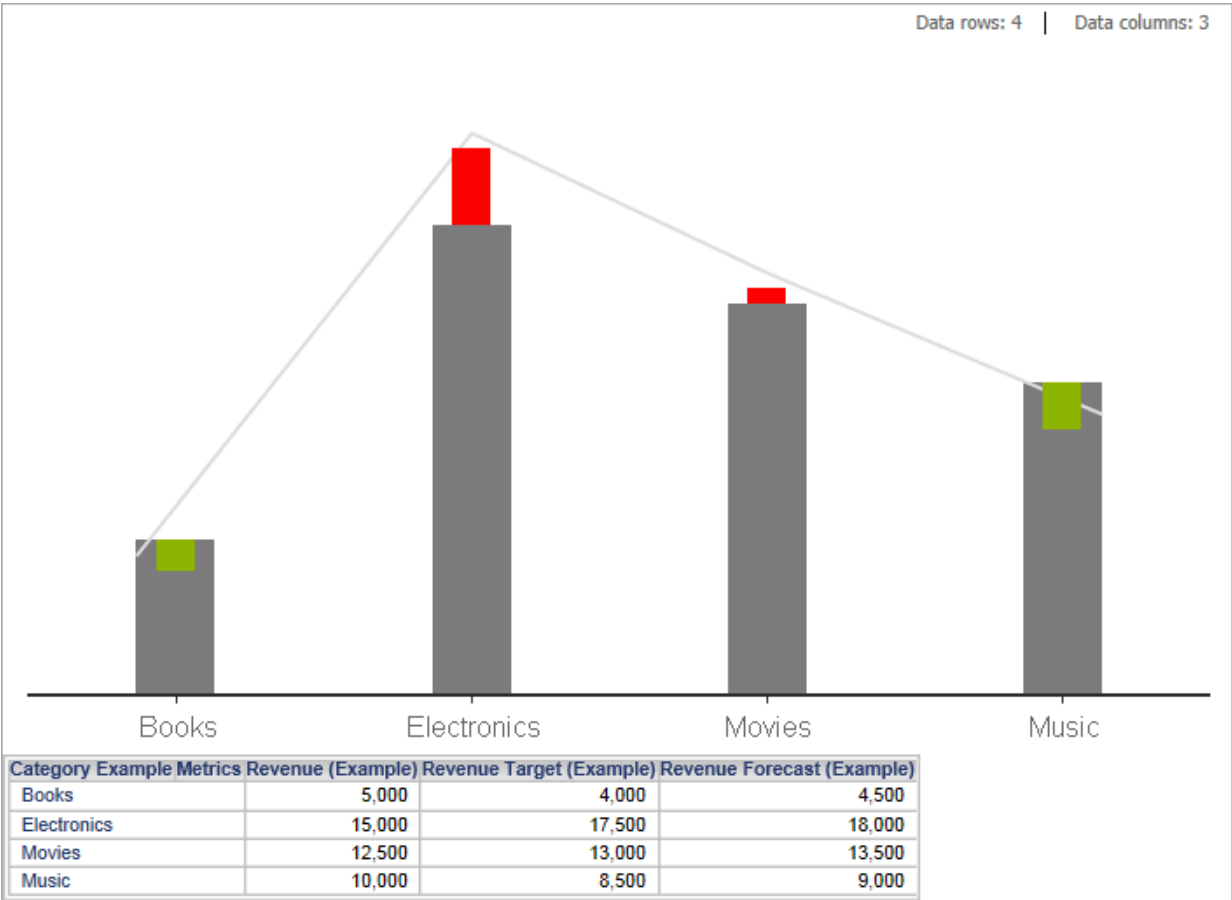


Lipstick and Overlapping Bar graphs

MicroStrategy includes the Lipstick and Overlapping Bar graph types to provide special formatting of values on a graph. These Budgeting graph types are provided to compare two sets of data that are closely related. The most common use is to compare actual data, such as revenue, and the targets that were set prior. Another common use is to compare the same actual data versus a forecast for the same data that was created using statistical analysis.

These Budgeting graph types help to compare two sets of data (the first two metrics on the graph) that are closely related in the following ways:

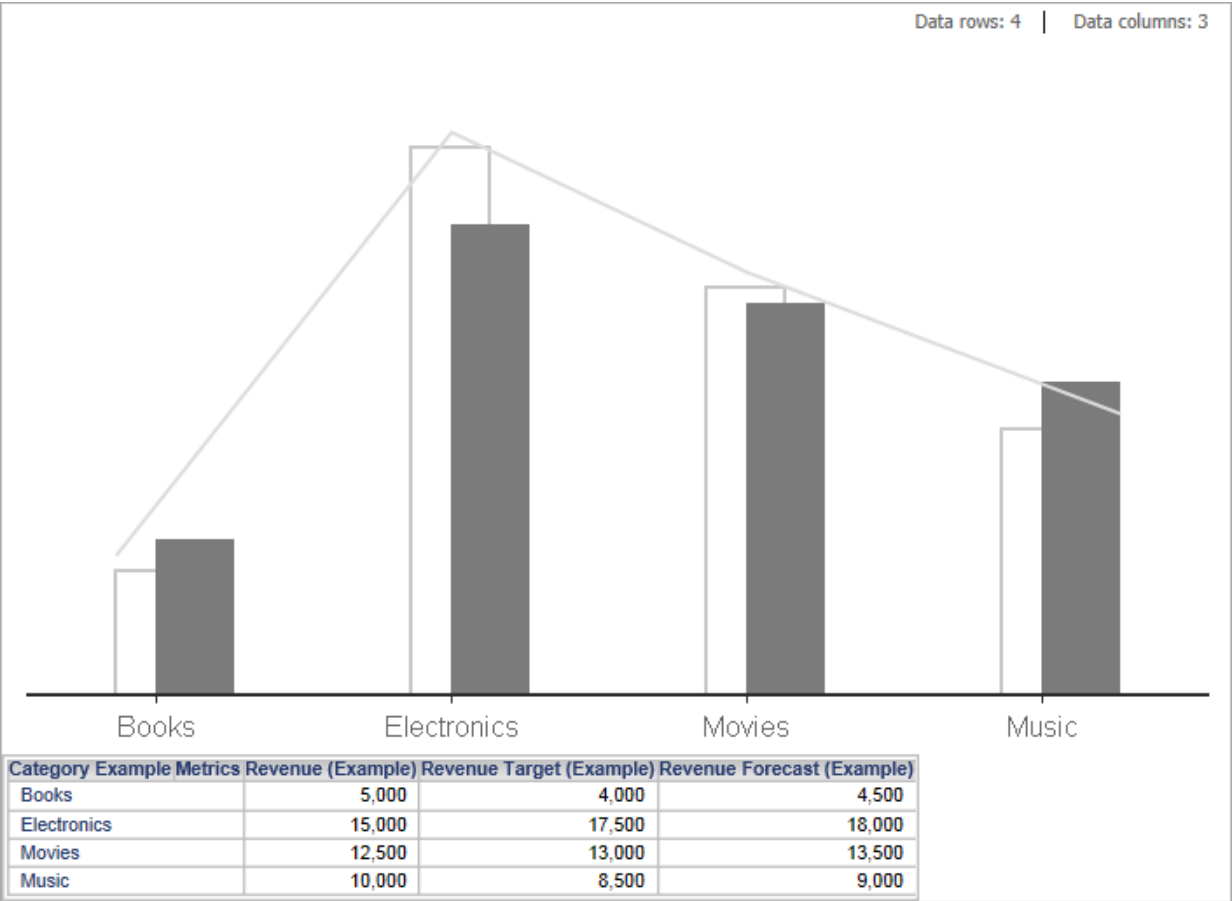
- Lipstick Bar graph: A Lipstick Bar graph combines what are traditionally two separate bars on a graph into a single bar with distinctive formatting, as shown in the image below.



By default, the data for the first metric is displayed as a standard bar on the graph with grey formatting, while the data for the second metric is displayed in one of the following ways:

- If the second metric is less than the first metric, the second thinner bar is displayed as a green bar that starts at the value of the second metric and stops at the value for the first metric. For example, in the Lipstick Bar graph shown above, when the target revenue value is lower than the actual value for revenue, as seen with books and music, this shows that results are beating expected targets. A green bar is shown to display the difference.
- If the second metric is greater than the first metric, the second thinner bar is displayed as a red bar that starts at the value of the first metric and stops at the value for the second metric. For example, in the Lipstick Bar graph shown above, when the target revenue value is higher than the actual value for revenue, as seen with electronics and movies, this shows that results are not meeting expected targets. A red bar is shown to display the difference.
- If no data exists for the first metric, or its values are null, then no bar is displayed for the first metric. Instead, a bar is only displayed for the value of the second metric. By default, these bars for the second metric are displayed with a lighter grey color than the bars displayed when the first metric has a valid value.

- **Overlapping Bar graph:** An Overlapping Bar graph displays two separate bars on a graph with an overlap of the bars to easily compare the values, as shown in the image below.



By default, the bars are displayed with contrasting colors and formatting, which makes it easier to quickly compare the data.

Since Overlapping Bar graphs use a slight overlap to compare two metric values, you can use this graph type for a wider range of analysis as compared to a Lipstick Bar graph. The Lipstick Bar graph formatting implies a direct relation of expected versus recorded values between the data being compared. With an Overlapping Bar graph you can also compare values such as the price of an item versus the cost of an item.

While the first two metrics on a Lipstick or Overlapping Bar graph are used to display and format the bars of the graph, additional metrics can also be included. Any additional metrics are displayed as simple line graphs, which can be used to display trends or highlight minimum and maximum values. In the example reports shown above, the revenue forecast is displayed as a line on the Lipstick and Overlapping Bar graphs.

The steps below show you how to create a Lipstick or Overlapping Bar graph.

Prerequisite:

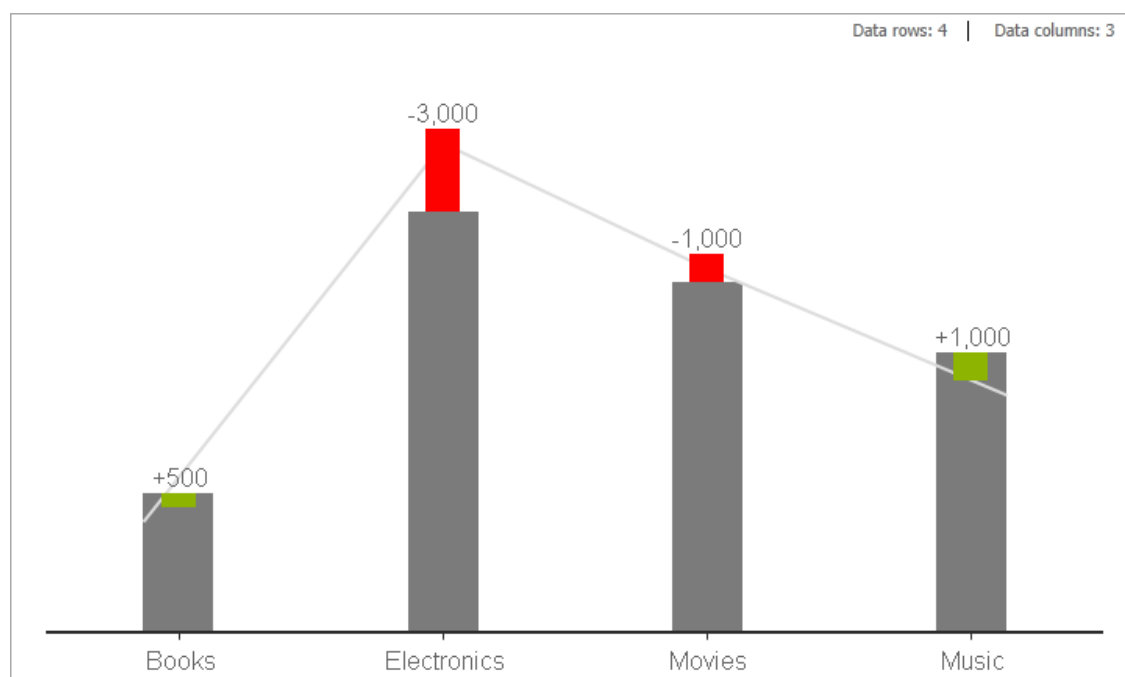
- Create a report with at least one attribute and two metrics. Additional metrics are displayed as lines on the graph.

To create a Lipstick or Overlapping Bar graph

- 1 In MicroStrategy Web, navigate to and click on the report to be displayed as a Lipstick or Overlapping Bar graph. The report is displayed.
- 2 From the toolbar, click the **Graph** icon to display the report as a graph.
- 3 From the **Format** menu, select **Graph**. The Format: Graph dialog box opens.
- 4 On the **General** tab, from the **Graph Type** drop-down list, select **Budgeting**.
- 5 From the **Graph Sub-type** drop-down list, select one of the following:
 - **Vertical Lipstick:** A Lipstick Bar graph in which the bars of the graph are displayed vertically.
 - **Horizontal Lipstick:** A Lipstick Bar graph in which the bars of the graph are displayed horizontally.
 - **Vertical Overlapping Bars:** An Overlapping Bar graph in which the bars of the graph are displayed vertically.
 - **Horizontal Overlapping Bars:** An Overlapping Bar graph in which the bars of the graph are displayed horizontally.
- 6 In the Max Bar Size area, you can define the size of the bars using the Type drop-down list. By default, Automatic is selected, which means that the size of the bars is determined automatically, based on the data for the graph. If you select Manual, you can type a value to define the size of the bars. The maximum value of 1 displays the bars so that the bars of adjacent categories on the graph appear to be touching.

7 You can perform additional formatting for the graph, such as:

- For Lipstick Bar graphs, you can display the value of the difference between the two values being compared:
 - a On the **Format** tab, from the drop-down list at the top, select **Increment/Decrement Value**.
 - b From the Format area, select the **Show** check box. This displays the value of the difference between the two values being compared, as shown in the graph below.



8 Perform any other graph formatting required for the graph report, and then save your changes. Once your changes are saved using MicroStrategy Web, you can view or modify this graph in MicroStrategy Web.

Using Bar graphs effectively

- The Bar graph is a flexible graph format since you can create one from any number of attributes, metrics, and other report objects. Its minimum requirements are that one attribute and one metric be present on the report grid. However, as explained in [Moving objects on a graph report, page 378](#), the location of these objects on your graph report determines whether and how the grid report can be generated as a Bar graph.

- When you are using a Vertical Bar: Dual-Axis Clustered graph (see [Vertical Bar: Dual-Axis Clustered graph example, page 385](#)), or other graphs similar to it, think about the order in which your metrics appear on the grid report. The order of the metrics in the series for the graph report determines which metrics are represented by the Y-1 and Y-2 axes. For example, in the Vertical Bar: Dual-Axis Clustered graph above, the order of the metrics in the series determines what the Y-1 and Y-2 axes represent.
- As shown in the graph in [Vertical Bar: Dual-Axis Clustered graph example, page 385](#), you can allow some graph types to have two numeric axes (Y1 and Y2). This allows you to chart some of the series on one axis and some of the series on another axis. This is most helpful when the series use different value formatting. For example, the Units Sold data is displayed as a general count format, while Revenue and Profit data are displayed in a currency format.
- Placing an attribute in the page-by field can not only enhance the look and feel of your graph, it is often an essential step in the graph design process. For example, recall the Vertical Bar: Stacked graph example above. If you move the Product Hierarchy Level 2 attribute off the page-by field onto the grid, the graph presents an excessive number of columns, making it difficult to focus on specific data.
- Apply a gradient to the bars in your graph to make them stand out from the other sections of the graph. For example, the graph above has a gradient applied to its series. To apply gradients and other effects, right-click your graph's series and access the Preferences dialog box. For these steps, see [Manually formatting graph series and objects, page 451](#).
- If you are comparing two sets of data that are closely related, such as actual and target data, or actual and forecast data, consider using a Lipstick or Overlapping Bar graph. These Budgeting graph types are described in [Lipstick and Overlapping Bar graphs, page 387](#).
- For detailed information about the specific options for Bar graphs in MicroStrategy, see the *MicroStrategy Developer help*. (From within MicroStrategy Developer, press F1.)

Line

A Line graph is a type of graph that highlights trends by connecting lines between different data points. Like the Bar graph and Pie chart, the Line graph is one of the most commonly used graph types. It is often used as a visual alternative to Bar graphs and other types.

Line graph types

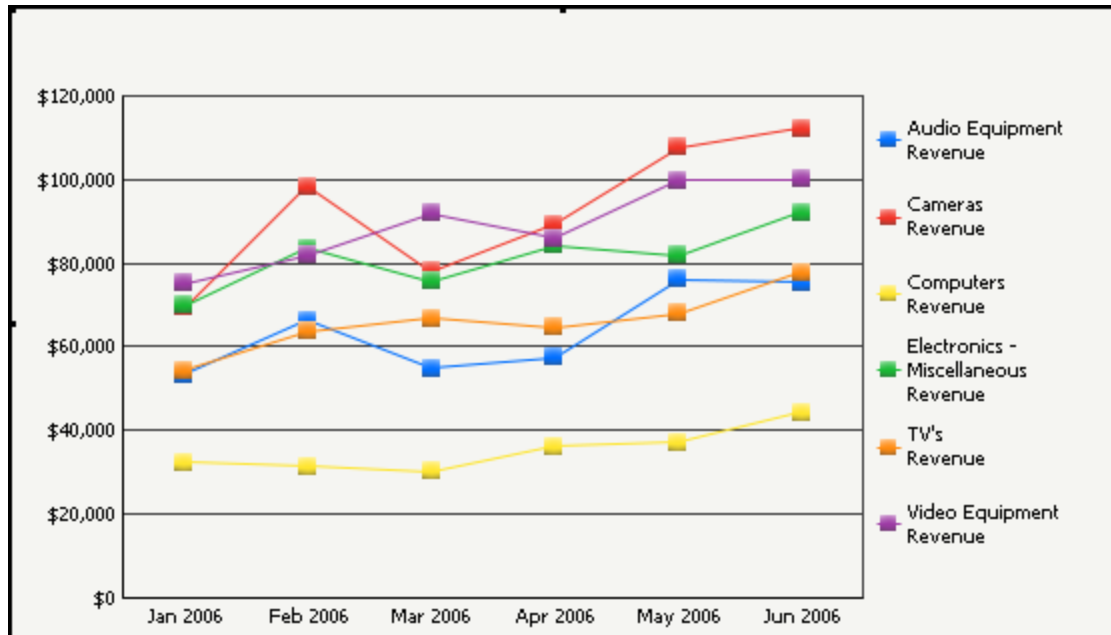
You can create many different types of Line graphs in MicroStrategy, including the following:

- Horizontal Line and Vertical Line styles:
 - Absolute
 - Stacked
 - Bi-polar Absolute
 - Bi-polar Stacked
 - Dual-axis Absolute
 - Dual-axis stacked
 - Percent
- 3D Line styles:
 - Series
 - Group

Vertical Line: Absolute graph example

One of the many variations of vertical Line graphs you can create in MicroStrategy is the Vertical Line: Absolute graph.

The following graph highlights the amount of revenue generated for various subcategories of Electronics per month.



Each colored line in the graph represents a different electronics subcategory, such as Cameras or TVs. The X-axis represents different months in chronological order and the Y-axis represents the amount of revenue. The upper most line in the graph represents Camera revenue. To analyze how much revenue was generated during a specific month, follow the appropriate line from left to right until you reach the correct Month along the X-axis. For example, follow the line for Cameras to discover that, in May 2006, revenue generated by camera sales was just over \$100,000.

Using Line graphs effectively

- The Line graph is a flexible graph format because you can create one from any number of attributes, metrics, and other report objects. Its minimum requirements are that one attribute and one metric be present on the report grid. However, as explained in [Moving objects on a graph report, page 378](#), the location of these objects on your graph report determines whether and how the grid report can be generated as a Line graph.
- You can apply a curved effect to the Line graph. To do so, from the **Graph** menu, select **Preferences**. Expand **Options**, and select the **General** category. Select the **Enable curved lines** check box.

- For detailed information about the specific options for Line graphs in MicroStrategy, see the *MicroStrategy Developer help*. (From within MicroStrategy Developer, press F1.)

Pie chart

A Pie chart is a style of graph in which data values such as percentages are represented as proportionally-sized slices of a pie.

Pie chart graph types

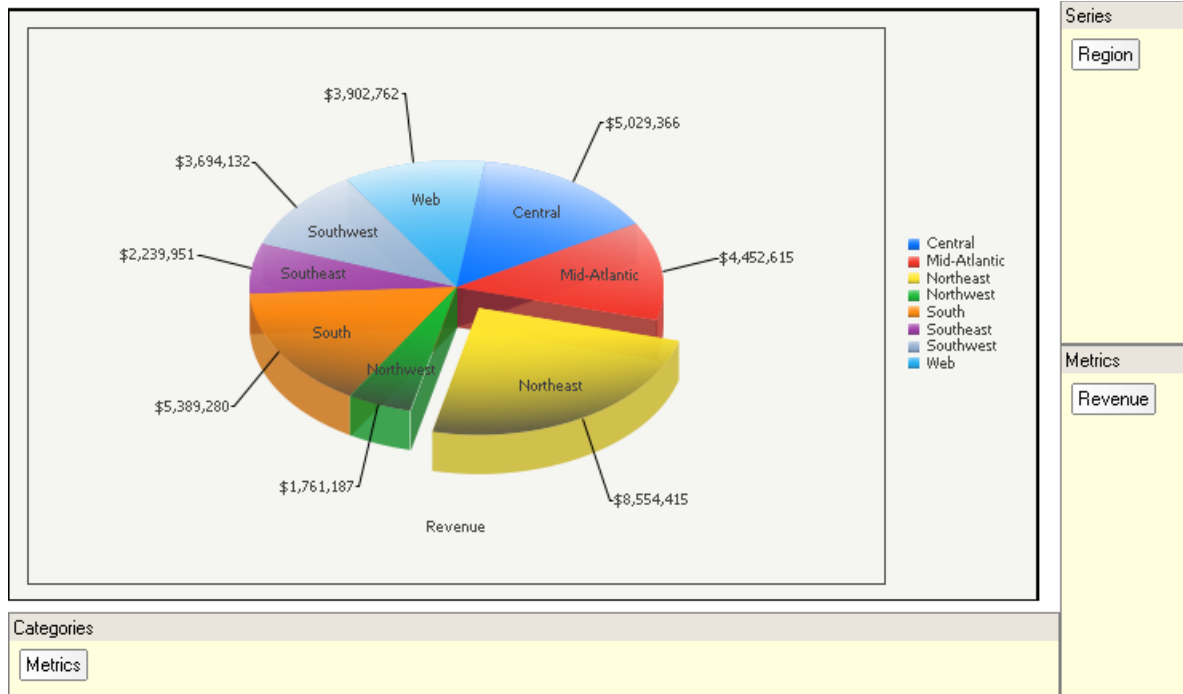
You can create several different types of Pie charts in MicroStrategy, including the following:

- Pie
- Ring
- Multiple Proportional Pie
- Multiple Proportional Ring

Pie chart example

In the following exploded Pie chart example, each slice of the graph represents a different region, for example, the Northeast or Mid-Atlantic. The size of each pie slice is determined by the values of the Revenue metric. For example, the Southeast region had \$2.2 million in revenue, compared to

over \$8.5 million in the Northeast. The Northeast slice has been moved away from the pie slightly to draw attention to it.



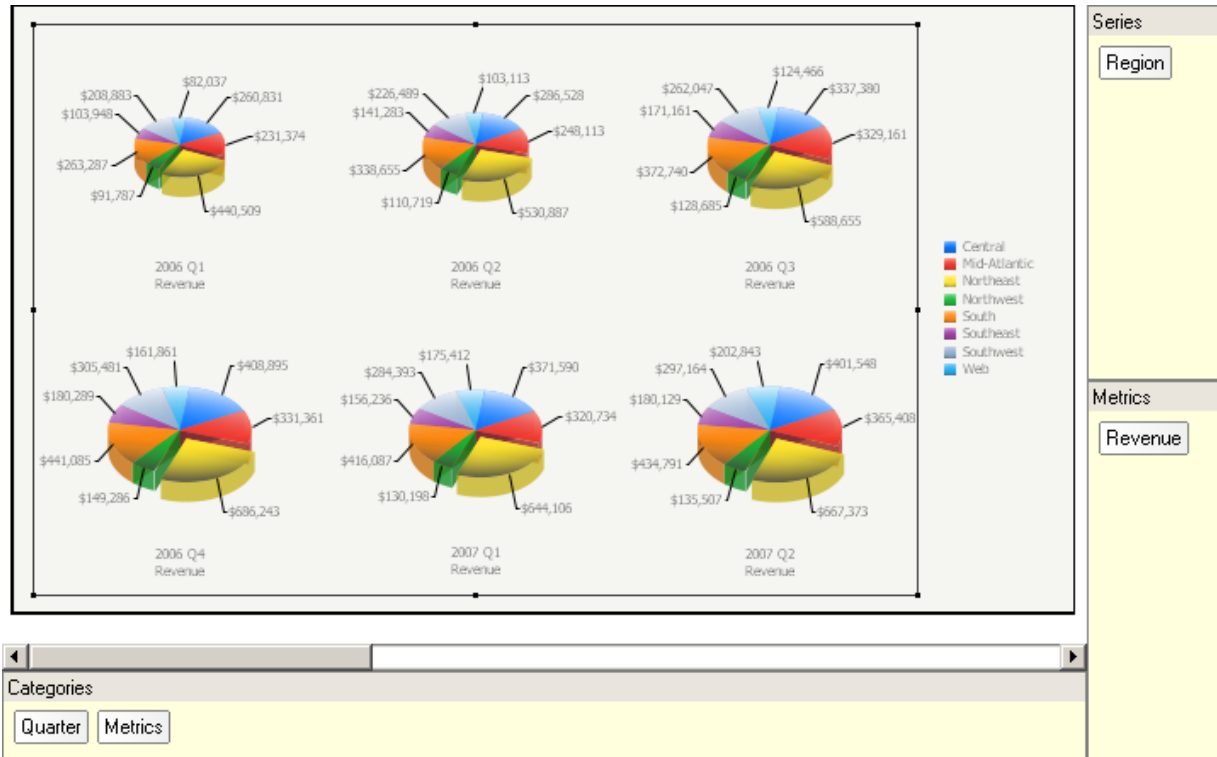
On the graph report, notice that the Region attribute is included in the series for the graph, and the Revenue metric is included in the categories for the graph. This causes the elements of the Region attribute, the regions themselves, to appear as slices in the Pie chart. Each region is assigned its own color to distinguish it from the other regions.

However, if you move the attribute to the categories and the metric to the series on the graph report, multiple Pie charts are created for each region. If you then view the report as a graph, each Pie chart is not divided into slices, which makes it more difficult for you to compare values. Keep this in mind when you are designing your own graph.

Pie: Multiple Proportional Pies example

The graph below is an example of a Pie chart displayed as multiple pies. Each pie is formatted identically to the single graph in the previous example. In this graph, each pie represents the total revenue for a quarter, divided among several geographic regions. The pies are of different sizes depending on the total amount of revenue for that quarter. For example, the pie for 2006 Q4 is

visibly larger than the pie for 2006 Q1, because the total revenue for Q4 was nearly \$1,000,000 greater than the total revenue for Q1.



Notice that in the grid report above, the Quarter attribute is in the categories, on the left of the metrics. This causes the graph to display one Pie chart for each available value of Quarter. The way a graph appears depends heavily on whether the attributes and metrics are in the series or the categories of the graph report.

Using pie charts effectively

- The Pie chart is a flexible graph format because you can create one from a number of attributes, metrics, and other report objects. Its minimum requirements are that one attribute and one metric be present on the report grid. However, as explained in [Moving objects on a graph report, page 378](#), the location of these objects on your graph report determines whether and how the grid report can be generated as a pie graph.
- When you create a Pie chart, move the attributes and metrics of your report between the series and categories until a useful and easy-to-follow graph is displayed. The location of the attributes and metrics on the grid directly affects whether one Pie chart with different slices is displayed or whether multiple pies are displayed. Review the Pie chart examples above for an example of this.

- For detailed information about the specific options for Pie charts in MicroStrategy, see the *MicroStrategy Developer help*.

Scatter plot

A Scatter plot is a graph showing points of data that are not connected by a line. A Scatter plot can help you identify the relationships that exist between different values. Each dot on the graph represents the intersection of the data on the X and Y axes. For example, if you are looking at numeric profit data on a report, the profits of various call centers are clear. However, when you display the report as a Scatter plot, the visualization of the data reveals that a certain call center is performing significantly better than other call centers.

Scatter plots are often used for quality control and management. However, like most graph styles, they are also useful because of their distinct visual style.

Scatter plot graph types

You can create several different types of Scatter plots in MicroStrategy, including the following:

- X-Y Scatter
- Dual-axis X-Y Scatter
- X-Y-Z Scatter Chart (three-dimensional)

Scatter: X-Y graph example

In the Scatter: X-Y graph below, the revenue generated by each customer appears along the X-axis, while the customer satisfaction for each customer

appears along the Y-axis. The individual circles within the graph represent each customer.



This graph report, named Scatter Plot Chart - Revenue vs. Customer Satisfaction, is available in the MicroStrategy Tutorial project.

The Scatter plot shown above helps to show the relationship between a customer's satisfaction with a company and how much revenue is generated by a customer. This graph suggests that customers who are satisfied with the company are likely to spend more money with the company.

Using Scatter plots effectively

- While a Scatter plot is a flexible graph format because you can create one from a number of report objects, its minimum requirements are that one attribute and two metrics be present on the report grid. However, as explained in [Moving objects on a graph report, page 378](#), the location of these objects on your grid report determines whether and how the grid report can be generated as a Scatter plot.
- If you include more than two metrics on the report grid for a Scatter plot like the one shown above, the graph may not be useful because Scatter plots are specifically designed to plot two distinct metric values. You can include three metrics on Scatter: X-Y-Z graphs, which display the metric values on a three-dimensional Scatter plot.

- If you include only one metric on the report grid and apply a Scatter plot graph style, an error message notifies you that there is not enough data to create the Scatter plot. For details on how the placement of objects on a report can affect the display of data for various graph styles, see [Placement of report objects and choice of graph styles, page 376](#).
- For detailed information about the specific options for Scatter plots in MicroStrategy, see the *MicroStrategy Developer help*.

Polar chart

A Polar chart is a graph drawn with circular, polar coordinates. As shown in the example that follows, data points often fall within a certain distance of the middle of the circular graph and another axis.

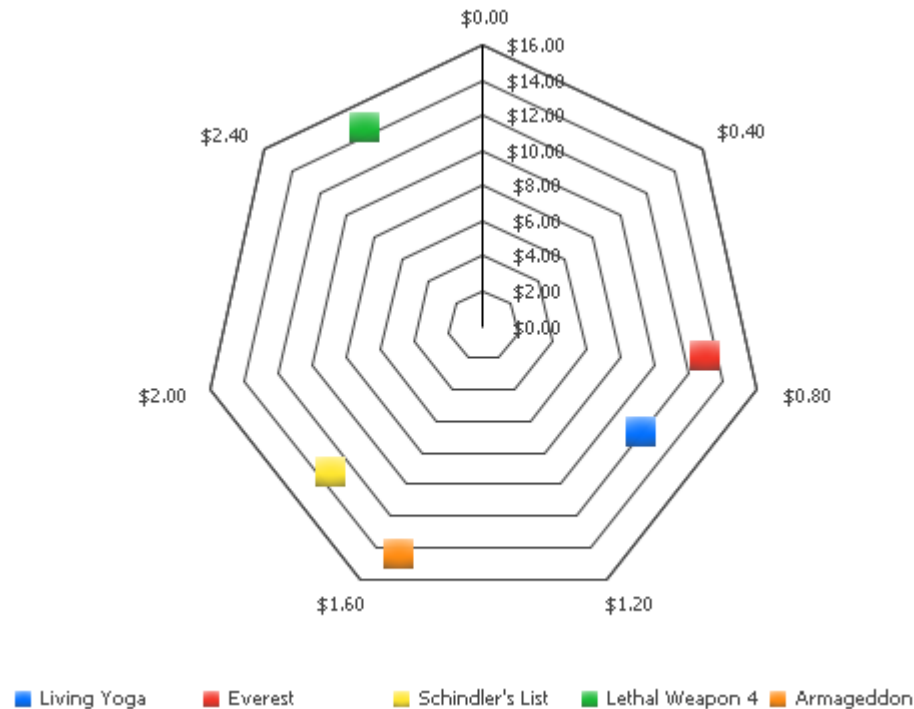
Polar chart graph types

You can create the following types of Polar charts in MicroStrategy:

- Coordinate Chart
- Dual-axis Coordinate Chart

Polar: Coordinate Chart graph example

The Polar: Coordinate Chart graph shown below provides data about the unit cost versus the unit profit for various items.



In this graph, the vertical line cutting through the Polar chart is the Y-axis, which represents the unit profit made for each item. The X-axis is represented via the seven-sided polygon. A range of values is listed in order, starting from the top, and moving clockwise around the shape. These values represent the unit cost for each item.

To determine the unit profit for each item, notice how high or low a marker resides based on the Y-axis. For example, notice in the graph above that Lethal Weapon 4 had over \$14.00 in profit, as shown by the data marker to the left of the vertical Y-axis. Another item, Everest, had between \$12.00 and \$14.00 in profit, as shown by the location of the marker along the Y axis.

To determine the unit cost for each item, line the item's data marker up with a location along the perimeter of the graph. For example, if the data marker is at the bottom of the graph, you know that the item had between \$1.20 and \$1.60 unit cost because those are the data values along the bottom of the graph. Notice in the graph above the Armageddon item's data marker can be lined up near the \$1.60 label along the outside of the graph. The Everest item has a much lower unit cost, as shown by how close its marker is to the \$.80 label on the graph.

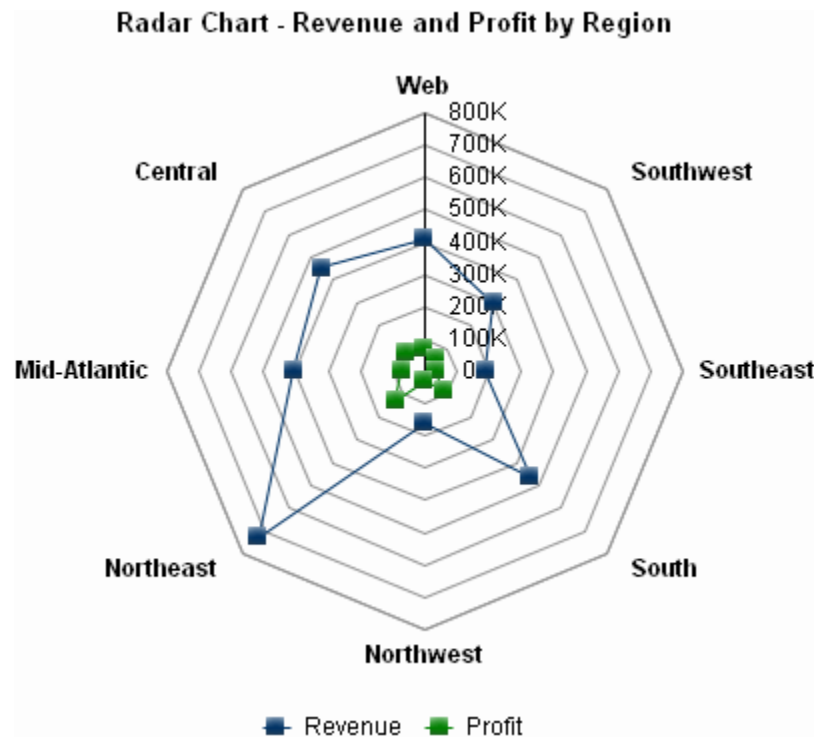
Using Polar charts effectively

- While a Polar chart is a flexible graph format because you can create one from a number of report objects, its minimum requirements are that one attribute and two metrics be present on the report grid. However, as explained in [Moving objects on a graph report, page 378](#), the location of these objects on your grid report determines whether and how the grid report can be generated as a Polar chart.
- When you design a Polar chart, try to include both the attribute and metric in the grid report's columns. This ensures that each attribute element is represented with a single marker on the Polar chart. To create the example Polar chart above, the Item attribute and the metrics Unit Cost and Unit Profit were all placed in the grid's columns. Therefore, each item is represented by its own marker on the graph.
- For detailed information about the specific options for Polar charts in MicroStrategy, see the *MicroStrategy Developer help*.

Radar chart

A Radar chart resembles a Polar chart, but works differently. Instead of allowing you to see distinct data markers, Radar charts connect these

markers to form a shape. Each corner of the shape represents a different data point, as shown in the example below.



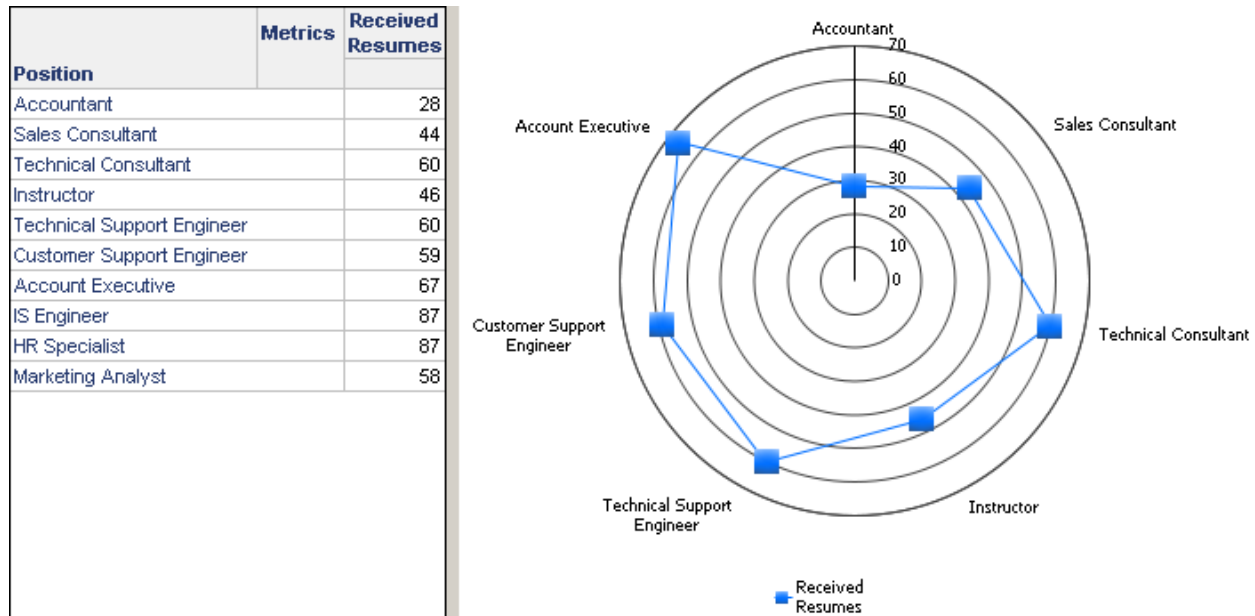
Radar chart graph types

You can create the following types of Radar charts:

- Line
- Stacked Line
- Dual-axis Line
- Dual-axis Stacked Line
- Area
- Stacked Area
- Dual-axis Area
- Dual-axis Stacked Area

Radar: Line graph example

The following graph is a Radar: Line graph. It is called a Radar: Line graph because of the line formed between the data points in the graph. The graph reveals information about the resumes received for each position in a company.



In the graph, notice that the Y-axis is represented as a vertical line that runs down the upper portion of the circular X-axis. The Y-axis lists various received resume amounts in ascending order from bottom to top. Each label on the circular X-axis represents a different position in the company, for example, Sales Consultant or Customer Support Engineer.

You can read this graph by displaying markers for each position on the graph. In the example above, each marker is displayed as a rectangle. These markers are placed based on the metric values for the attributes. For example, the marker for Customer Support Engineer is displayed between the 50 and 60 lines since there were a total of 59 resumes received for this position.

This graph sample serves as an example of the flexibility of Radar charts, as well as Polar charts. In this example, this graph gives quick insight into how many resumes are received for each position and how this relates to the amount received for other positions. This can highlight positions that a Human Resources department may need to recruit more heavily for to find additional candidates. For example, more recruiting may be necessary for the Accountant position, as only 28 resumes were received.

Using Radar charts effectively

- The Radar chart is a flexible graph format because you can create one from a number of attributes, metrics, and other report objects. Its minimum requirements are that one attribute and one metric be present on the report grid. However, as explained in [Moving objects on a graph report, page 378](#), the location of these objects on your grid report determines whether and how the grid report can be generated as a Radar chart.
- In the Radar: Line graph example above, notice that there are ten positions listed in the grid, but only seven positions appear on the graph. The IS Engineer, HR Specialist, and Marketing Analyst positions are missing from the graph because the **Maximum number of categories** for this graph style is set to seven; therefore, only seven positions appear on a page. In this case, you can ensure all of the attribute elements appear on the graph by selecting **Preferences** from the Graph menu. In the **Maximum number of categories** field, type **10**, the number of attribute elements you want to appear on the graph.
- For detailed information about the specific options for Radar charts in MicroStrategy, see the *MicroStrategy Developer help*.

Bubble chart

A Bubble chart is a Scatter plot that also displays the size of each data marker, using bubbles to represent the different sizes of data markers.

To create a Bubble chart, you must include at least one attribute and three metrics on your report grid. Where you place each object on the report grid affects how that object is displayed in the resulting graph, as follows:

- The first metric on the grid determines what is placed on the X-axis.
- The second metric on the grid determines what is placed on the Y-axis.
- The third metric on the grid determines the size of each bubble.

Bubble chart graph types

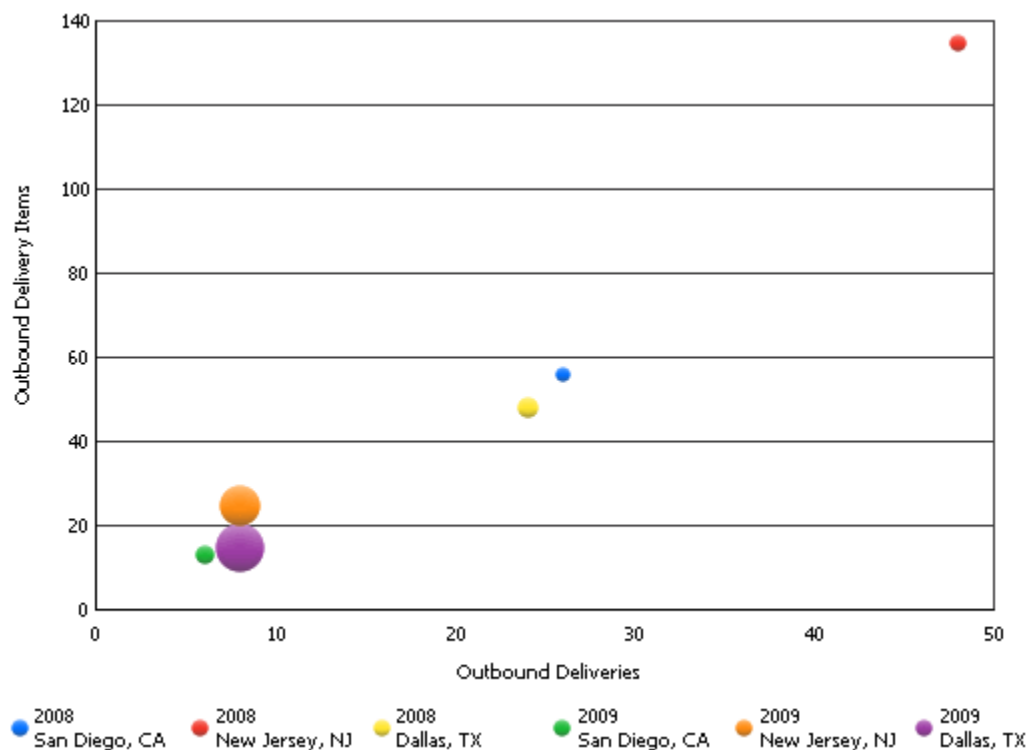
You can create the following types of Bubble charts in MicroStrategy:

- Bubble
- Dual-axis

Bubble chart example

The graph below provides an example of how three metrics are displayed within a Bubble chart. In the graph, a colored bubble is designated for each shipping point for the years 2008 and 2009. The size of each bubble represents that shipping point's average total delivery time for outbound items.

The X-axis represents the Outbound Deliveries metric, which displays the number of outbound deliveries. The Y-axis represents the Outbound Delivery Items metric. A third metric, Average Total Delivery Time for Outbound Deliveries (Days), is included as the third metric on the columns of the grid report. The third metric placed in the columns determines the size of each bubble.



Notice that the largest bubble in the graph is the one for the 2009 Dallas, TX; this shows that the average delivery time was the largest for the Dallas, TX shipping point during the year 2009. Its placement along the X-axis informs you that the Dallas, TX shipping point for 2009 had fewer than 10 outbound deliveries. The bubble's vertical position along the Y-axis emphasizes that the Dallas, TX shipping point for 2009 shipped fewer than 20 outbound items for all outbound deliveries. This analysis suggests that further analysis needs to be done to determine why shipping times were so long for this shipping point in 2009, since the number of outbound deliveries and delivery items were both relatively small.

Using Bubble charts effectively

- At least one attribute and three metrics must be present on a grid report to be able to create a Bubble chart. However, to create an effective Bubble chart, it is best to avoid more than three metrics on the grid report. If the grid contains more than three metrics on the grid, the graph may not be useful because Bubble charts are designed to plot two values (two metrics) at a time. The size of each bubble represents the third metric.
- For detailed information about the specific options for Bubble charts in MicroStrategy, see the *MicroStrategy Developer help*.

Combination

Combination graphs are graphs that use default formatting to display risers with multiple marker types. For example, one combination graph can display both bars and lines together on the same graph. Combination graphs use the formatting options of other standard MicroStrategy graph types, as described below:



You can change the layout of combination graphs on the Layout category. However, if you select a three-dimensional layout, the multiple marker type formatting is lost. All other layout types retain the multiple marker type formatting for the combination graphs.

- **Bar Area:** This combination graph displays a combination of bar risers and area risers. It uses the formatting options available for Vertical Bar graphs.
- **Dual Axis Bar Area:** This combination graph displays a combination of bar risers and area risers on a dual axis. It uses the formatting options available for Vertical Bar graphs.
- **Bar Line:** This combination graph displays a combination of bar risers and line risers. It uses the formatting options available for Vertical Bar graphs.
- **Dual Axis Bar Line:** This combination graph displays a combination of bar risers and line risers on a dual axis. It uses the formatting options available for Vertical Bar graphs.
- **Area Line:** This combination graph displays a combination of area risers and line risers. It uses the formatting options available for Vertical Area graphs.

- **Dual Axis Area Line:** This combination graph displays a combination of area risers and line risers on a dual axis. It uses the formatting options available for Vertical Area graphs.

Custom

Custom graphs are a list of graph templates that are supplied with MicroStrategy products. The custom graph files (.3TF files) contain attributes and data that define the format and look of a graph.

The Custom tab has the following categories of custom graph styles:

- Basic
- Big Data Sets
- Corporate
- Gradients
- LetterSize B&W
- LetterSize Color
- Surveys
- Textured
- Trendlines

Each category corresponds to a folder in the `\support files\templates\` folder within the MicroStrategy Common Files folder. You can create your own category by adding a new folder under this directory. Each custom graph style corresponds to a 3TF file stored in that directory.

Advanced graph style usage and examples

This section describes the advanced graph styles available in MicroStrategy Developer and MicroStrategy Web. It provides general descriptions and images of these graph styles and offers insight on the type of data best presented with each style. This section also provides tips on how to create the graphs in MicroStrategy.

This section provides information about the following advanced graph styles:

- [Stock, page 409](#)
- [Histogram, page 414](#)
- [3D Surface, page 416](#)
- [Gauge, page 418](#)
- [Funnel, page 421](#)
- [Pareto chart, page 424](#)
- [Boxplot, page 427](#)
- [Gantt chart, page 430](#)



Some of the graphs illustrated in this section serve only as conceptual examples and, therefore, the data shown does not necessarily exist in MicroStrategy Tutorial.

Stock

A Stock graph is designed to plot high, low, open, and close values. It displays these values over a chosen period of time. Therefore, Stock graphs are particularly useful in depicting data that often fluctuates, such as stock prices or inventory levels.

In a Stock graph, columns are used to represent different ranges of values, from low to high. The size of each column represents the difference between the high and low values.

Stock graphs typically consist of the following elements:

- **High value:** The high value is marked at the top of each column. This represents the highest value achieved during a certain period of time.
- **Low value:** The low value is marked at the bottom of each column. This represents the lowest value achieved during a certain period of time.
- **Open value:** The open value is marked as a tail on the left side of each column. This represents the value when it is first recorded. For example, for a daily stock price, the open value represents the price of the stock at the beginning of the day, just as the stock market opens.
- **Close value:** The close value is marked as a tail on the right side of each column. This represents the value when it is last recorded. For example,

for a daily stock price, the close value represents the price of the stock when the stock market officially closes for the day.

Stock graph types

You can create several types of Stock graphs in MicroStrategy, including the following:

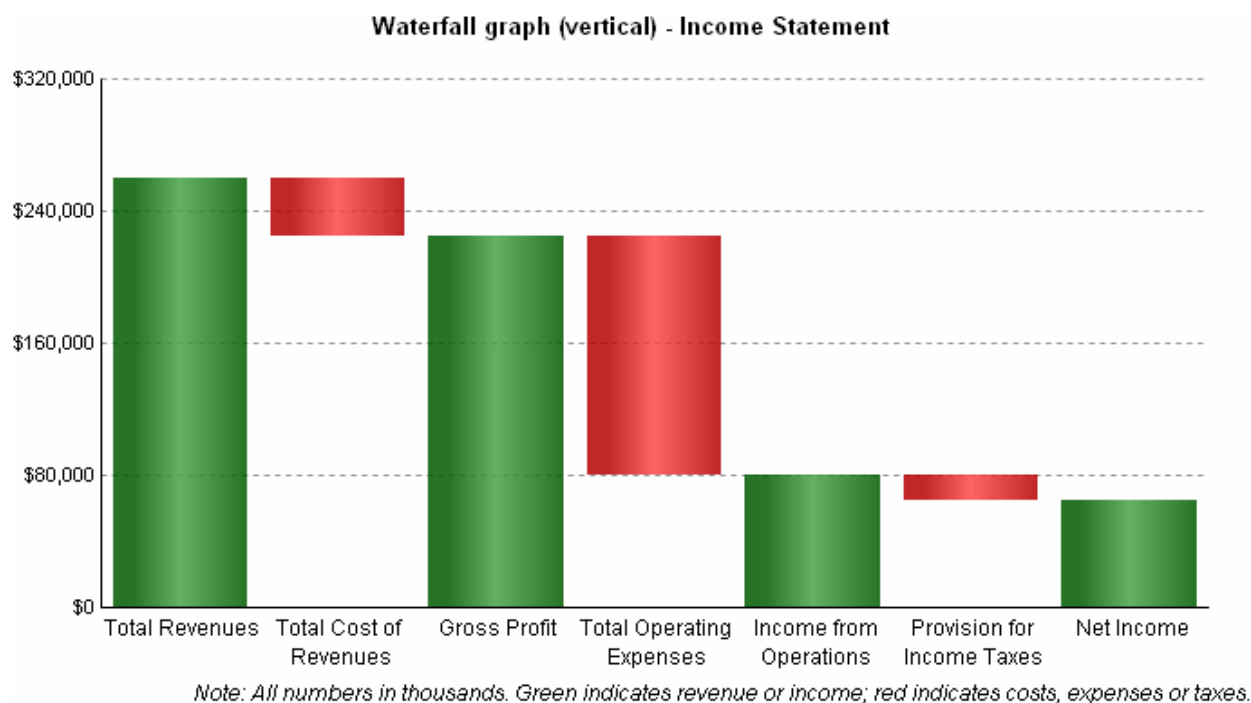
- Hi-Lo Stock graph
- Hi-Lo-Open Stock graph
- Hi-Lo-Open-Close Stock graph

Hi-Lo Stock graph example

One of the types of Stock graphs you can create in MicroStrategy is the Hi-Lo Stock graph. A Hi-Lo Stock graph allows you to view the range of a set of values between the highest and lowest values.

For example, the following Hi-Lo graph displays a waterfall-style income statement for a company. This style of graph shows all the factors involved in the net income for a company. Thresholds are applied to the report to

highlight income or revenue with green formatting, and to highlight costs or expenses with red formatting.

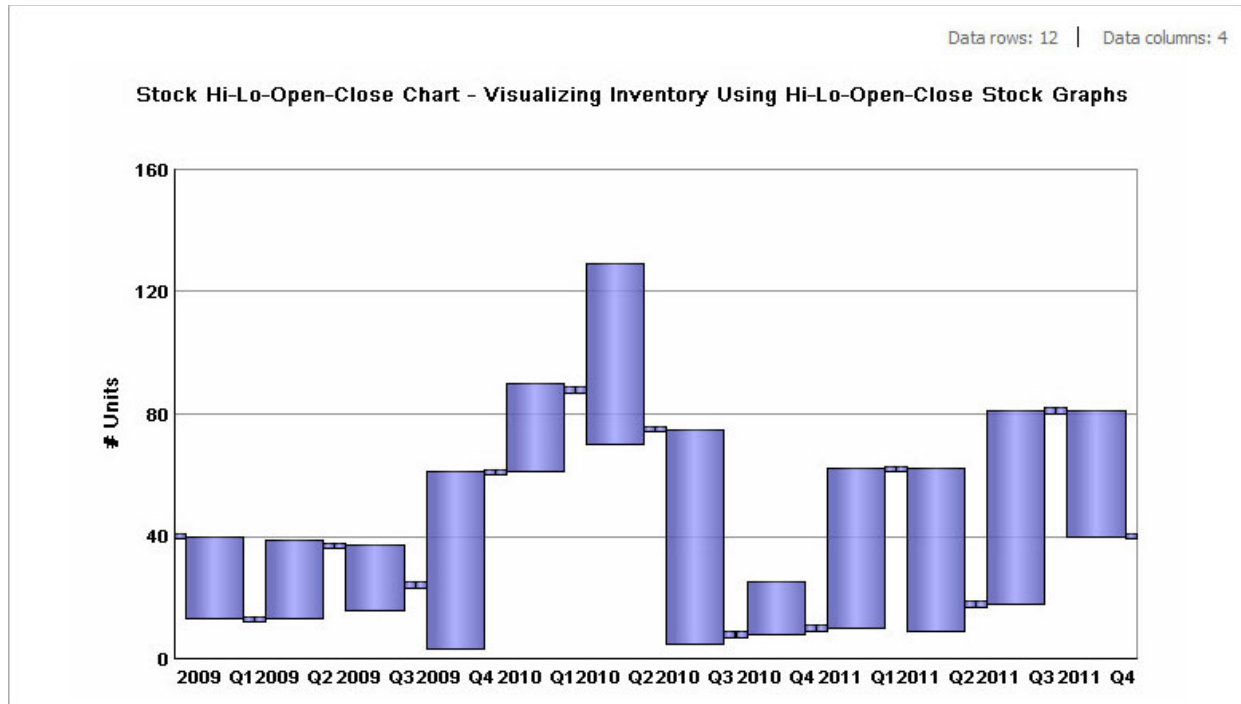


This graph report, named Waterfall graph (vertical) - Income Statement, is available in the MicroStrategy Tutorial project. In addition to providing an example of a Stock graph, it also provides an example of a Freeform SQL report. This report was created using MicroStrategy's Freeform SQL functionality to report on an operational data mart that stores income data. For information on using Freeform SQL, see [Chapter 11, Custom SQL Queries: Freeform SQL and Query Builder](#).

Hi-Lo-Open-Close Stock graph example

Another type of Stock graph that you can create in MicroStrategy is the Hi-Lo-Open-Close graph. In this type of graph, you can view high and low values, as well as opening and closing values, all in a single graph. The following Hi-Lo-Open-Close graph sample provides a look at the quarterly high, low, open, and close inventory numbers of a company. Open values are

displayed as tails on the left side of each column and close values are represented by tails on the right side of each column.



For example, at the beginning (open) of Q4 of 2009, there were 24 copies of the item "500 Best Vacation Home Plans," as shown by the tail on the left side of the column. The fewest items ever available (low) in the company's inventory during that quarter was three and the most items the company had (high) was 61. These high and low values are represented by the top and bottom of the column respectively. Additionally, the tail on the right side of the column shows that at the end of the quarter (close) there were 61 items, which is also equal to the high value for the quarter.

This graph report, named Stock Hi-Lo-Open-Close Chart - Visualizing Inventory Using Hi-Lo-Open-Close Stock Graphs, is available in the MicroStrategy Tutorial project.

Using Stock graphs effectively

- When you create any type of Stock graph, you must include the correct number of attributes and metrics, and arrange the columns of the graph's report in the proper order. This ensures the graph's columns are displayed as shown in the examples above.
 - When you create a Hi-Lo Stock graph in MicroStrategy, you must include at least one attribute and two metrics in the graph report

definition. For example, an attribute that displays the quarter, a maximum stock price metric, and a minimum stock price metric are used to display stock prices. If you do not include a sufficient number of attributes and metrics on the report, you cannot view the report as a Hi-Lo Stock graph.

- When you create a Hi-Lo-Open-Close Stock graph, you must use at least one attribute and four metrics in the graph report definition. For example, a Quarter attribute, maximum inventory metric, minimum inventory metric, opening inventory metric, and closing inventory metric are used in the Hi-Lo-Open-Close graph above. The Item attribute is also included in the Page-by for the graph report, allowing each page to display inventory values for an individual item.
- For stock graphs, arrange the report's metrics in the following order for the series of the graph:

Graph Type	First Metric	Second Metric	Third Metric	Fourth Metric
Hi-Lo Stock graph	High value	Low value	Not displayed on the graph	Not displayed on the graph
Hi-Lo-Open Stock graph	High value	Low value	Open value	Not displayed on the graph
Hi-Lo-Open-Close Stock graph	High value	Low value	Open value	Close value

- If you display the data labels for a stock graph, only a single value is displayed. You can select to display either the high, low, open, or close value as the data label for the stock graph. For Hi-Lo Stock graphs, you should select to display either the high or low data label, as open and close are not applicable. Similarly, for Hi-Lo-Open Stock graphs, you should only select to display either the high, low, or open data label, as close is not applicable.
- If you view a stock graph using MicroStrategy Web, all possible values for the graph can be displayed using tooltips. For steps to create tooltips in a graph, see [Creating tooltips in a graph, page 481](#).
- If the open value tails of a Hi-Lo-Open-Close graph are not displayed when you create the graph in MicroStrategy, right-click the graph background and select **Graph Options**. The Graph Preferences dialog box opens. Expand the **Layout** category, then expand **Hi-Lo options**, and then select the **Show Open** check box. You can also select the **Show Close** check box to display the close values on your graph.



The options to display open and close values are available through drop-down lists in MicroStrategy Web.

- For detailed information about the specific options for Stock graphs in MicroStrategy, see the *MicroStrategy Developer help*.

Histogram

A Histogram is designed to group data frequencies in a Bar graph format. Histograms capture how certain data falls within specific intervals, and are often represented as bell curves. The intervals (for example, profit ranges of \$200,000-\$300,000) appear on one axis. The number of attribute elements (for example, the number of call centers) that fall within each interval are displayed on the other axis. Therefore, a Histogram graph can allow you to answer questions such as:

- How many call centers generated profits within the \$200,000 to \$300,000 range?
- In which profit range did the majority of the call centers' profits exist?

A Histogram can also display the variation in a performance and helps you make decisions about a process, product, or procedure that could be improved after examining the variation.

The data in a Histogram can be skewed to the left or right. If the Histogram shows a long line of short columns on the left side of the Histogram and the high value on the graph appears on the far right, the data is defined as “left-skewed” or “negatively skewed.” If a long line of short columns appears on the right side and the high value on the graph appears on the far left, the data is defined as “right-skewed” or “positively skewed.” Data that is skewed either to the left or right can indicate inconsistent procedures or processes. The decisions you make based on a Histogram can include determining the appropriateness of the direction of the skew.

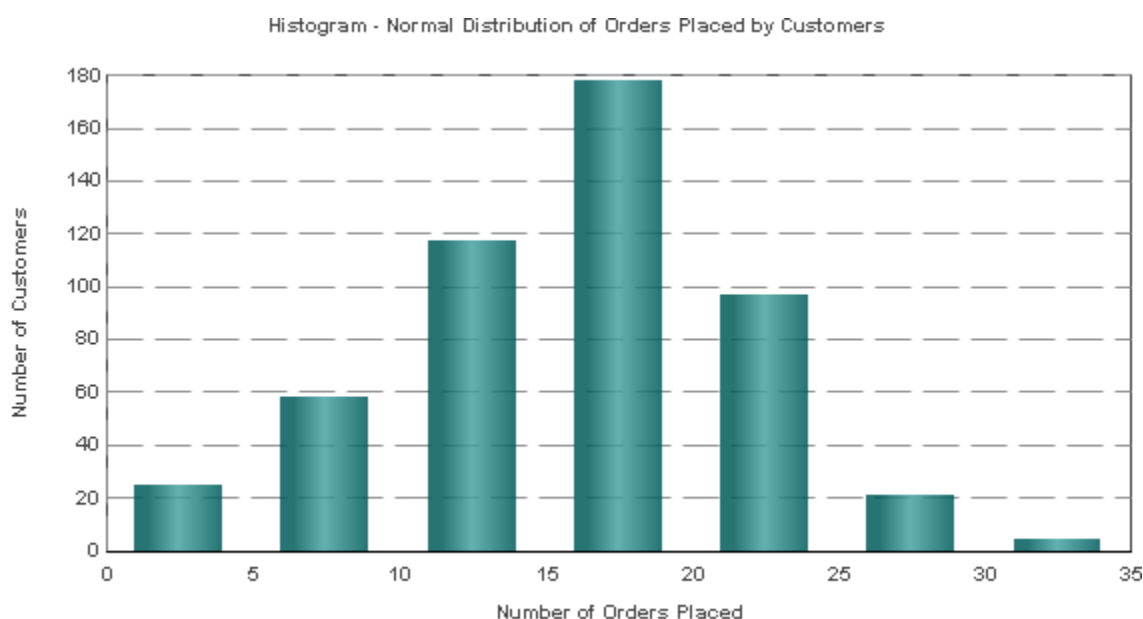
Histogram graph types

You can create the following types of Histogram graphs in MicroStrategy:

- Vertical Histogram graph
- Horizontal Histogram graph

Vertical Histogram graph example

A vertical Histogram graph is shown below. The height of the columns represents the number of customers who placed an order, while the X-axis highlights different ranges of the number of orders placed. For example, the tallest column shows that about 180 customers placed between 15 to 20 orders each.



Note: 500 customers were used to create the histogram.

This graph report, named Histogram - Normal Distribution of Orders Placed by Customers, is available in the MicroStrategy Tutorial project.

Using Histogram graphs effectively

- As shown in the example above, Histogram graphs are best suited for large sets of statistical data. In this case, a sample of 500 customers is used, as indicated by the note below the graph.
- Your Histogram graph may not display all of its data on one page when you first build it. If this occurs, from Graph or Grid Graph View, select the **Graph** menu and then **Preferences**. Within the General category, adjust the **Maximum number of categories** setting to reflect the number of report-level elements on which the Histogram is based. For example, the graph above contains order data for 500 different customers, and Customer is the report-level attribute. In this case, you can adjust the

Maximum number of categories setting to 500 to reflect the 500 different customers represented in the graph.



In MicroStrategy Web, you can set the maximum number of categories by selecting **Graph** from the **Format** menu and adjusting the **Maximum: Categories** setting within the General tab.

- For information about the specific options for Histograms in MicroStrategy, see the *MicroStrategy Developer help*.

3D Surface

A 3D Surface graph is designed to display data points using shading, color, and multiple dimensions. It helps you analyze patterns, comparisons, and trends in data, and is especially useful to interpret the relationships between three different sets of data.

Graphs that contain large data sets are generally good candidates to be displayed as three-dimensional surfaces because 3D Surface graphs provide a third axis on which to visually display data. MicroStrategy allows you to choose several different styles of 3D Surface graphs to accommodate your graphing preferences.

3D Surface graph types

You can create several types of 3D Surface graphs in MicroStrategy, including the following:

- 3D Surface
- 3D Surface with Sides
- 3D Honeycomb Surface

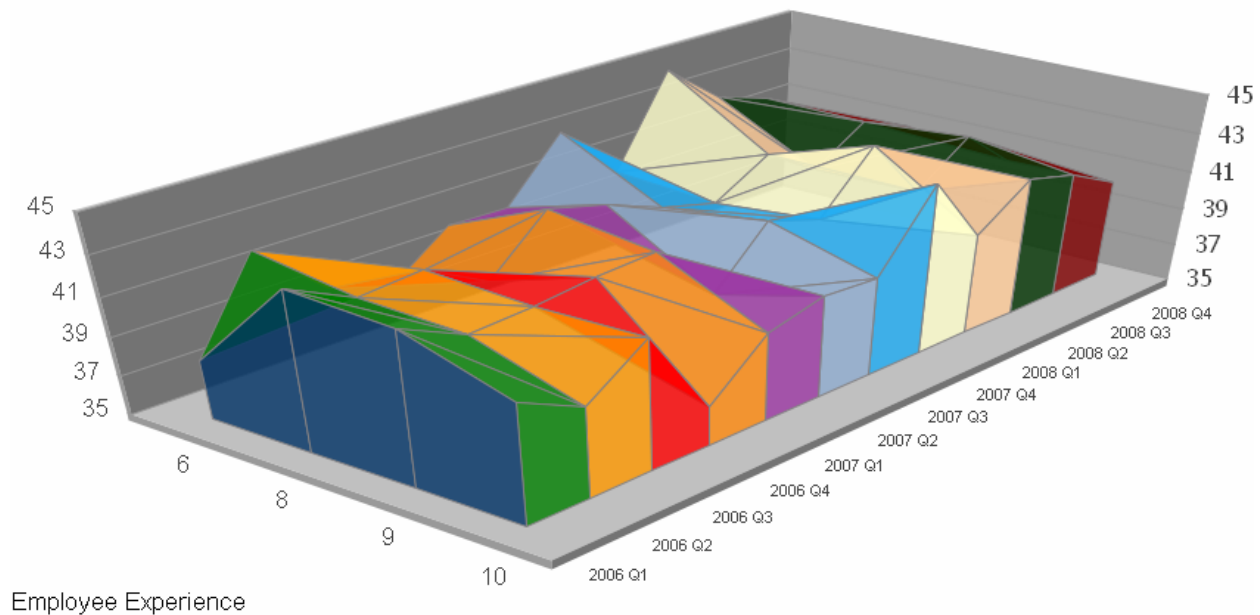
3D Surface Graph with Sides example

One of the types of 3D Surface graphs you can create in MicroStrategy is a 3D Surface Graph with Sides. The following graph highlights the type of data you can use to create a 3D Surface Graph with Sides. It also shows how you can analyze the data on the graph.

The example highlights the effect that time and employee tenure have on the average revenue contribution of each employee. Notice that the Y-axis, which represents average revenue per employee, appears on both the left and right side of the graph because of the graph's dimensionality.

The bottom-left side of the graph represents the number of years an employee worked for the company and the bottom-right side of the graph represents different quarters. Notice that the amount of generated revenue can differ depending on employee experience. For example, in 2006 Q1 (represented by the blue side on the far left of the graph), revenue contribution was lowest for employees with six years of experience. Revenue contribution was also relatively low for employees with ten years of experience. During the same quarter, employees with eight and nine years of experience contributed the most to revenue. Similar results are also apparent throughout future quarters. However, some quarters (2007 Q4 and 2008 Q2) show increases in revenue contribution by employees with six or ten years of experience.

3D Surface Graph - How do Tenure and Time Affect Average Employee Contribution to Revenue?



Using 3D Surface graphs effectively

- To format the color of each category in a 3D Surface graph in MicroStrategy Developer, right-click the graph's background and select **Graph Options**. Within the **Display** category, from the **Color Mode** drop-down list, select **Color by Categories**. It is also helpful to set the risers to be colored according to their height. To do this, from the **Color**

Mode drop-down list, select **Exact Color By Height**. Note that you can make only one selection at a time from the Color Mode drop-down list.



In MicroStrategy Web, you can access graph formatting options by selecting **Graph** on the **Format** menu. However, some graph formatting options, such as assigning the color of columns by category or height, are not available in Web.

- For information about the specific options for 3D Surface graphs in MicroStrategy, see the *MicroStrategy Developer help*.

Gauge

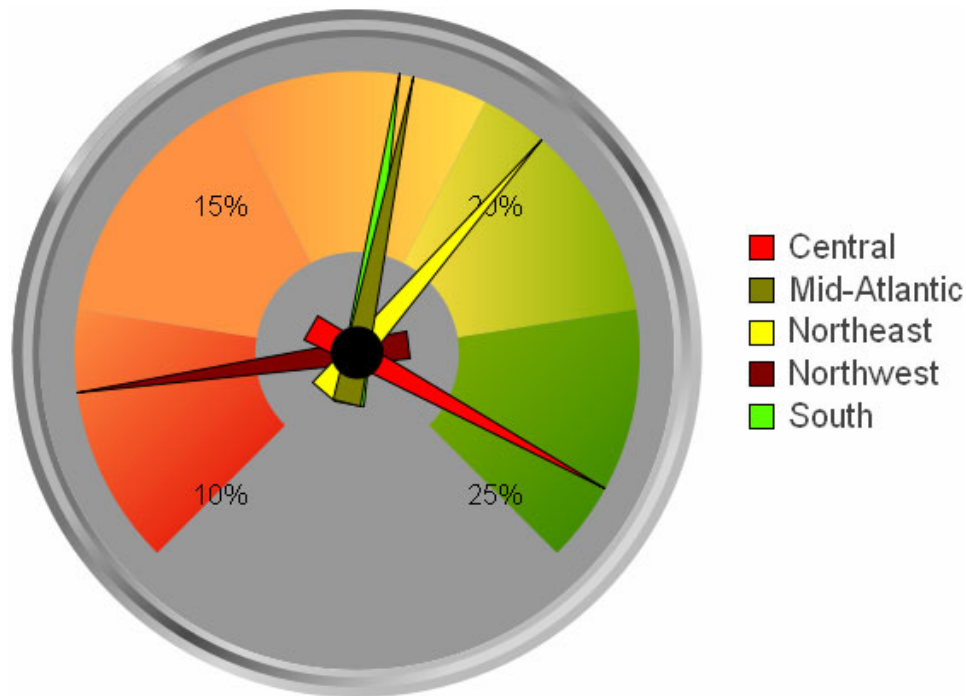
A Gauge graph is designed to display a specific data set using a semicircular dial with indicators. The graph's indicators are situated within the dial's range to depict whether the monitored data is within defined limits.

There is only one type of Gauge graph available in MicroStrategy.

Gauge graph example

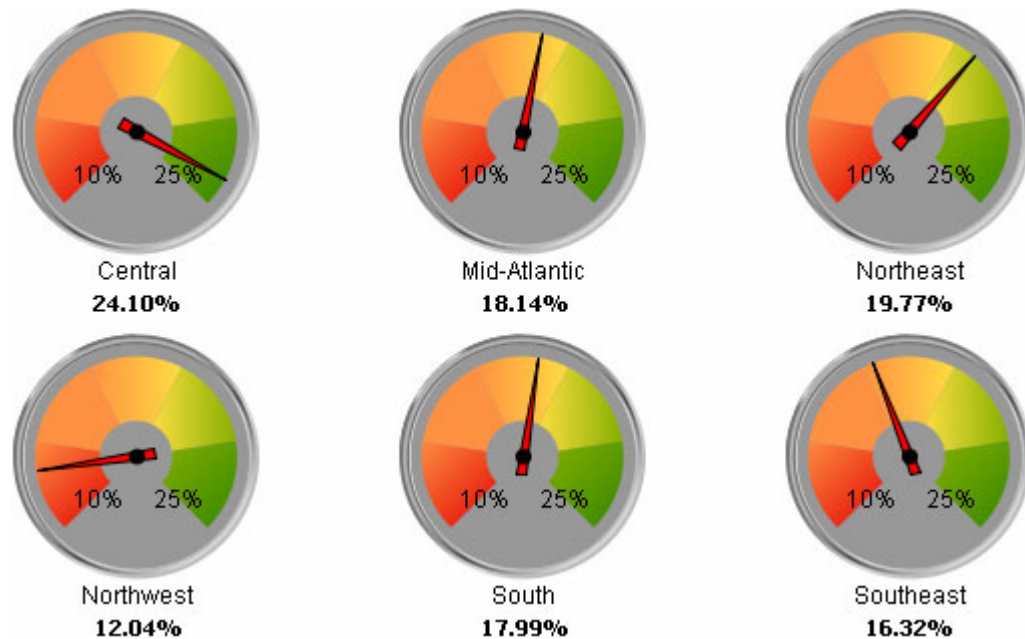
In this example, annual revenue data for specific regions is displayed in the following Gauge graph. The Gauge graph reveals the percentage by which certain regions grew their revenue from last year to the current year. Notice

that the Central region achieved the largest revenue growth, as indicated by the needle closest to the maximum of the gauge (25%).



You can display the same information in a different way by showing each region's revenue data in side-by-side Gauge graphs, as shown below. To do this in MicroStrategy Developer, move the Region attribute from the series of the graph report to the categories. The Gauge graph has also been defined to display three gauges on each row.

This graphing approach allows you to quickly determine which region was most successful when compared to the previous year. The Gauge graph below depicts this alternative way of using gauges to analyze revenue data.



This graph report is a modified version of the graph report named Regional % Revenue Growth (Gauge), which is available in the MicroStrategy Tutorial project.

Using Gauge graphs effectively

- To display one gauge that displays data for every item, place both the metric and attribute in the columns. For example, on the report template for the first graph in this section, the Region attribute is placed on the report's columns so all data for the regions is displayed in one gauge.
- To display multiple gauges, place the attribute on the rows and the metric on the column. For example, on the report template for the second graph in this section, the Region attribute is placed on the report's rows so multiple gauges are displayed at the same time in the graph.

To adjust the number of gauges that are shown on one page, right-click the graph's background and select **Graph Options**. Within the **General** category, adjust the **Gauges Per Row** setting to specify the number of gauges to appear on one row of the page. For example, for the graph

above, specifying three as the number of gauges per row results in three gauges displaying on each row.



To specify how many gauges appear on one row in MicroStrategy Web, from the **Format** menu select **Graph**. Then, modify the Maximum Categories setting to match the desired number of graphs.

- You can determine the range of numbers that is displayed for each quality band in the gauge. To do this, from the **Graph** menu select **Graph Options**. Within the **Display** category, adjust the minimum and maximum values for each band.
- You can adjust the thickness of the gauge dial. To do this, from the **Graph** menu select **Graph Options**. Within the **General** category adjust the **Range Thickness** setting.
- For information about the specific options for Gauge graphs in MicroStrategy, see the *MicroStrategy Developer help*.

Funnel

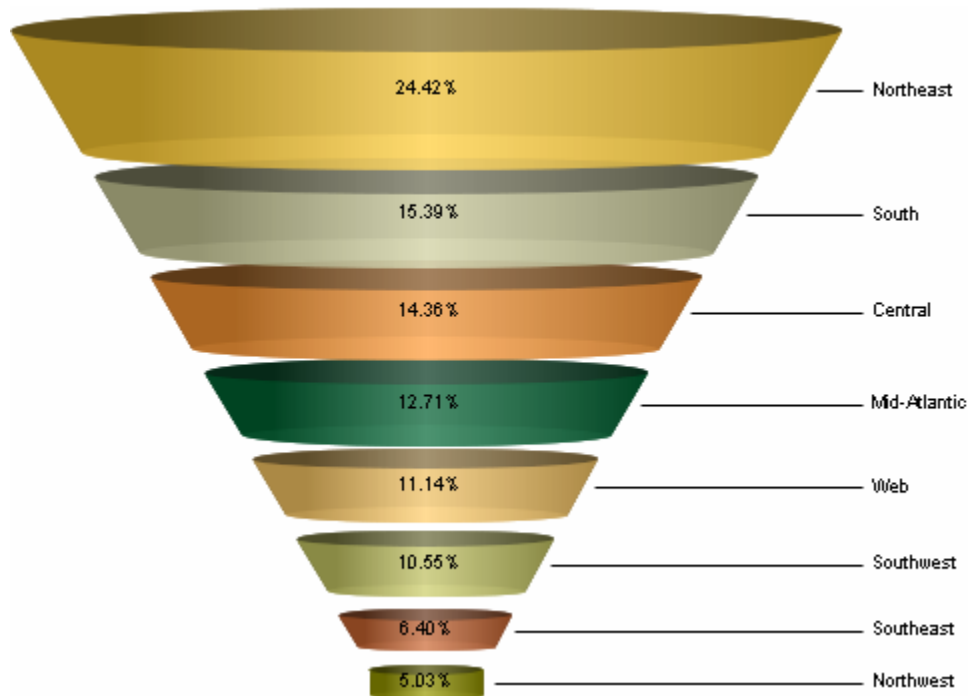
A Funnel graph provides another visually distinctive way of depicting report data. Funnel graphs are generally designed for conducting pipeline analyses for sales forecasts. They are also useful in illustrating the percent contribution of sales data, as they usually display data that adds up to 100%.

There is only one type of Funnel graph available in MicroStrategy.

Funnel graph example

The following Funnel graph below provides information about the sales contribution of different regions. Each funnel slice, with its own color, represents a different region, such as Mid-Atlantic. The size and location of each slice depicts how a region's sales contribution compares to the other

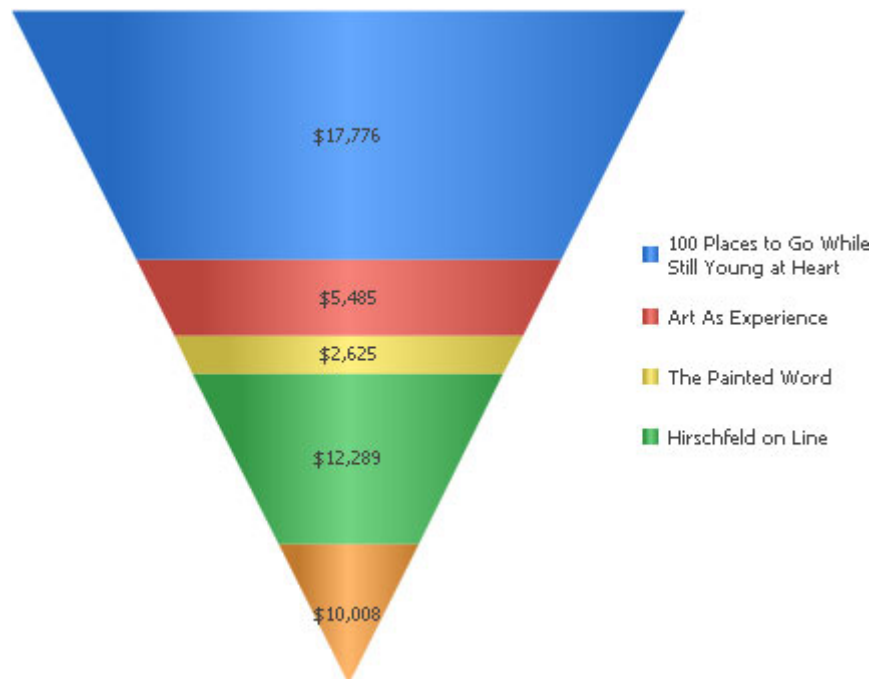
regions' figures. For example, the graph shows that the Northeast region contributed 24.42% of the total sales, the most of any region.



This graph report, named Funnel Graph - Sales Contribution By Region, is available in the MicroStrategy Tutorial project.

As demonstrated in the following graph, you can also use Funnel graphs to analyze simple sales and profit figures, in the same way you might use a Bar

graph. The graph below displays the profit figures of five different books and shows how each book's profits compare with the others.



Using Funnel graphs effectively

- How a report is sorted determines what data is displayed at the top of the funnel. For example, in the first graph in this section, you can view the lowest sales contribution percentages at the top of the funnel instead of at the bottom by sorting the Percent to Total Over Rows (Revenue) metric column in ascending order.
- If your Funnel graph does not display all of its data on one page when you first build it, with the report in Graph or Grid Graph View, from the **Graph** menu, select **Preferences**. On the **General** category, adjust the **Maximum number of series** setting to reflect the number of report-level attribute elements that the Funnel graph is based on. For example, suppose a graph contains profit data for 20 different items and Item is the report-level attribute. In this case, you can adjust the **Maximum number of series** setting to 20 to display the 20 different items represented in the graph on one page.

In MicroStrategy Web, you can set the maximum number of series. From the **Format** menu, select **Graph**. Within the **General** category, in the

Maximum area, in the **Series** field, type the maximum number of series to display for the graph.



For information about the specific options for Funnel graphs in MicroStrategy, see the *MicroStrategy Developer help*.

Pareto chart

A Pareto chart is designed to help identify the cause of a quality problem or loss. It includes a Histogram that often shows how often a specific problem is occurring or the different types of problems that are occurring. In general, Pareto charts allow you to display the specific areas in which improvement or investigation is necessary.

Pareto chart graph types

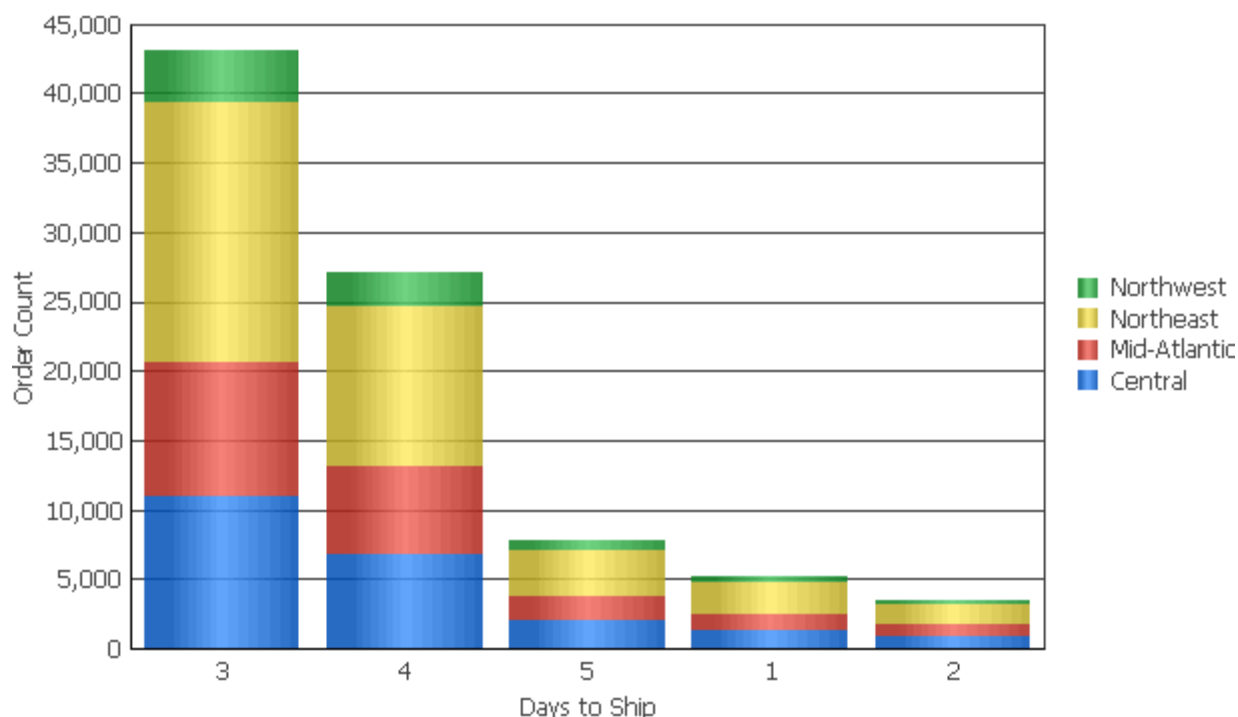
You can create the following types of Pareto charts in MicroStrategy:

- Pareto graph
- Pareto Percent graph

Pareto chart example

The following Pareto chart and its associated grid report reveal data about how long it took each region to ship its orders and the number of orders each region shipped. The Y-axis represents the number of orders shipped and the X-axis displays the number of days it took for an order to ship. The column bands each represent a different region. It is beneficial to view this

information in a Pareto chart because it allows you to easily analyze and compare the shipping trends of different regions.



	Region	Central	Mid-Atlantic	Northeast	Northwest	Total
	Metrics	Order Count	Order Count	Order Count	Order Count	Order Count
Days to Ship						
3		10,921	9,591	18,642	3,755	42,909
4		6,823	6,226	11,643	2,406	27,098
5		2,010	1,783	3,340	708	7,841
1		1,373	1,148	2,239	410	5,170
2		910	824	1,591	305	3,630

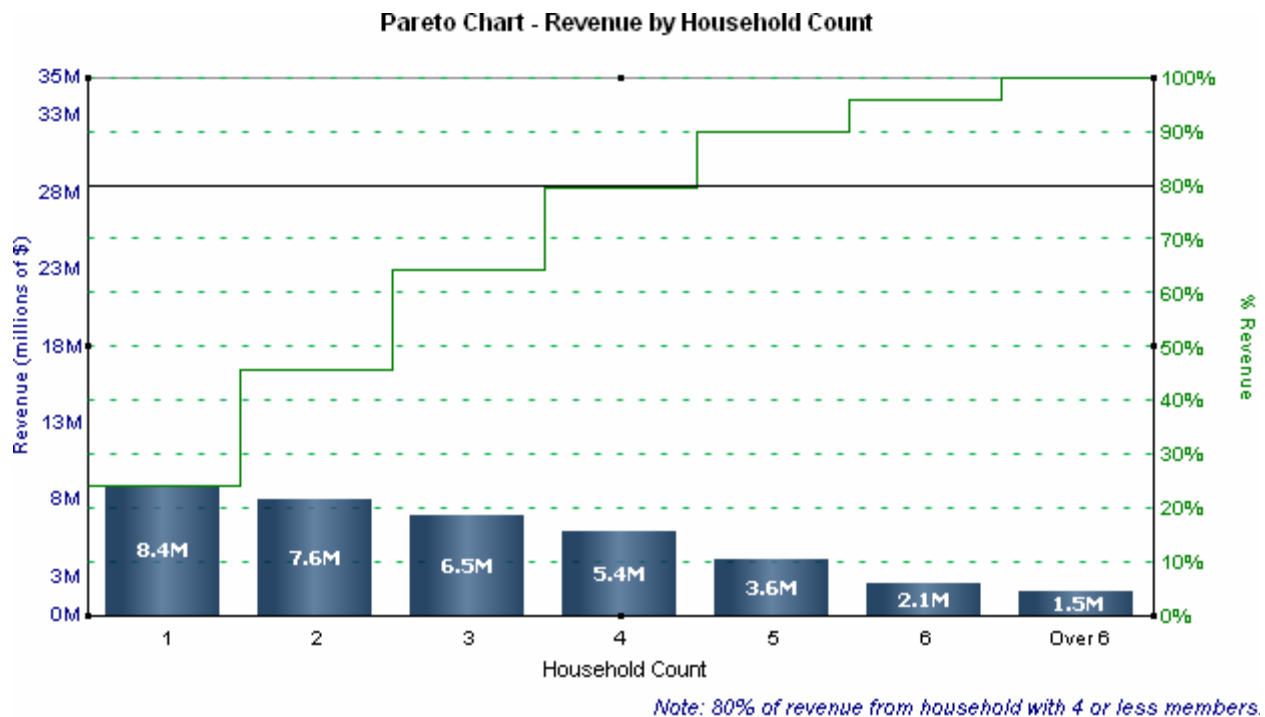
Notice that the majority of orders were shipped in three days, as indicated by the first and tallest column in the graph. Within the same column you can also see that the Northeast region shipped the most orders. Meanwhile, the smallest column on the right reveals that fewer than 5,000 orders were shipped in two days for any region displayed on the graph.

While this graph looks similar to a stacked graph (see [Vertical Bar: Stacked graph example, page 386](#)) it should be noted that the values are represented differently. In a stacked graph, each item is displayed with a height equal to that item's value, with the item displayed at the top having the highest value. This gives a representation that the items are stacked, with the item with the smallest value at the front and the item with the largest value at the back. Therefore, a stacked graph is a good option for viewing the totals of each item in relation to the other items.

In a Pareto chart such as the one shown above, all the items are displayed on top of each other so that the height of the column represents the total of all items. Each item's relative size within a column represents how much of that total comes from that particular item. Therefore, a Pareto chart is a good option for viewing the total value of a group of items, along with each item's relative contribution to that total.

Pareto Percent example

The Pareto chart below is an example of a Pareto Percent graph, another type of Pareto chart you can create in MicroStrategy. The graph reveals the amount and percentage of revenue generated from households of different sizes. In this Pareto Percent graph, the X-axis represents the number occupants in the household. The Y-axis represents the total revenue generated for each household.



The Line graph above the columns represents the cumulative contribution to revenue, which is tracked on the Y2-axis on the right side of the graph. For example, \$8.4 million in revenue was generated by households with a single occupant, which represents about 24% of the total revenue.

The difference between the other increments on the line shows the percent contribution of each household count to the total revenue. For example, \$7.6 million in revenue was generated by households with two occupants, which represents about 21% of the total revenue. You can also obtain this 21% value

by subtracting the 24% above the household count of one, from the 45% above the household count of two. In this way, the Line graph provides another way of viewing the graph information displayed in the Pareto Percent graph's columns.

This graph also includes a line at the 80% of revenue mark. This line provides the immediate recognition that 80 percent of the total revenue was received from households with four or fewer occupants.

This graph report, named Pareto Chart - Revenue by Household Count, is available in the MicroStrategy Tutorial project.

Using Pareto charts effectively

- To add data labels to the columns of your Pareto chart, right-click the graph background and select **Graph Options**. Expand the **Series** category, select the **Labels and Values** category, and then select the **Show Data Labels** check box. You can use the drop-down lists that become enabled to customize the data labels.

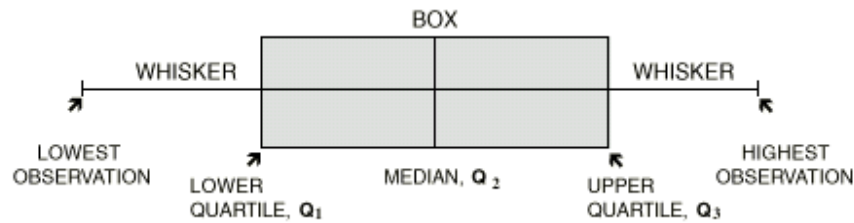


- In MicroStrategy Web, you can make data labels visible using the Format: Graph dialog box. To open the Format: Graph dialog box, from the **Format** menu select **Graph**. In the **Format** category, from the left-most drop-down list, select **Series Values**. To display data labels on the graph, select the **Show** check box.
- For information about the specific options for Pareto charts in MicroStrategy, see the *MicroStrategy Developer help*.

Boxplot

A Boxplot (sometimes called a box and whisker plot) is a graph type designed to compare similar distributions at a glance. The goal of Boxplot graphs is to make the center, spread, and overall range of values immediately apparent. The Boxplot is useful for isolating the extreme values and identifying the range of middle values in your data. The box shows the median values of a variable, while the whiskers stretch to the greatest and lowest values of that

variable. Boxplots can also help you identify symmetrical or skewed distributions among values.



This type of graph is most beneficial when you are comparing one variable for several different groups, for comparing similar distributions (because the center, spread, and overall range are immediately apparent), and for detecting symmetrical or skewed distributions.

The image above provides an overview of the values and statistics depicted in Boxplot graphs, and is not an actual representation of a MicroStrategy Boxplot graph. A Boxplot consists of five summary statistics, each of which is described below. As shown in the image, a box is drawn around the quartile values, and the whiskers extend from each quartile to the extreme data points.

The summary statistics within box and whisker plots include the following:

- **Median:** The middle of the data when it is arranged in order from least to greatest.
- **Lower quartile or 25th percentile:** The median of the lower half of the data.
- **Upper quartile or 75th percentile:** The median of the upper half of the data.
- **Minimum value:** The smallest observed value. In MicroStrategy graphs, this is depicted as a whisker on the bottom of a column.
- **Maximum value:** The largest observed value. In MicroStrategy graphs, this is depicted as a whisker on the top of a column.

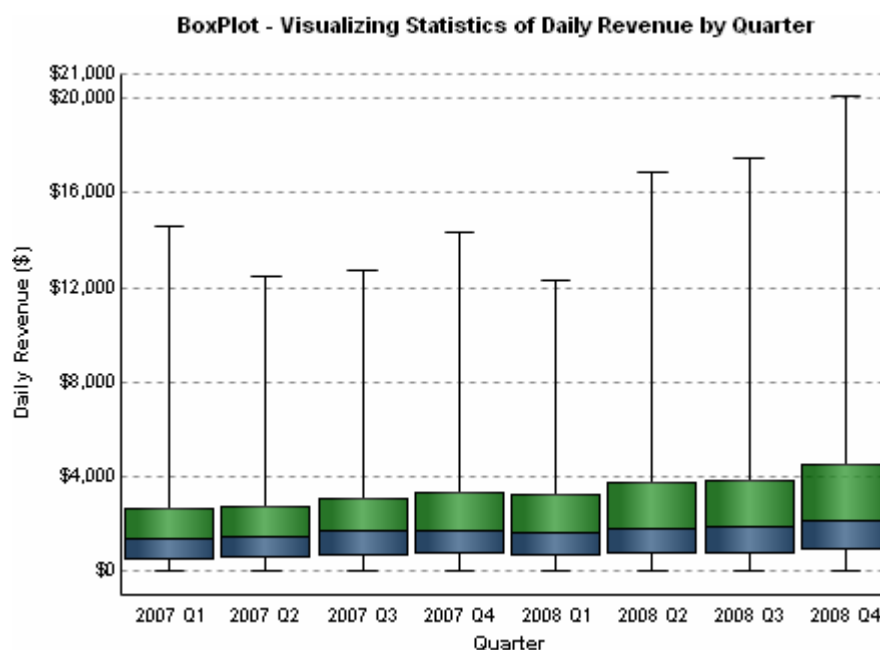


Boxplot graphs only work well when there is enough data to provide the statistics described above. For example, you use a Boxplot graph for data that covers certain years or quarters to see the different time periods plotted. If you only have one or two years or quarters in your data, the Boxplot graph does not display all the expected statistics because it cannot define quartiles and medians from only one or two values.

Boxplot graph example

The following Boxplot graph and its corresponding grid report provide data about the amount of revenue generated each day during a given quarter. Each column is split into two sections. The top section represents the upper quartile or the 75th percentile, and the bottom section represents the lower quartile or the 25th percentile. The whiskers extending from the top and bottom of each column indicate the highest and lowest daily revenue values respectively.

For example, during Q4 of 2007, daily revenue reached a low of about \$12, as shown by the whisker below the column. Daily revenue that quarter did, however, reach a high of over \$14,000, as evidenced by the whisker at the top of the column. The median revenue value for the quarter, \$1,758, is shown as a horizontal line that splits the column. The upper and lower quartiles are shown above and below the median line respectively.



	Metrics	Minimum Daily Revenue	1st Quartile	Median Daily Revenue	3rd Quartile	Maximum Daily Revenue
Quarter						
2007 Q1		\$12	\$565	\$1,368	\$2,656	\$14,573
2007 Q2		\$10	\$646	\$1,438	\$2,741	\$12,450
2007 Q3		\$14	\$731	\$1,701	\$3,083	\$12,721
2007 Q4		\$12	\$788	\$1,758	\$3,314	\$14,355
2008 Q1		\$9	\$723	\$1,647	\$3,222	\$12,288
2008 Q2		\$10	\$784	\$1,814	\$3,793	\$16,878
2008 Q3		\$8	\$814	\$1,863	\$3,815	\$17,497
2008 Q4		\$9	\$977	\$2,154	\$4,545	\$20,132

This graph report is a modified version of the graph report named BoxPlot - Visualizing Statistics of Daily Revenue by Quarter, which is available in the MicroStrategy Tutorial project.

Using Boxplot graphs effectively

- When you create a Boxplot graph you must include the correct number of metrics for the graph to resemble the example above. To view the quartile sections and median line within the graph columns, as well as the high-low whiskers, five metrics must appear on your report template in the following order:
 - Minimum value metric (for example, the Minimum Daily Revenue metric in the graph above).
 - Low value metric (for example, the 1st Quartile metric).
 - Base or median value metric (for example, the Median Daily Revenue metric).
 - High value metric (for example, the 3rd Quartile metric).
 - Maximum value metric (for example, the Maximum Daily Revenue metric).
- For details about building the types of metrics used in the graph above, open the Boxplot graph in the Graph Styles folder in MicroStrategy Tutorial and view each metric's definition.
- For information about the specific options for Boxplot graphs in MicroStrategy, see the *MicroStrategy Developer help*.

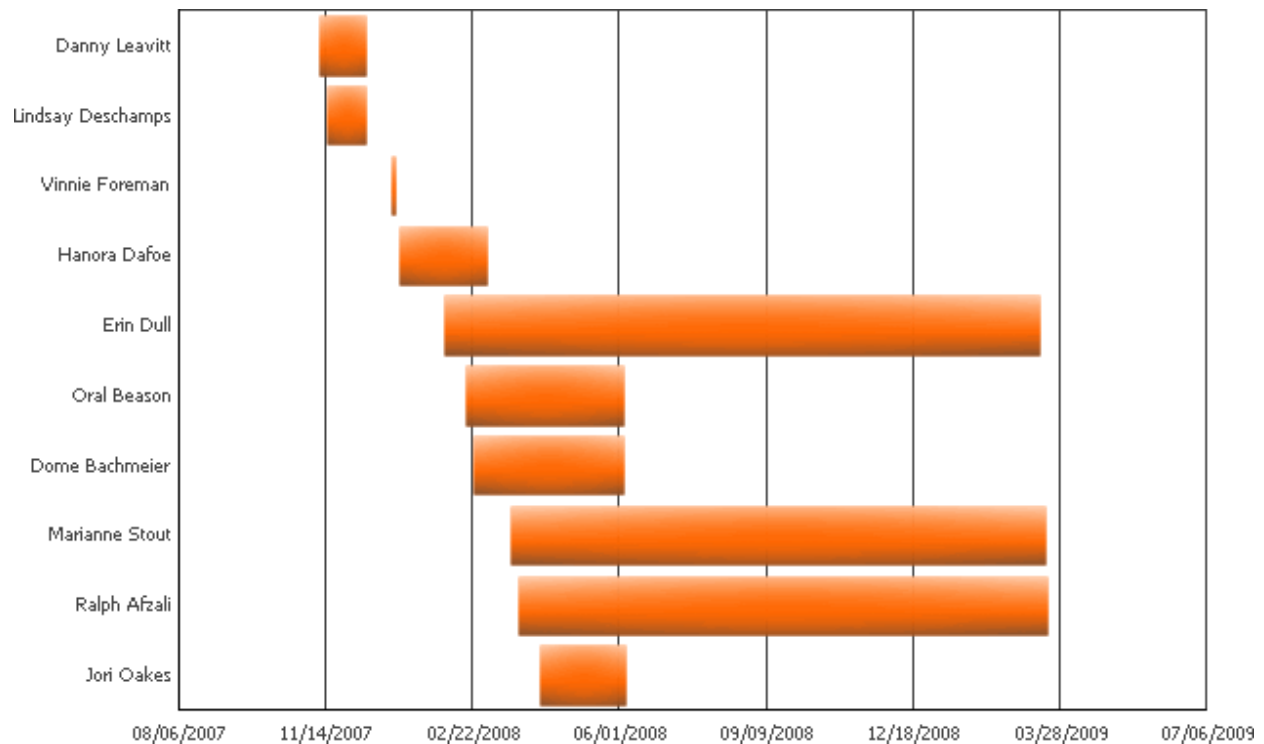
Gantt chart

Gantt chart (also referred to as project time lines) are Bar graphs that help plan and monitor project development or resource allocation on a horizontal time scale.

A Gantt chart is essentially a horizontal bar chart. Attributes are displayed along the X-axis and metrics are displayed along the Y-axis. The Gantt chart provides a graphical illustration of a schedule that can help you plan, coordinate, and track specific tasks in a project. The data analyzed in a Gantt chart has a defined starting and ending value; for example, Project A begins 4/15/09 and lasts for a total of 40 days. The start time and duration are what allows you to display information as a Gantt chart.

Gantt chart example

Gantt charts are useful when analyzing human resources information. For example, the following Gantt chart depicts how long certain employees have been working for a company. The beginning of each horizontal bar represents a specific employee's hire date and the end of each bar represents the time at which an employee left the company. The overlap of the bars helps to visualize the workforce in the company during various periods of time. The report shown below displays the employee tenure for a subset of employees in a company.



This information can potentially help a project manager in a number of ways. Given the information obtained in a Gantt chart such as this, he or she can better manage his or her resources to accommodate for the lack of personnel, as well as plan future hiring and project distribution.

Gantt charts require conceptual information from attributes, as well as two metrics:

- The first metric defines a starting point for the graph. This can be a particular date or a starting value. For the graph shown above, a Hire Date metric that provides the day an employee was hired serves as the starting point for the employee's employment with the company.

- The second metric defines the duration of the event that you are interested in, and it must match the time interval represented by the starting point. For the graph shown above, the duration of the Gantt chart is determined using a Tenure metric that provides the number of days an employee was hired with the company.

To create the Gantt charts displayed above, a report with the following information shown below was created.

Employee	Metrics	Hire Date	Tenure	Leave Date
Danny Leavitt		11/10/2007	32	12/12/2007
Lindsay Deschamps		11/15/2007	27	12/12/2007
Vinnie Foreman		12/30/2007	2	01/01/2008
Hanora Dafoe		01/04/2008	59	03/03/2008
Erin Dull		02/03/2008	406	03/15/2009
Oral Beason		02/18/2008	107	06/04/2008
Dome Bachmeier		02/23/2008	102	06/04/2008
Marianne Stout		03/20/2008	363	03/18/2009
Ralph Afzali		03/25/2008	360	03/20/2009
Jori Oakes		04/09/2008	58	06/06/2008

The Hire Date metric defines the starting point for each bar in the Gantt chart. The Tenure metric defines the end point for each bar in the Gantt chart. This metric is calculated by determining the number of days between the Hire Date and the Leave Date. While the Leave Date metric is not part of the display of the Gantt chart, it is shown here to help understand how the data for the Gantt chart was created.

Listed below is some additional information that you can use to re-create the Gantt chart shown above, or to create a similar Gantt chart using your data:

- This Gantt chart was created using the Human Resources Analysis Module, which is a part of the MicroStrategy Analytics Modules.
- The metrics Hire Date and Leave Date were created using the Hire Date and Leave Date facts, respectively. These facts use data that is provided in a date format, and displayed in mm/dd/yyyy format.
- The Tenure metric was created as a compound metric with the formula $([\text{Leave Date}] - [\text{Hire Date}])$. This metric returns integer values for the number of days that an employee was with the company.
- When date data is initially displayed on a graph, it may not be displayed in a date format by default. To display the time intervals correctly on the axis, you must modify the number formatting of the axis labels with a date format that matches your date data. For information on defining the

number formatting for graphs, see [Manually defining the formatting of numeric values, page 472](#).

- The minimum value for the graph axis that displays time intervals was increased. The default minimum value for a graph axis is zero, but since dates are considered as numbers when creating the graph, this minimum must be increased to improve the display of the graph. To determine the numeric values associated with the time intervals on the axis, you can modify the number formatting of the axis labels to use the general number format. Once you determine the numeric value, you can define a relevant minimum value for the graph axis, and then switch the axis labels back to the proper date format for the date data. For information on defining the minimum or maximum value of a graph, see [Scaling a graph, page 489](#).

Using Gantt charts effectively

When you create a Gantt chart, consider the following:

- At least two metrics must be displayed in the columns:
 - One metric serves as the starting data point.
 - One metric serves as the length, or duration, from the starting point.



If you use two metrics that both return dates such as a start date and an end date, the Gantt chart is not displayed correctly.

- A pair of metrics defines start and length values which are translated as a horizontal bar in the graph. These metrics must use the same time interval to accurately display data on a Gantt chart. For example, if your starting point is a specific day such as 07/25/2010, the duration must be provided in terms of days. Therefore, a project that started on 07/25/2010 and ends on 07/31/2010 must have its duration provided as seven days. If, however, project start and end dates are based on a quarterly schedule (for example, starting in quarter one of 2009 and ending in quarter one of 2010) the duration must be provided as the number of quarters that the project was active for.
- Multiple metric pairs can be added to the columns to represent more start-length pairs. If you add an odd number of metrics to the report, the last metric on the report is not displayed on the graph.
- If you use dates to define the time period for the Gantt chart, you may need to change the minimum value of the graph's axis to better display data. For information on defining the minimum or maximum value of a graph, see [Scaling a graph, page 489](#).

- For information about the specific options for Gantt charts in MicroStrategy, see the *MicroStrategy Developer help*.

Formatting a graph report

Formatting a graph report enables you to specify font, line, and fill characteristics for text, lines, and areas in a graph. This allows you to highlight important information, and to make your graph reports easier to read and understand.

You can format several objects within your graph, including the following:

- Title
- Legend
- Subtitle
- Footnote
- Background
- Frame
- X-axis Label
- Y1-axis Label
- Y2-axis Label

You can format the above graph elements by using the **Formatting** option in the Graph menu in Grid Graph or Graph View.



The Title, Subtitle, Footnote, X-axis Label, Y1-axis Label, and Y2-axis Label options are enabled only if the option is used in the graph. For example, you can select the option Title only if the graph includes a title.

You can format the following elements of a graph by selecting an item on the graph, right-clicking it, and then choosing **Format <object name>** from the right-click menu:

- Labels
- Headers
- Risers (bar, area, and so on)

- Pie slices
- 3D objects

Controlling how a graph report appears

Report designers can control the look of a graph report at any time, even after it has been analyzed and potentially reformatted by users.

When thinking about how to format a graph report, you must first consider the various layers of formatting that exist within the report. Each report in MicroStrategy contains several layers of formatting, including the formatting of metrics on the grid report, axes, subtotals, column banding, thresholds, and more. Report formatting is contained in these different layers, allowing you to control different aspects of a graph's formatting and ensure that the important information is still emphasized. For example, as explained in [Formatting numeric values on a graph, page 468](#), a metric's formatting in a grid report plays a role in how the metric is formatted in a graph. The metric's formatting is one report formatting layer you can modify as you consider different ways to format your graph.

For specific details about the report formatting layers, see [Understanding how formatting impacts report display, page 317](#).

Formatting a graph using colors, shading, and effects

You can change the colors and overall look and feel of specific objects, such as series, on your graph. For example, you can format individual bars in your Bar graph or pie slices in your Pie chart using thresholds, as described in [Displaying thresholds on graph reports, page 446](#). The graph legend or axes can be formatted as well.

You can format a graph report by adjusting colors, and applying a variety of graphical effects, including gradients, transparency, patterns, pictures, textures, and bevel effects. You can also apply a set of color and shading schemes to all the objects on a graph report.

Formatting colors, shading, and effects in MicroStrategy Developer

You can change the colors, shading, and other effects on your graph report in a number of ways. For example, you can do the following:

- Select a color/shading scheme from the **Color Styles** drop-down menu on the Graph toolbar. When you select a color/shading scheme for a graph report, the colors of the graph are changed, and shading and other effects are applied as well. For these steps, follow the procedure *To change the colors, shading, and other effects on a graph report in MicroStrategy Developer* below.
- Select from several color palettes in the **Color Palette** drop-down menu on the Graph toolbar. When you select a color palette for a graph report, the colors of the graph are changed, but shading and other effects are not applied. For these steps, follow the procedure *To change only the colors of a graph report in MicroStrategy Developer* below.
- Manually format the color of individual objects and apply effects such as gradients to them. For example, you can adjust the color of and apply a texture to the bars of a Bar graph. For more information, see *Manually formatting graph series and objects, page 451*.
- Define metric-specific graph colors that are used when a metric is displayed as a series in a graph. Metric-specific graph colors override any colors defined by the color palettes and color styles selected for the graph, as well as any manual color formatting selected for the series of the graph. For steps to define metric-specific graph colors, see *Defining a graph color for metrics in MicroStrategy Developer, page 437*.
- Define graph colors for individual graph elements such as bars in your Bar graph or pie slices in your Pie chart using thresholds, as described in *Displaying thresholds on graph reports, page 446*.

The procedure below provides steps to change the colors, but not the shading and other effects, on your graph report.

To change only the colors of a graph report in MicroStrategy Developer

- 1 View a report in Graph or Grid Graph View. To do so, from the **View** menu, select **Graph View** or **Grid Graph View**.

- 2 On the Graph toolbar, click the arrow next to the **Color Palettes**  icon. A list of available color palettes is displayed.



If the Graph toolbar is not displayed, from the **View** menu, select **Toolbars**, then **Graph**.

- 3 Select a color palette to apply its color scheme to the graph report. The graph report is updated with the new colors.

The steps below show you how to change colors, shading, and other effects on your graph report.

To change the colors, shading, and other effects on a graph report in MicroStrategy Developer

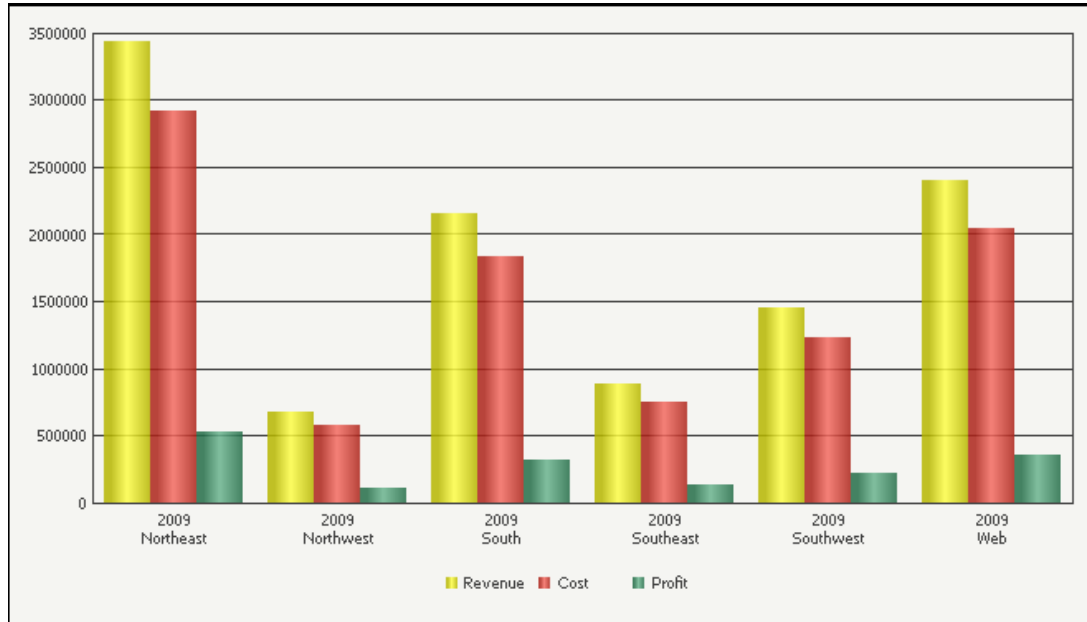
- 1 View a report in Graph or Grid Graph View. To do so, from the **View** menu, select **Graph View** or **Grid Graph View**.
- 2 On the Graph toolbar, click the arrow next to the **Color Styles**  icon. A menu of available color/shading schemes is displayed.

 If the Graph toolbar is not displayed, from the **View** menu, select **Toolbars**, then **Graph**.
- 3 Select a color/shading scheme to apply to the graph report. The graph report is updated with the new color and shading scheme.

Defining a graph color for metrics in MicroStrategy Developer

You can define the color used for a metric when it is displayed as a series in a graph. The graph color that you define for a metric overrides any default color schemes for the graph report. This lets you maintain any metric-specific color schemes that are utilized throughout a MicroStrategy

project. For example, you can define a Profit metric to always display in green on graph reports.

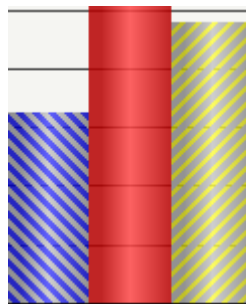


To allow colors for metrics to be inherited in graph reports, you must take the following actions:

- Define the colors to use for a metric when it is displayed as a series in a graph. You can define this color formatting for the metric itself, or for the metric on a specific graph report:
 - You can define the graph color for a metric, and this color is then used as the default graph color for the metric in all graph reports. Steps on how to complete this configuration are described in [To define the default graph color for a metric in MicroStrategy Developer, page 439](#).
 - You can define the graph color for a metric on a specific graph report. This overrides any default metric graph color options. Steps on how to complete this configuration are described in [To define the graph color for a metric in a graph report, page 440](#).
- Enable the graph report to inherit a metric's graph color. Steps on how to complete this configuration are described in [To enable or disable a graph report to inherit a metric's graph color in MicroStrategy Developer, page 441](#).

To define the default graph color for a metric in MicroStrategy Developer

- 1 Using MicroStrategy Developer, log in to a project. You must log in with an account that has privileges to modify metrics.
- 2 Browse to a metric, right-click the metric, and then select **Edit**. The Metric Editor opens.
- 3 From the **Tools** menu, point to **Formatting**, and then select **Headers**. The Format Cells dialog box opens.
- 4 From the **Chart** tab, in the **Series Color** drop-down list, select one of the following options:
 - **<default>**: Allows the color for the metric to be defined on the graph report. For steps on how to define the graph color for a metric in a graph report, see [To define the graph color for a metric in a graph report, page 440](#).
 - **Solid**: Displays a single color for the metric when the metric is displayed as a series in a graph report. You can select the color from the **Fill color** drop-down list.
 - **Gradient**: Displays a blend of two colors in a gradual color change for the metric when the metric is displayed as a series in a graph report. You can select the two colors for the gradient from the **Color 1** and **Color 2** drop-down lists, and select the angle for the gradient.
 - **Pattern**: Applies a pattern to the object's background. Patterns can help to distinguish between the risers of different series in a graph without having to rely solely on color. For example, the three bar risers shown below can be distinguished by color as well as by the pattern that they use.



- **From Graph**: Allows the color for the metric to be defined on the graph report. When defining the default graph color for the metric at the metric level, this From Graph option is the same as using the

<default> option. For steps on how to define the graph color for a metric in a graph report, see [To define the graph color for a metric in a graph report](#) below.

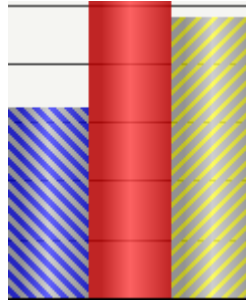
- 5 Click **OK** to save your changes and close the Format Cells dialog box.
- 6 Click **Save and Close** to save your changes and close the Metric Editor.

You can repeat these steps for any metrics that require a specific graph color. Each metric should use a color that can be easily distinguished from the colors used for other metrics. This ensures that graph report results are easy to decipher based on color.

To define the graph color for a metric in a graph report

- 1 Using MicroStrategy Developer, log in to a project and navigate to a graph report.
- 2 Right-click the graph report and select **Run**. The Report Editor opens, displaying the report as a graph.
- 3 From the **View** menu, select **Grid View**. The report is displayed as a grid.
- 4 In the report results, right-click the metric, point to **Formatting**, and select **Metric Name Headers**. The *Metric Name* is the name of the metric. The Format Cells dialog box opens.
- 5 From the **Chart** tab, in the **Series Color** drop-down list, select one of the following options:
 - **<default>**: Displays the default graph color that was defined for the metric using the Metric Editor. For steps on how to define the graph color for a metric using the Metric Editor, see [To define the default graph color for a metric in MicroStrategy Developer, page 439](#).
 - **Solid**: Displays a single color for the metric when the metric is displayed as a series in a graph report. You can select the color from the **Fill color** drop-down list.
 - **Gradient**: Displays a blend of two colors in a gradual color change for the metric when the metric is displayed as a series in a graph report. You can select the two colors for the gradient from the **Color 1** and **Color 2** drop-down lists, and select the angle for the gradient.

- **Pattern:** Applies a pattern to the object's background. Patterns can help to distinguish between the risers of different series in a graph without having to rely solely on color. For example, the three bar risers shown below can be distinguished by color as well as by the pattern that they use.



- **From Graph:** Displays the color defined by the graph's default color scheme. Manually selecting this option overrides any default metric graph color options defined using the Metric Editor.
- 6 Click **OK** to save your changes and close the Format Cells dialog box.
 - 7 From the **View** menu, select **Graph View** to display the report results as a graph.

You can repeat these steps for any metrics on the report that require a specific graph color. Each metric should use a color that can be easily distinguished from the colors used for other metrics. This ensures that graph report results are easy to decipher based on color.

To enable or disable a graph report to inherit a metric's graph color in MicroStrategy Developer

Inheriting metric graph color is enabled by default. The steps below show you how to ensure that metric graph colors are inherited, as well as how to disable this formatting.

- 1 Using MicroStrategy Developer, log in to a project and navigate to a graph report.
- 2 Right-click the graph report and select **Run**. The Report Editor opens, displaying the report as a graph.
- 3 From the **Graph** menu, select **Preferences**. The Preferences dialog box opens with the Options - General page displayed.

- 4 By default, all metric-specific graph colors are inherited. To disable this inheritance and apply the graph color scheme to all metrics, in the **Other** area, clear the **Apply metric formatting when available** check box.
- 5 Click **OK** to save your changes and return to the graph report.

Formatting the Y-axis: Graph series in MicroStrategy Web

The procedure below describes how to format a graph in MicroStrategy Web using the Format: Graph dialog box, because this dialog box contains all formatting options for a graph. As an alternative, you can quickly format a graph by right-clicking the graph component and selecting a formatting option. You can also format the graph using the Formatting toolbar, which is available from the **View** menu.

Prerequisites

- You must have the necessary Web Professional privileges to format graph series.

To format the series of a graph in MicroStrategy Web

- 1 Run a report and display it in either Graph View or Grid and Graph View.
- 2 From the **Home** menu, select **Graph**.
- 3 Right-click the graph and select **Format**. The Format: Graph dialog box opens.
 - If DHTML is disabled, click **Go**. The Format: Graph panel is displayed.
- 4 Select the **Format** tab.
 - If DHTML is disabled, select the **Format** link.
- 5 From the left-most drop-down list at the top, select **Series Shapes**. From the next drop-down list, select the specific series to format. In a bar graph, you can select **Bar Riser for [name of attribute/metric combination]**. For example, this may be displayed as Bar Riser for Electronics Revenue.



In MicroStrategy Developer, you can choose to format a metric displayed as a series in a graph, to always display as a specific

color. For example, you can define a Profit metric to always display in green on graph reports. This formatting overrides the formatting options you can select in the Series Colors options. For steps to assign a specific color to a metric, see [Defining a graph color for metrics in MicroStrategy Developer, page 437](#).

- If DHTML is disabled: From the drop-down list, under **Series Shapes**, select the specific series to format. In a bar graph, you can select **Bar Riser for [name of attribute/metric combination]**. For example, this may be displayed as Bar Riser for Electronics Revenue. Click the check mark.
- 6 To apply a background color, from the **Fill Color** drop-down list, select the background color for the graph series.
 - 7 You can apply any of the following effects to the shape of the series:
 -  If DHTML is disabled, you cannot create custom colors. You also cannot apply gradients, transparency effects, or bevel effects.
 - Apply a color gradient, which is a combination of two colors. To do so, in the **Fill** area click the arrow on the right side of the color selector, and then click **Gradients**. In the Gradients dialog box that opens, select the two colors to use for the gradient. Select a shading style to determine the direction in which the two colors are blended together.
 - Adjust the transparency of the series color using the **Fill Transparency** scroll bar. Click and drag the scrolling bar until you reach the desired transparency. As you scroll, the percentage of transparency is displayed. When you set the transparency to a low percentage, the colors of the object appear brighter. When you set the transparency to a high percentage, the colors of the object appear more faded.
 - 8 In the Line/Border area, you can define various formatting for the border line of the series. For example, from the **Color** drop-down list, specify a color for the border lines.
 -  By default, all series in a graph share the same border formatting, which means that changing the border formatting for one series applies to all series for the graph. To define unique border formatting for each series in a graph, you must first select the general graph preferences option **Allow different borders per series** using MicroStrategy Developer. Once this option is selected and the changes are saved using MicroStrategy Developer, unique border formats for each series in a graph can then be applied and saved from either MicroStrategy Web or Developer.

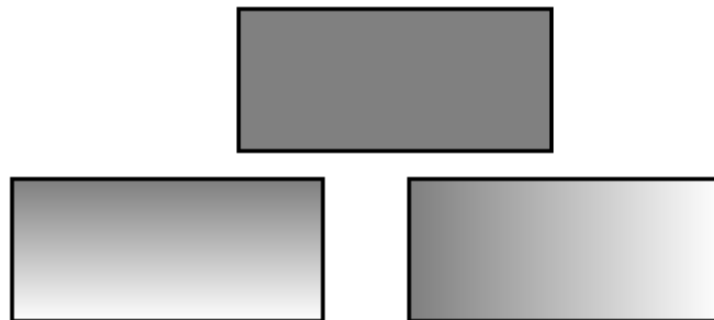
- 9 If you are formatting a graph that contains lines, such as a Line, Area, Polar, or Radar graph, you can specify whether the lines in the graph are straight or curved. In many cases, curved lines provide a cleaner look and feel. To apply curved lines, on the **General** tab select the **Use curved lines** check box.
- 10 If the graph report includes series markers, from the left-most drop-down list at the top of the Format tab, select **Series Markers**. You can define various formatting for the series markers. For example, from the **Effects** drop-down list, select a bevel effect, such as Chiseled Edge or Sphere.

The Donut and Sphere effects are applied to all circular graphs (such as a Pie graph), polygonal graphs (such as a Radar graph), and rectangular graphs (such as a bar graph). The Smooth and Chiseled Edge effects are not applied to any circular types of graphs.
- 11 Click **Apply** to update the graph's appearance or click **OK** to update the graph and close the dialog box.

Applying gradient colors to a graph in MicroStrategy Web

Gradient colors blend two colors to create a gradual color change in the background of a graph or a control in a Report Services document. (For an introduction to controls in documents, see the *Report Services Document Creation Guide*.)

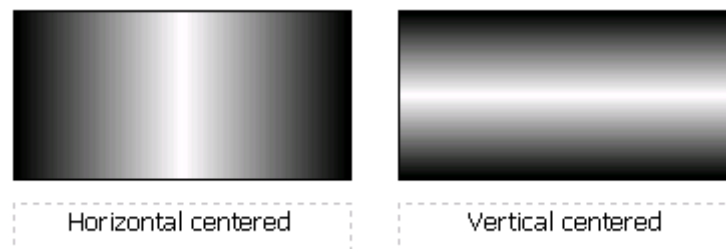
In the example shown below, the rectangles on the bottom use gradient colors, while the rectangle on the top uses a single, solid color.



You can select the two colors, as well as the direction of the blending. The direction is called the gradient shading style, and several variants are available within each shading style (for example, Horizontal centered is one variant of the Horizontal shading style). The available shading styles are:

- Horizontal
- Vertical
- Diagonal Up
- Diagonal Down
- From Corner
- From Center
- Custom

The Horizontal and Vertical shading styles contain Horizontal centered and Vertical centered variants. These variants, shown below, are three-way or mirror gradients.



These types of gradients blend the colors from the middle of the graph series out to the edges. One color is applied in the center of the graph series and the other to the edges of the graph series, then the two are blended. The two halves are identical, as though a mirror was placed along the center. You can choose to blend the colors horizontally or vertically, as shown in the rectangles above, by specifying the gradient shading style and variant.



The procedure below describes how to format a graph in MicroStrategy Web using the Format: Graph dialog box, because this dialog box contains all formatting options for a graph. As an alternative, you can quickly format a graph by right-clicking the graph component and selecting a formatting option. You can also format the graph using the Formatting toolbar, which is available from the **View** menu.

To apply gradient colors to the series of a graph in MicroStrategy Web

- 1 Run a report, and display it in either Graph View or Grid and Graph View.

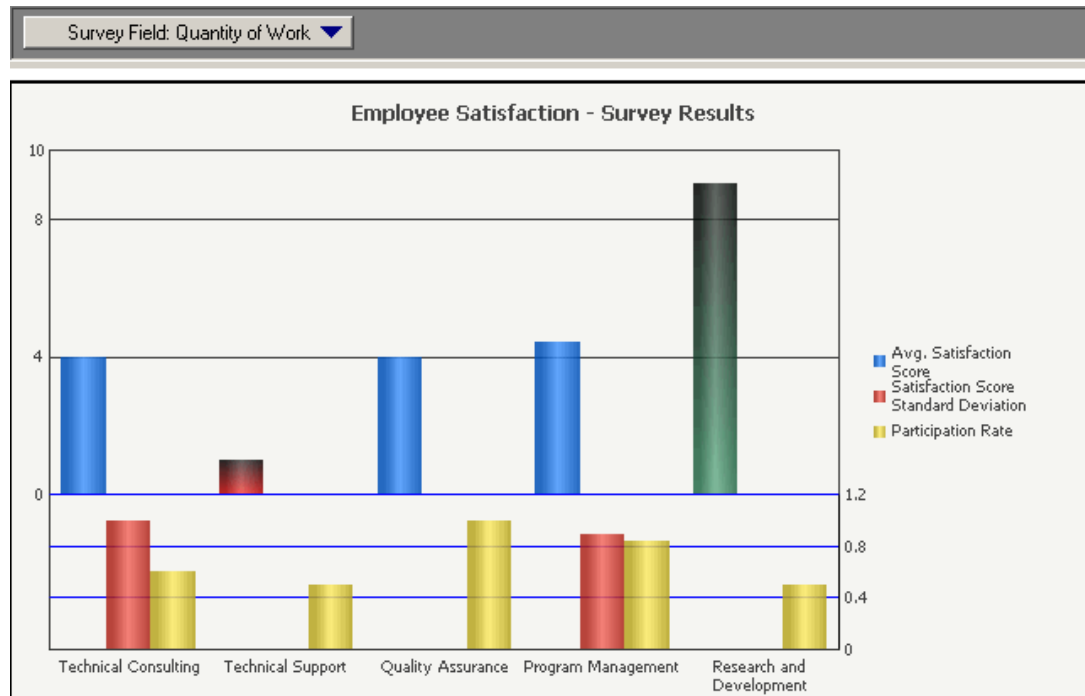
- 2 Right-click the graph to modify, select **Fill**, and then **Gradients**. The Gradients dialog box opens.
- 3 Select the two colors to use for the gradient from the **Color 1** and **Color 2** drop-down lists.
- 4 From the **Shading Styles** area, select a shading style in which to display the gradient. The shading style determines the direction in which the two colors are blended together.
- 5 If you select **Custom** from the Shading Styles area, you can specify the following:
 - **Transition Type**: Determine the direction in which the gradient colors transition.
 - **Angle**: Determine the angle at which the gradient colors are blended.
 - **X Offset**: Determine the horizontal position of the point at which the gradient colors are blended.
 - **Y Offset**: Determine the vertical position of the point at which the gradient colors are blended.
- 6 Select a graph variant on the right.
- 7 Click **OK** to apply the changes.

Displaying thresholds on graph reports

Thresholds are a standard MicroStrategy analysis feature that allow conditional formatting for metric values. This conditional formatting is called a threshold, because when a given value in your data reaches a certain point, or threshold, the conditional formatting is applied (see [Formatting thresholds, page 320](#)). This conditional formatting is displayed on grid reports by default, but it can also be displayed on graph reports.

Enabling the display of thresholds on a graph report allows the conditional formatting to be applied to the series of a graph report, thus highlighting data of particular importance on the graph. The graph report shown below

uses thresholds to highlight when employee satisfaction is above seven or below three for a given survey field.



In the report shown above, gradients are used to highlight each threshold on the Avg. Satisfaction Score metric. These two thresholds help to distinguish when employee satisfaction is considerably high or low.

Any background effect can be used to display a threshold. This includes a solid color, a pattern, a gradient, and any other available background effects. If your thresholds that are displayed on a grid report only include formatting such as symbols and text formatting, you can add background formatting to the threshold so that the background formatting can be displayed when the report is displayed as a graph.



Note the following:

- If a threshold does not include any background formatting, the threshold formatting cannot be displayed on a graph report.
- If a threshold includes background formatting as well as other types of formatting, only the background formatting can be displayed on the graph report.

Not all graph types can display thresholds. For example, Area graphs combine all data into a single area object. Since the data is all visually connected into the same area object, thresholds cannot be displayed for

specific data points. However, many graph types display a separate series or data marker for each data point, and thus can display thresholds.

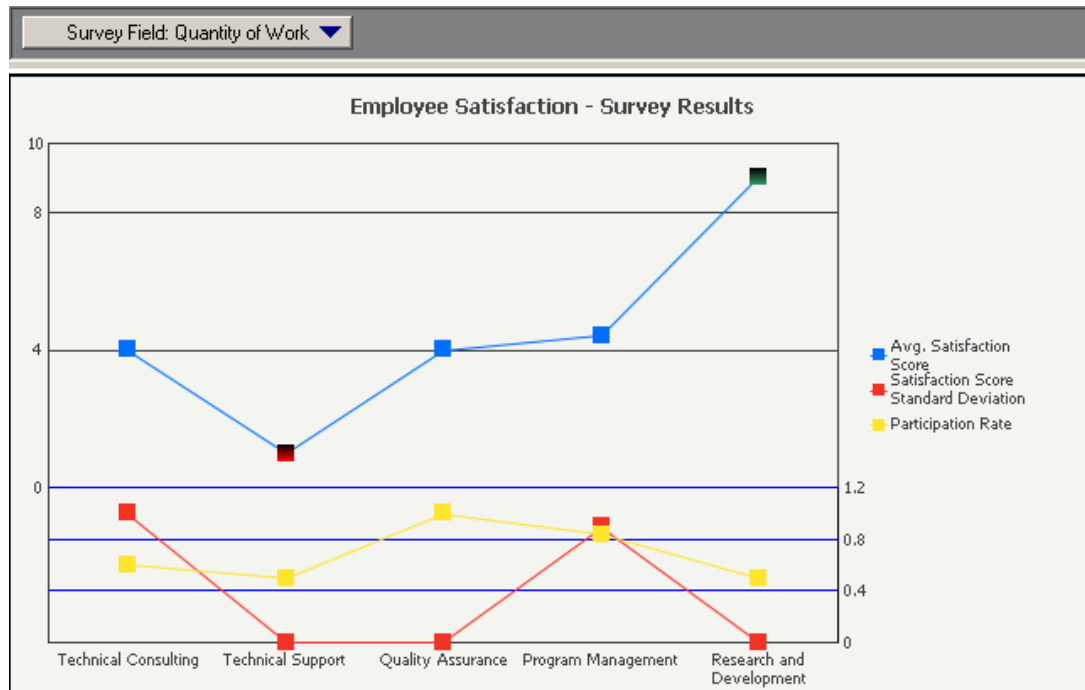
The following graph types can display thresholds directly on the series of the graph:

- Bar
- Boxplot
- Bubble chart
- Funnel
- Histogram
- Pareto chart
- Pie chart
- Stock

The following graph types can display thresholds on the data markers that highlight specific data points on the series of the graph:

- Gauge
- Line
- Polar chart
- Radar line chart
- Scatter plot and three-dimensional scatter plot

For example, the graph report shown below is the same report used in the previous example. However, while the previous example used a Bar graph style, the report shown below uses a Line graph style.



Rather than highlighting the entire line of the Line graph, the threshold formatting is displayed only on the data markers.

Prerequisites

- You have created a graph report, which also includes thresholds on the data.
- The thresholds that you plan to display on the graph report include some type of background effect such as a solid color, a gradient, or a pattern. Background formats are the only threshold formatting that can be displayed on graph reports.

To enable the display of thresholds on a graph report

- 1 Using MicroStrategy Web, log in to a project and navigate to a graph report that includes thresholds.
- 2 Run the report and display it in Graph View.

- 3 From the **Data** menu, select **Advanced Thresholds Editor**. The Advanced Thresholds Editor dialog box opens.
- 4 From the list of thresholds, select a threshold. On the toolbar, click **Enable threshold on Graph** (). Repeat this step for any thresholds that are to be displayed on the graph.

If a threshold does not include any background formatting, it cannot be displayed on a graph report and the **Enable threshold on Graph** option is not available.

If you have a threshold that should be displayed on a graph, you can add a background effect to the threshold. You can then enable the background formatting of this threshold to be displayed on the graph report. This background effect is also displayed on the report when it is displayed as a grid.

- 5 Once you have enabled the required thresholds to be displayed on the graph report, click **OK** to close the Advanced Thresholds Editor dialog box.
- 6 Click **OK**.
- 7 Review the graph report. The thresholds should be displayed for applicable data on the graph report.

If you are using one of the graph types that support thresholds, and you do not see thresholds on the graph report, use the steps provided below to display the thresholds:

- a From the **Format** menu, select **Graph**. The Format:Graph dialog box opens.
- b Click **General** and clear the **Apply rounded effects to all series** check box. This removes the automatic beveling effects used for the series and data markers of the graph report, but it also allows the display of thresholds for certain graph types. You can still apply and create your own custom bevel effects for the graph report, as described in [Manually formatting graph series and objects](#), page 451.
- c Click **OK** to save your changes and return to the graph report. The thresholds should now be displayed on the graph report.

Manually formatting graph series and objects

While you can apply broad formatting changes to your entire graph (as explained in *Formatting a graph using colors, shading, and effects*, page 435), it is often convenient to manually format and fine-tune individual graph objects. For example, you can adjust colors, and apply a variety of graphical effects, including gradients, transparency, patterns, pictures, textures, and bevel effects to the objects.

To manually format individual graph objects

- 1 If the report is not already displayed as a graph, from the **View** menu, select **Graph View** to display the report as a graph.
- 2 Right-click a graph object such as a riser, marker, graph frame, legend, and so on, and select **Format name of object** from the right-click menu. The Preferences dialog box opens.



This option is named based on the object you select. For example, for a line in a Line graph, this option is named Format Data Marker.

- 3 You can use the following formatting categories to format graph objects:



The formatting categories Font, Line, Fill, Border, and Number Formatting are available depending on the graph object that is being formatted. For example, you cannot modify the font of a graph object which cannot hold any text. Also, the Number Formatting category is only shown if the graph object includes numeric characters.

- **Font:** Select this category to modify the text of a graph object. You can modify the font type, font size, font style, color, and text frame formatting.
- **Line:** Select this category to modify the formatting of a Line graph object. You can modify the line style, color, thickness, and transparency of the line.

This category may use a different name to reflect the type of line you are formatting such as **Axis Line**, **Major Gridlines**, **Data Line**, and so on.

- **Border:** Select this category to modify the formatting of a border graph object such as a frame or background border. This category

provides the same line style, color, thickness, and transparency of the line options available for the **Line** category.



By default, all series in a graph share the same border formatting, which means that changing the border formatting for one series applies to all series for the graph. To define unique border formatting for each series in a graph, you must first select the general graph preferences option **Allow different borders per series** using MicroStrategy Developer. Once this option is selected and the changes are saved using MicroStrategy Developer, unique border formats for each series in a graph can then be applied and saved from either MicroStrategy Web or Developer.

- **Number Formatting:** Select this category to modify the numeric formatting of a graph object. Each numeric format type provides various formatting options to match the formatting for your numeric data. For information on the Number Formatting options and how to use them in graphs, see [Formatting numeric values on a graph, page 468](#).
- **Fill:** Select this category to modify the background effects of a graph object. You can modify the color, background effect type, bevel formatting, and transparency of the graph object. Some of these formatting options include:



When defining the Fill options for a series object of a graph, the options described below are ignored if the metric for the series has a metric-specific graph color. To apply the manual Fill options described below for a series object, you can disable the display of metric-specific graph colors by clearing the Apply metric formatting when available check box. Steps to perform this configuration are described in [Defining a graph color for metrics in MicroStrategy Developer, page 437](#).

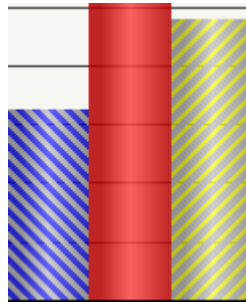
- From the **Background type** drop-down list, select **Gradient** to apply a gradient. This type of background style displays a blended effect to gradually change from one color to a second color. For example, the image shown below displays a gradient that changes from a light yellow color to a dark green color.



- Use the **Transparency** slider to apply a transparency effect. The slider uses a range of zero to 100. A value of zero means the object

is not transparent and thus appears as a solid object. Higher values make the object more transparent, which allows you to see objects behind the transparent object.

- Click **Bevel** to apply a bevel effect to the object's background. Bevel effects apply three-dimensional framing effects to the graph object.
- From the **Background type** drop-down list, select **Pattern** to apply a pattern to the object's background. Patterns can help to distinguish between different series risers in a graph without having to rely solely on color. For example, the three bar risers shown below can be distinguished by color as well as by the pattern they use.



- From the **Background type** drop-down list, select **Picture** to apply a picture to the object's background. Pictures can be applied to graph frames or backgrounds to apply a standard company background style or watermark to all graphs.
 - From the **Background type** drop-down list, select **Texture** to apply a texture to the object's background. Textures can help to distinguish between different series risers in a graph without having to rely solely on color. However, the formatting applied for each texture can be drastically different from other textures. For this reason it is important to make sure the textures you apply to your graph objects are visually appealing when displayed on the same graph.
- 4 Click **Apply** to apply any formatting changes to the object.
 - 5 Click **OK** to close the dialog box and return to the graph.

Creating and formatting titles

All graphs can include titles and subtitles which can help to clearly identify the graph. They can also include footnotes that provide additional information about the graph.



By default, the title of a graph is the name of the report, but you can manually change this as described in the procedure below.

Most graphs can also include a category axis title and a numeric Y1-axis title. Some graph styles, such as Histograms, and graph sub-types, such as 3D graphs, can include additional titles, as described in the table below.

Graph Sub-Type/ Graph Style	Additional titles that can be included
All Surface graphs	Series Axis Title, which provides additional detail on the information that is being shown on the series axis
All 3D graphs	Series Axis Title, which provides additional detail on the information that is being shown on the series axis
All Dual-Y graphs	Value Title (Y2), which can provide information about the Y-2 axis
All bi-polar graphs	Value Title (Y2), which can provide information about the Y-2 axis
Bubble	X-Axis Title
Scatter	X-Axis Title
Histogram	X-Axis Title

Creating and adjusting titles in MicroStrategy Developer

The steps below show you how to display and format titles for a graph.

To create and adjust titles in MicroStrategy Developer

- 1 Open a report in Graph View.
- 2 From the **Graph** menu, select **Titles and Labels**. The Preferences dialog box is displayed.
- 3 Within the **Titles** category, select the check box for the title or label to display. For example, you can select the **Title**, **Footnote**, or **Subtitle** check boxes to display these titles and labels.

- 4 Type the text to display for the title or label.

You can also use variables that display various report information for the titles and labels such as the report name, report creation time, report description, and so on. For information on creating a dynamic title, see [Creating dynamic titles using variables in MicroStrategy Developer, page 455](#).

- 5 Click **Apply** to apply any title changes.
- 6 Click **OK** to close the dialog box and return to the graph.
- 7 From the **File** menu, select **Save** to save the updated information.

Creating dynamic titles using variables in MicroStrategy Developer

For some types of titles, you can use variables so the titles become dynamic. Variables generate the titles automatically every time the report is run. Variables can be used to display the report name, current date and time, report description, name of the project, name of the user executing the report, and so on. You can use multiple variables in the same title or combine static text and variables.

The following table describes the variables you can use to insert dynamic text into your graph's titles and labels.

Variable	Description
{ &REPORTNAME }	Name of the report.
{ &REPORT DESCRIPTION }	Report description as entered in the Properties dialog box.
{ &CREATIONDATE }	Date the report was created. All dates and times are displayed in the format designated for the computer.
{ &CREATIONTIME }	Time the report was created. All dates and times are displayed in the format designated for the computer.
{ &AXISINFO }	(Scatter plots and Bubble charts only) Innermost label from the grid that represents either the X or Y metric.
{ &PROJECT }	Project in which the report is stored.
{ &USER }	Full name (not MicroStrategy Developer login) of the user executing the report.

Variable	Description
{ &PROMPTn& }	User's answers to prompts in the report, where <i>n</i> is the number of the prompts, in order; {&PROMPT1&} returns the answer to the first prompt, {&PROMPT2&} returns the answer to the second prompt, and so on. Applies to object prompts and attribute qualification prompts only. If <i>n</i> is greater than the number of prompts in the report, the code itself is displayed in the report.
{ &PROMPTDETAILS }	All prompt answers for the report. Applies to metric qualification prompts, object prompts, and attribute qualification prompts only. This variable is useful if you do not know the exact order or number of prompts in the report.
{ &EXECUTIONTIME }	Date and time the report was executed. All dates and times are displayed in the format designated for the computer.
{ &FILTERDETAILS }	Report filter and report limit used in the report. Applies to attribute element list prompts, hierarchy qualification prompts, metric qualification prompts, and attribute qualification prompts only, as well as all filters and limits that do not contain prompts. If the report does not have a filter, the text "Empty Filter" is printed. If the report does not have a limit, the text "Empty Limit" is printed.
{ &PAGEBYDETAILS }	Current attribute elements selected in the page-by.

Positioning and formatting graph titles in MicroStrategy Developer

You can also determine the location, font, and color of any of the titles by using various formatting options, as described in the procedure below.

To change the formatting of a graph title in MicroStrategy Developer

- 1 Open a report in Graph View.
- 2 On the **Graph** menu, point to **Formatting**, and then choose the title to format. The Preferences dialog box opens.
- 3 Define the color, font, alignment, frame, and other font options for the text.
- 4 Click **Apply** to apply any formatting changes.
- 5 Click **OK** to close the dialog box and return to the graph.

6 On the **File** menu, select **Save** to save the new graph title format.



Note the following:

- When you view a report in Graph View and set the font size for the axis label to a very large value, the graph size shrinks to accommodate the font size for the axis label. However, this functionality differs in 3D graphs, as explained in [Resizing a 3D graph using Auto Arrange, page 488](#).
- You can also apply a specific font to graph titles and labels in selected sets of reports and templates throughout a MicroStrategy project. To do so, use the Find and Replace feature in MicroStrategy Developer. You can access this feature from the Tools menu. For more information on the Find and Replace feature, refer to [Find and replace report, template, and metric formatting, page 329](#).

Displaying and formatting titles in MicroStrategy Web

You can create titles for a graph or various graph components, and you can change the titles that appear on a graph, including the font and text size. Titles include:

- Title of the graph
- Subtitles
- Footnotes
- Names of graph axes

You can automatically update a title or part of a title with variable information, such as the name of the user who executes the report, the date and time the report was created, and so on. These are called dynamic titles. The table below provides shortcuts to use in a dynamic title.

To display and format titles in MicroStrategy Web

- 1** Run a report, and display it in either Graph View or Grid and Graph View.
- 2** From the **Format** menu, select **Graph**. The Format: Graph dialog box opens.
 - If DHTML is disabled, click **Go**.

- 3 Select the **Format** tab. If DHTML is disabled, select the **Format** link.
- 4 Do one of the following:
 - From the left drop-down list at the top, select **Titles**. From the next drop-down list, select **All Titles**, **Title**, **Subtitle**, or **Footnote** to specify the graph title to format. Depending on the type of graph, you may also be able to format the **Category Axis Title**, **Y1 Axis Title**, or **Y2 Axis Title**.

If you select **All Titles**, you can format all titles on the graph at one time, but cannot change the actual text of the titles.
 - If DHTML is disabled: From the drop-down list, under **Titles**, select **All Titles**, **Title**, **Subtitle**, or **Footnote** to specify the graph title to format. Depending on the type of graph, you may also be able to format **Category Axis Title**, **Y1 Axis Title**, or **Y2 Axis Title**. Then click the check mark.
- 5 To display the title on the graph, select the **Show** check box.
- 6 Modify the name of the title as desired. You can use dynamic titles using shortcuts; see the table below for the text you need to type to achieve a dynamic title.
- 7 Specify the font type, size, color, and effects to apply to the selected title.
- 8 Click **OK** to update the graph and close the dialog box.

Shortcuts for dynamic titles in MicroStrategy Web

You can create dynamic titles for a graph by typing a shortcut in the title. A dynamic title is automatically generated every time a report is run. You can use multiple shortcuts in the same title, and combine static text and shortcuts.

For example, if you type `Profit Forecast - {&CREATIONDATE}` in a graph title, when the report is executed the date that the report was created is automatically displayed in the graph title, as follows: Profit Forecast - 4/15/11.

The table below lists the shortcuts you can use to create dynamic titles for a graph.

Type This Shortcut In Title	Title Displays This When Report Is Run
{&REPORTNAME}	Name of the report, designated by the report's creator.
{&REPORTDESCRIPTION}	Report description, as it appears in the report's Properties dialog box.
{&CREATIONDATE}	Date the report was created.
{&CREATIONTIME}	Time the report was created.
{&PROJECT}	Project in which the report is located.
{&USER}	Full name of the user executing the report.
{&PROMPTn&}	User's answer to a specific prompt in the report. Type a number in place of n, where n is the order number of the prompt (for example, {&PROMPT1&} returns the answer to the first prompt, {&PROMPT2&} returns the answer to the second prompt, and so on). This shortcut applies only to object prompts and attribute qualification prompts. Note: If n is greater than the total number of prompts in the report, the variable cannot be replaced with actual data, so the code itself is displayed in the title. See the next shortcut for an alternative if you do not know the number of prompts.
{&PROMPTDETAILS}	User's answers to all prompts in the report. This shortcut applies only to metric qualification prompts, object prompts, and attribute qualification prompts. This variable is useful if you do not know the exact order or number of prompts in the report.
{&EXECUTIONTIME}	Date and time the report is executed.
{&FILTERDETAILS}	Report filter and any report limit used in the report. This shortcut applies to all filters and limits in reports. It also applies to reports with attribute element list prompts, hierarchy qualification prompts, metric qualification prompts, and attribute qualification prompts. Note: If the report does not have a filter, the text "Empty Filter" is displayed. If the report does not have a limit, the text "Empty Limit" is displayed.
{&PAGEBYDETAILS}	Attribute elements currently displayed in the page-by area.

Formatting number values on a graph in MicroStrategy Web

You can format the display of numbers on a graph. For example, you can display the values along the Y-axis as currency, with two decimal places.



The procedure below describes how to display number values on a graph in MicroStrategy Web using the Format: Graph dialog box. You can also format a graph using the Formatting toolbar, which is

accessible from the **View** menu: point to **Toolbars**, then select **Formatting**. On the toolbar, click **Data Values**.

To format number values on a graph in MicroStrategy Web

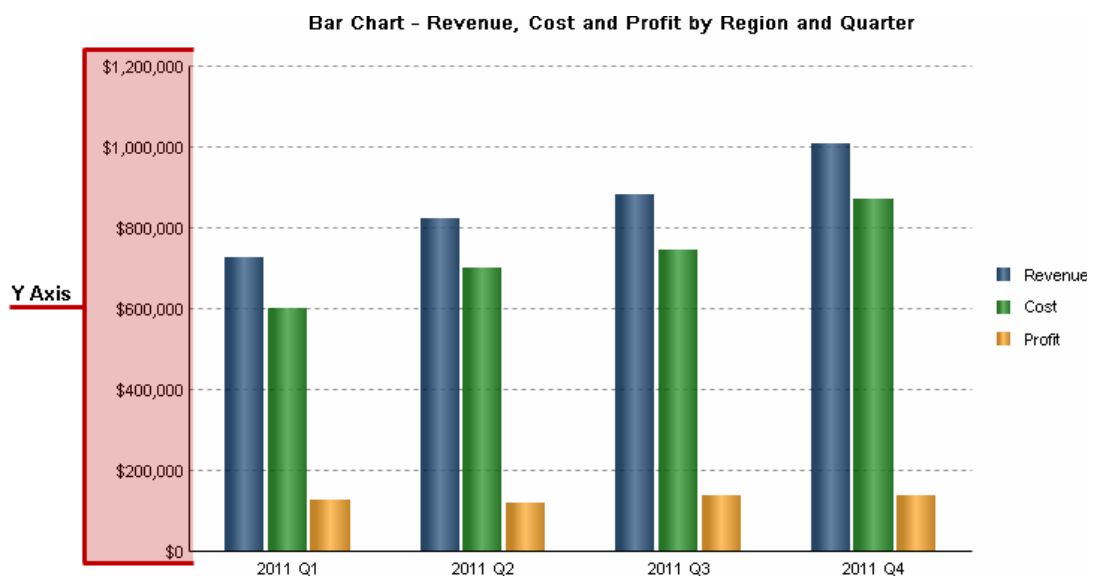
- 1 Click the name of a report to execute it. Display the report in either Graph View or Grid and Graph View.
- 2 From the **Home** menu, select **Graph**.
- 3 Right-click the graph and select **Format**. The Format: Graph dialog box opens.
 - If DHTML is disabled, click **Go**.
- 4 Select the **Format** tab. If DHTML is disabled, select the **Format** link.
- 5 From the left drop-down list at the top, select **Series Labels**. From the next drop-down list, select **All Data Values** or the specific series label to format.
 - If DHTML is disabled: From the drop-down list under **Series Labels**, select **All Data Values** or a specific series label to format. Then click the check mark.
- 6 To ensure that labels (such as currency symbols, percentages, and so on) are displayed on the graph, select the **Show** check box.

You can add tooltips that display additional information for the values on the graph. For more information, see [Creating tooltips in a graph, page 481](#).
- 7 Specify the font type, size, color, and effects to apply to the selected label.
- 8 To apply labels, such as currency, percentages, or fractions, select the **Number** tab. If DHTML is disabled, select the **Number** link.
- 9 From the **Series Labels** drop-down list at the top, select the series labels to format. Then select a format category to display, such as Currency or Percentage. When selected, some format categories have additional settings such as the number of decimal places and how to display negative numbers.
- 10 Click **OK** to apply the changes to the graph.

Formatting the axes on a graph

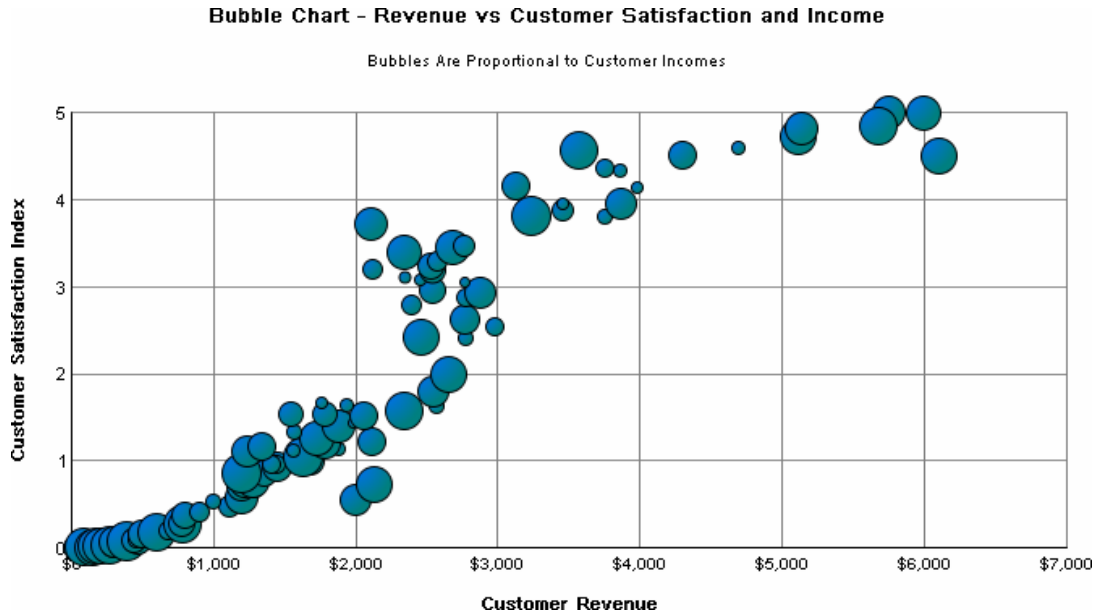
The axes of a graph help to label and give context to the data displayed on a graph. Depending on the graph style you have selected, different axes are used to display the data as a graph:

- **Y Axis** (or **Y1 Axis** for dual-axis graphs) is a primary numeric axis for graphs of all styles, except Pie, for which the Grids and Scales options is grayed out. It plots the values from the rows and columns in your data sheet. By default, the value labels on the Y axis are drawn on the left side of a vertical graph or the bottom of a horizontal graph. The Y axis is highlighted in the simple graph shown below.



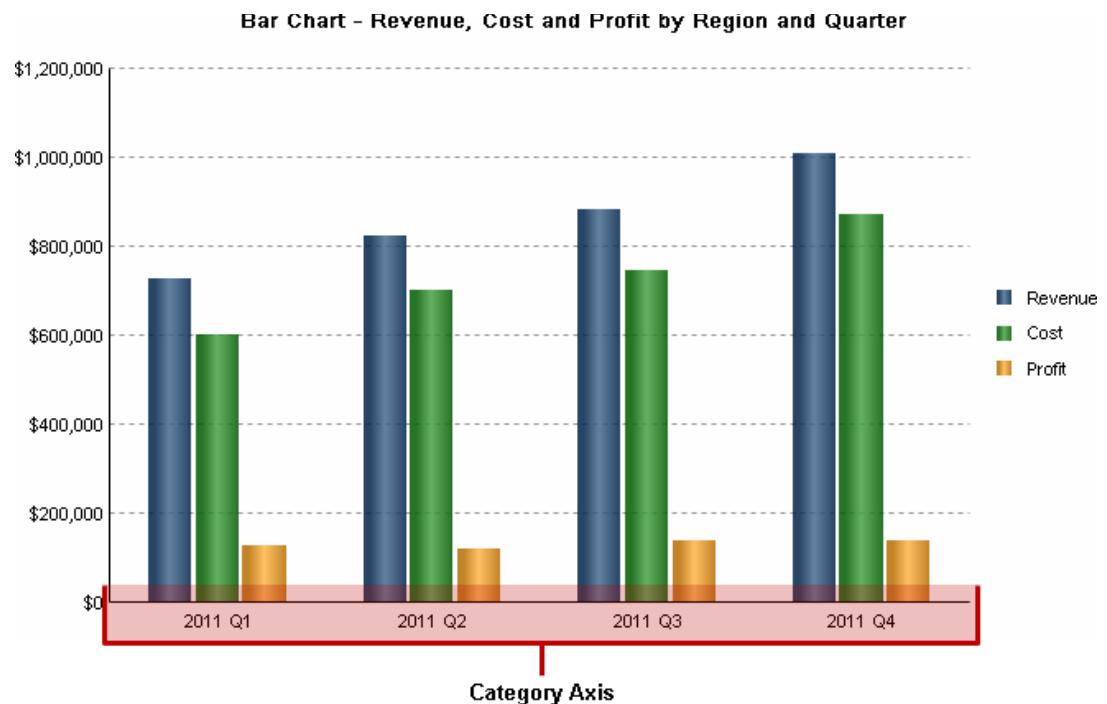
- **Y2 Axis** is an axis for dual-axis graphs, for example, Area, Bar, Bubble, Column, Line, and Scatter. Histogram, Pie, Pareto, and 3D Scatter graphs do not have a Y2 Axis. When a dual-axis graph is selected, the series are automatically divided in half, with half of them assigned to one axis and the other half to the second axis. The two axes can be drawn up or out from the same base line or physically split into two separate sections on the graph. By default, the numeric labels for the Y2 axis are displayed on the right side of a vertical graph or the top of a horizontal graph. For an example of a dual-axis graph, see [Vertical Bar: Dual-Axis Clustered graph example, page 385](#).
- **X Axis** is for Bubble, Histogram, and Scatter graphs. These graphs include two numeric axes: a Y axis that is drawn on the left side of the graph and an X axis that is drawn on the bottom of the graph frame. By default, the value labels for X Axis are displayed across the bottom of the graph.

- **Z Axis** is for Bubble and 3D Scatter graphs. For Bubble graphs, this axis is different from other graph axes in that there is no line that shows the range of values. Instead, the Z axis determines the size of the bubble, as shown in the image below.



- **Category Axis** is for Area, Column, Line, Stock, Gauge, and Surface graphs. The Category Axis defines the groups of items that are being charted.
 - For all graph types except Bubble, Histogram, and Scatter, the labels for Category Axis are taken from the values defined in the first row of your data sheet.
 - For Area, Column, Gauge, Box Plot, Pareto, and Line graphs, the Category Axis labels are drawn immediately below the graph frame.
 - For Bar graphs, the Category Axis labels are drawn immediately to the left of the graph frame.
 - For Surface graphs, the Category Axis labels are drawn on the lower right side of the 3D cube.

Along with the Y Axis, the Category axis is included on all graphs except for Pie graphs. The Category axis is highlighted in the simple graph shown below.



- **Series Axis** is for 3D Surface graphs only. In two-dimensional graphs, the series or rows of objects being charted are shown in the legend area of the graph. In three-dimensional graphs, these objects are shown on the Series Axis.

To format the axes of a graph in MicroStrategy Developer

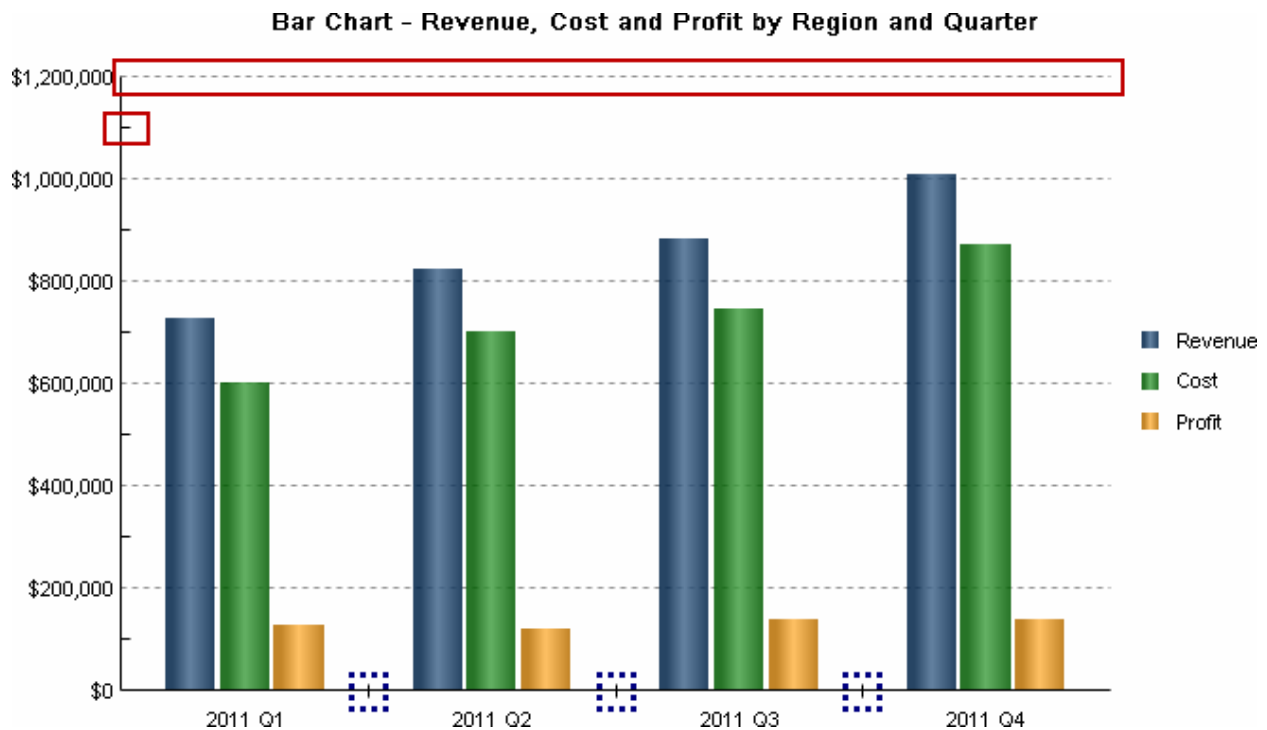
- 1 Open a report in Graph View.
- 2 From the **Graph** menu, select **Grids and Scales**. The Preferences dialog box is displayed with the Axes category selected.



If the graph is a Pie graph there is no Grids and Scales option, since Pie graphs do not have axes.

3 Within the **Axes** category, you can expand each available axis and select the following formatting options:

- **Grids and Scales:** You can define the formatting of the grid itself and its scaling. The options that are available depend on the graph type and the axis, some of the formatting options include:
 - Displaying or hiding grid lines, which provide lines and marks that denote range intervals on a graph's axis. For example, the graph below displays a major dashed grid line and a mark on the minor grid line for the vertical Y axis (highlighted with solid, red rectangles). Additionally, the horizontal Category axis includes a mark between each category to give some separation to the elements displayed on the graph (highlighted with dashed, blue rectangles).



- Displaying a custom line, which draws a line on the graph at a specific point or value. This can help to highlight a certain point or threshold of values on a graph. For an example graph that uses a custom line, see [Pareto Percent example, page 426](#).
- Using a logarithmic scale for the graph data. This can improve the display of data that follows a logarithmic trend.
- Defining how to display data on a graph that is not within the range of values defined for the graph's axes. For an example of

defining how off-scale values are displayed, see [Displaying off-scale values in a graph, page 496](#).

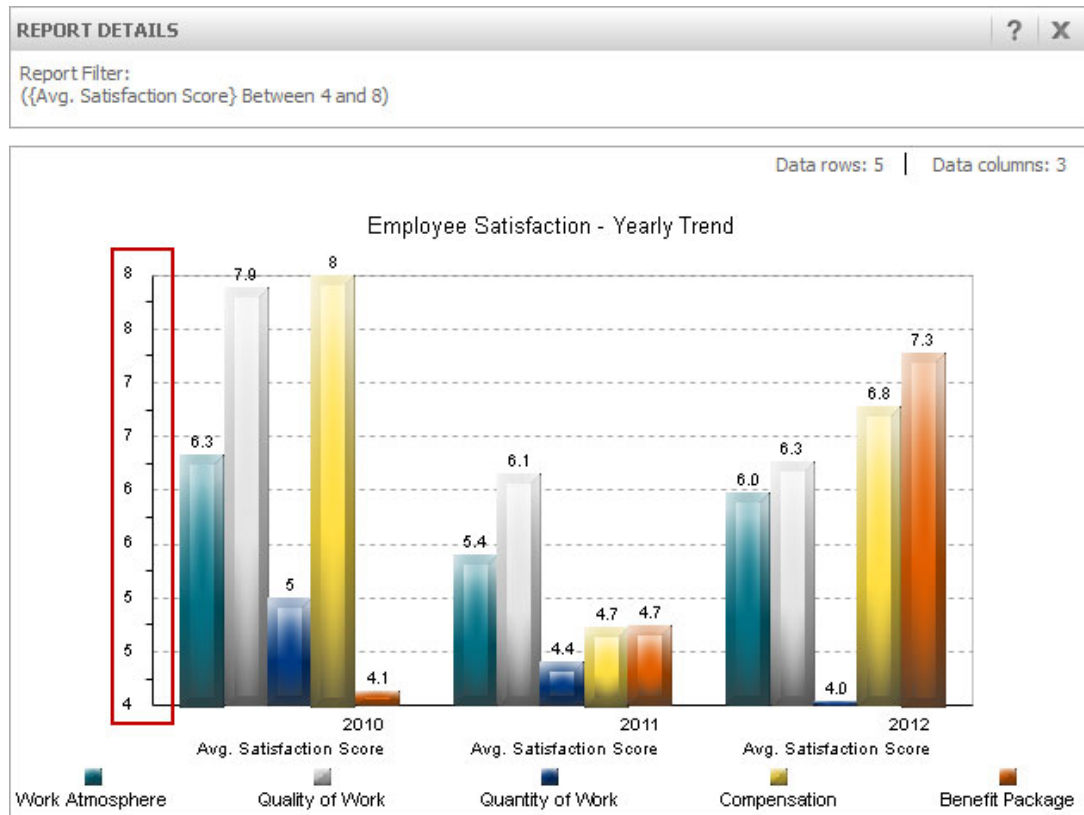
- Defining the scale, or range of values displayed on a graph. While the scale of a graph is automatically determined based on the data for a graph, you can manually define the range of values to improve the look-and-feel of the graph or to show or hide values based on your graphing requirements. For an example of manually defining the scale of a graph, see [Scaling a graph, page 489](#).
- Displaying or hiding fractional grid lines, which is described in [Displaying or hiding fractional grid lines, page 465](#).
- **Axis Labels:** You can define the formatting of the labels that are displayed for the axis. The options that are available depend on the graph type and the axis, some of the formatting options include:
 - Defining where the labels for the axis are displayed. For example, you can display labels for a vertical Y axis on the left side, the right side, or both sides of the graph.
 - Displaying the values on the axis in reverse order. For example, for graphs that include multiple years of data, you can display the values on the axis in reverse order to list the most recent data first.
 - Formatting the numeric values for axes that include numeric data, which is described in [Formatting numeric values on a graph, page 468](#).

4 Click **OK** to update the graph and close the Preferences dialog box.

Displaying or hiding fractional grid lines

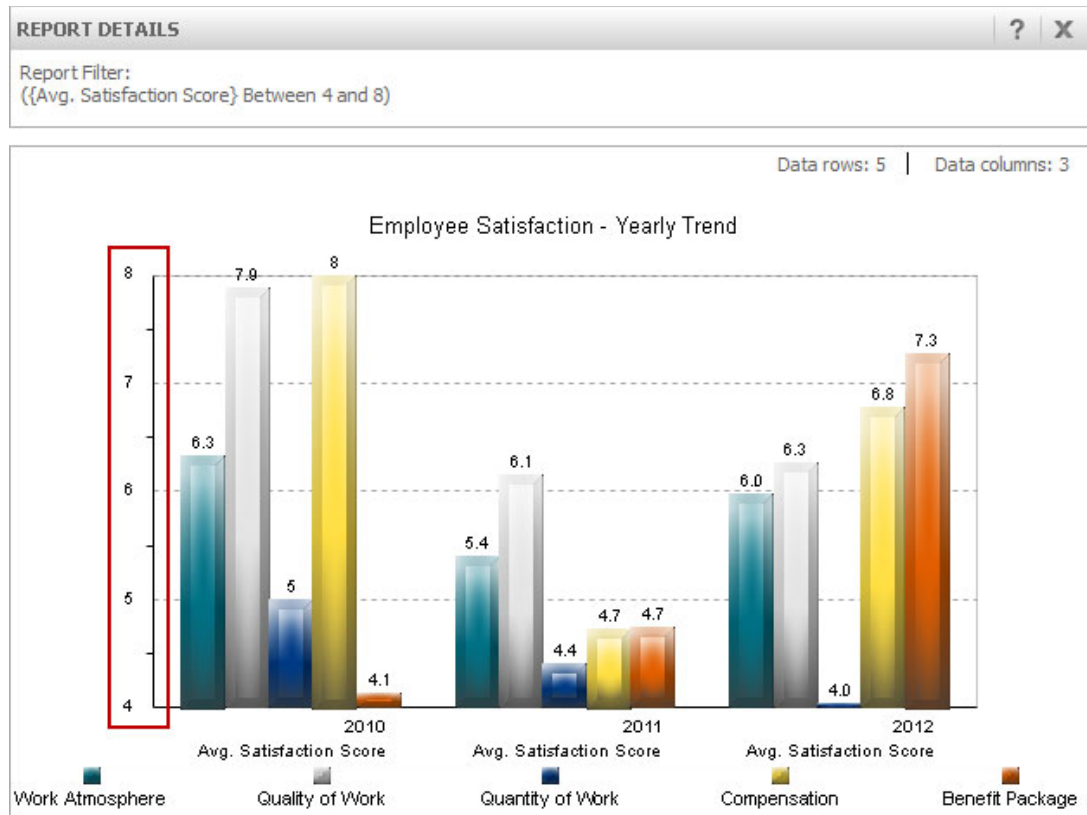
When data is displayed on a report as a graph, by default, the scale of values on an axis and the interval labels displayed for the axis are automatically determined based on the data for the report. If the range of values for an axis is relatively small, this can cause the graph report to include fractional values for the interval labels displayed on the graph's axis. If the number formatting for the axis labels uses a numeric format that does not display fractional,

decimal values, this can cause graphs that appear to be displaying duplicate values, as shown in the graph below.



One way to improve the formatting of this type of graph is to modify the numeric formatting of the axis so that it displays these fractional values. For steps to format these numeric values, see [Formatting numeric values on a graph, page 468](#).

Another option is to hide the interval labels and grid lines for these fractional values. This displays a graph with only integer values along the axis, as shown in the graph below.



The steps below show you how to display or hide fractional interval labels and grid lines.

To display or hide fractional grid lines

- 1 Open a report in Graph View.
- 2 From the **Graph** menu, select **Grids and Scales**. The Preferences dialog box is displayed with the Axes category selected.
- 3 With the Axes category expanded, select the axis that includes the duplicated interval labels. The Grids and Scales options are displayed.
- 4 Select the check box **Disable fractional grid lines**. This hides any fractional interval labels and grid lines on the graph.



You cannot define whether to display or hide fractional grid lines if you use manual graph formatting such as defining the intervals

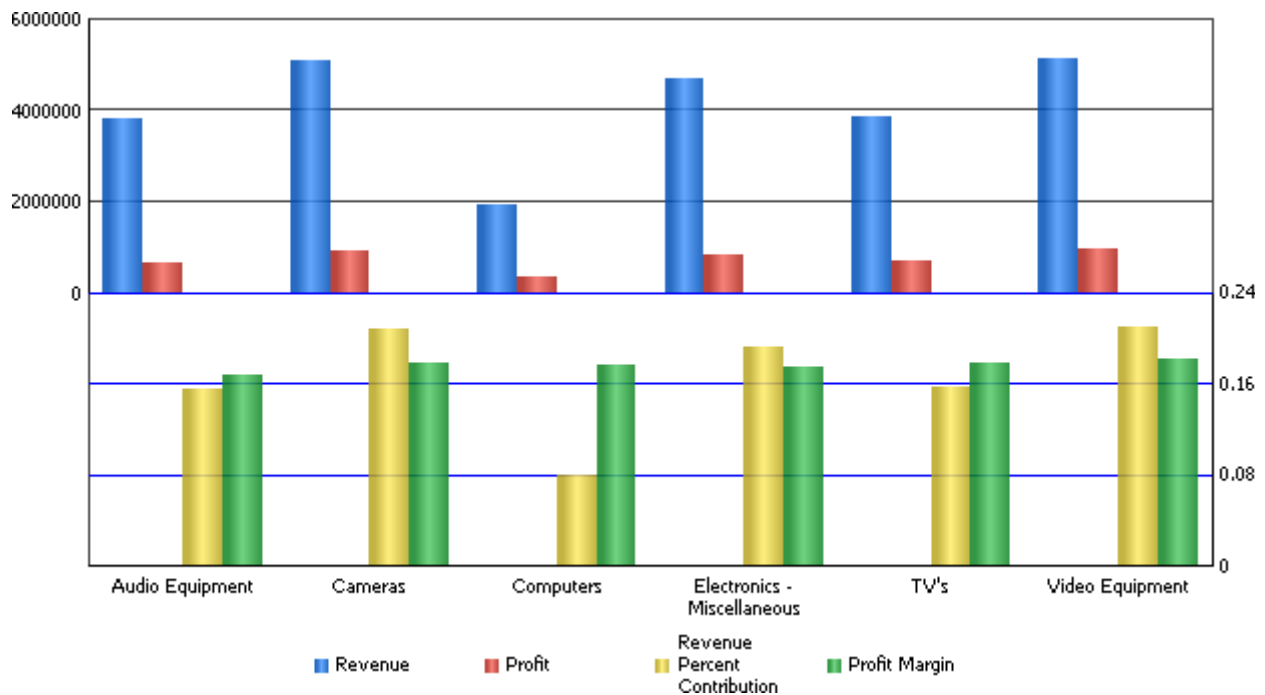
displayed on an axis or defining the scale of a graph. If the Disable fractional grid lines option is greyed out and not accessible, you may have to clear other formatting options such as Use Manual Grid or Use Manual Setting Maximum Value and Minimum Value.

- 5 Click **OK** to update the graph and close the Preferences dialog box.

Formatting numeric values on a graph

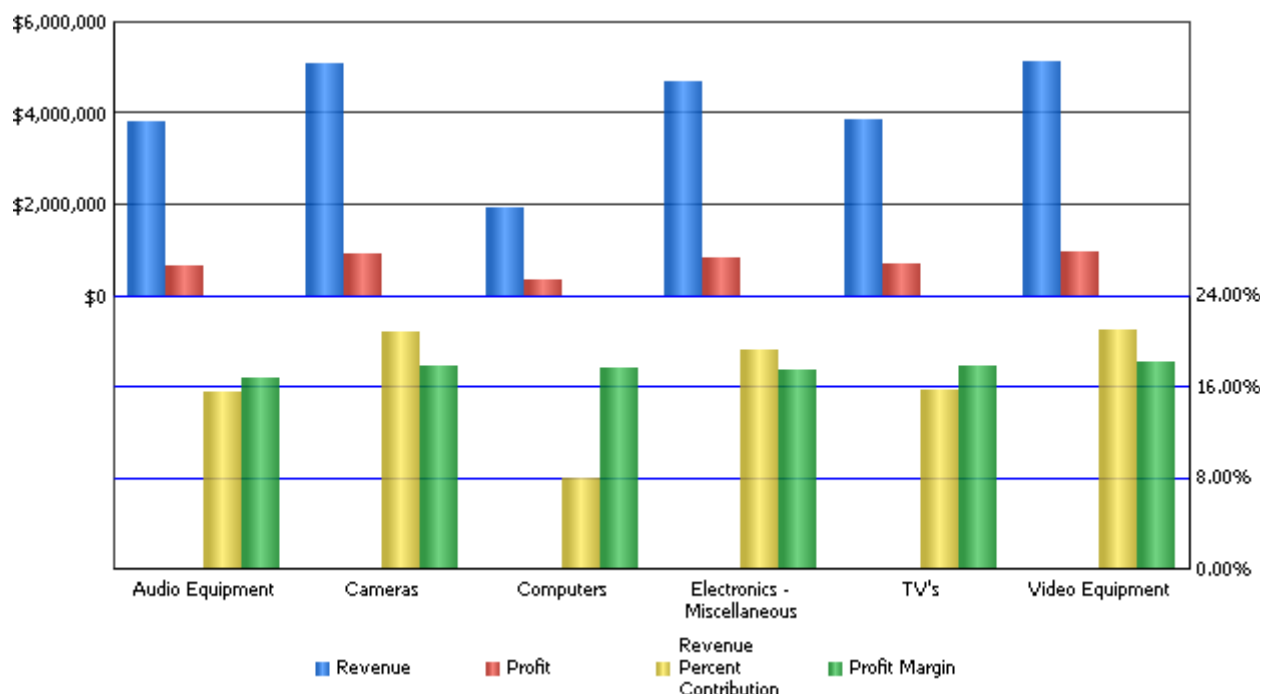
While graphs rely on colors, objects, and patterns to provide insight into data, the formatting of the numeric data is also a key component to developing a graph that is easy to analyze and interpret.

Numeric data is often included on a graph as either a label for graph risers and data points, or as a label for the increments of the graph axes. To understand the effect that numeric data formatting can have on the graph, consider the graph shown below.



This graph displays revenue, profit, revenue percent contribution, and profit margin information for various electronic product categories. Relationships can be seen between the data for metric pairs of Revenue and Profit as well as Revenue Percent Contribution and Profit Margin. However, notice that the labels for each axis use a generic number formatting. This makes it difficult to fully understand the data that is displayed for each metric.

The same graph is displayed below, with formatting applied to the numeric values that serve as labels for the increments of the graph axes.



Notice that now the Revenue and Profit metrics are on an axis that is displayed with a currency format, while the Revenue Percent Contribution and Profit Margin metrics are on an axis that is displayed with a percentage format. This simple formatting change makes it easy to understand the data values at a glance.

There are two methods to applying formats to numeric values in a graph, which are described in the following sections:

- [Inheriting the formatting of numeric values, page 469](#)
- [Manually defining the formatting of numeric values, page 472](#)

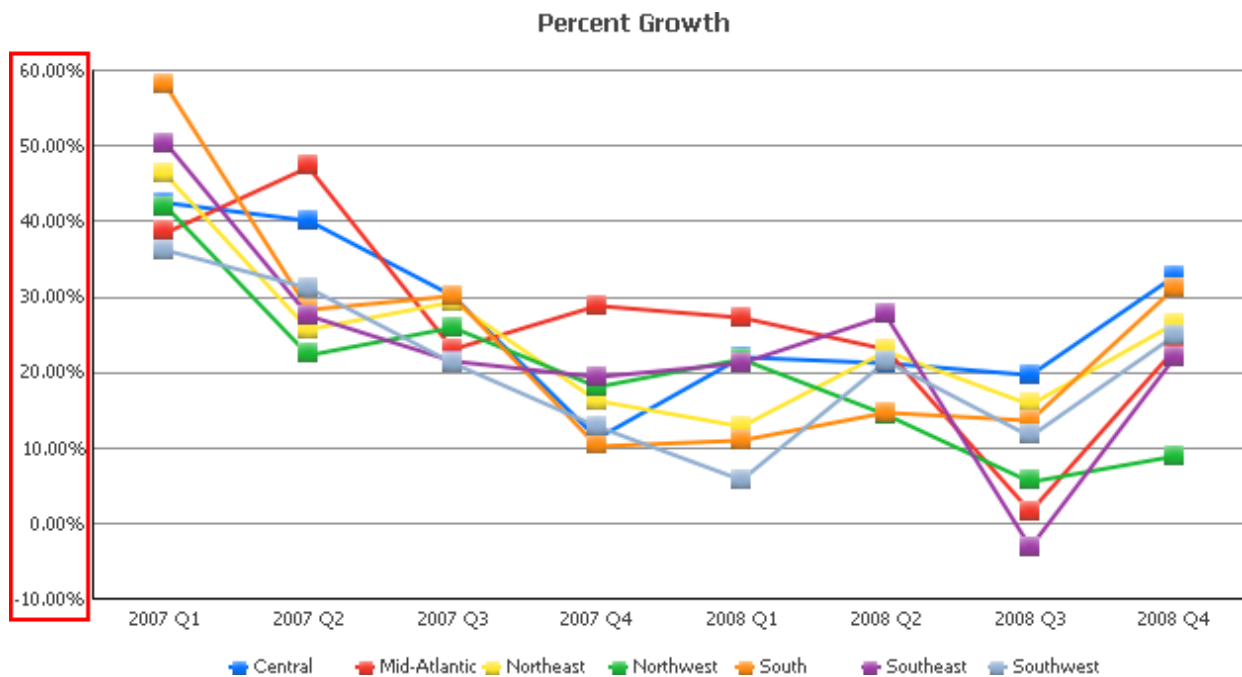
Inheriting the formatting of numeric values

Graphs in MicroStrategy can inherit the formatting of metric values and other numeric values. This allows the same default formatting seen on a grid

report to also be displayed on a graph report. For example, the simple report shown below displays the quarterly percent growth for each region.

Region	Quarter	2007 Q1	2007 Q2	2007 Q3	2007 Q4	2008 Q1	2008 Q2	2008 Q3	2008 Q4
	Metrics	Percent Growth	Percent Growth	Percent Growth	Percent Growth	Percent Growth	Percent Growth	Percent Growth	Percent Growth
Central		42.46%	40.14%	30.13%	11.24%	22.02%	21.13%	19.68%	32.69%
Mid-Atlantic		38.62%	47.27%	23.15%	28.89%	27.19%	23.01%	1.48%	22.97%
Northeast		46.22%	25.71%	29.31%	16.23%	12.94%	22.93%	15.87%	26.37%
Northwest		41.85%	22.39%	25.95%	18.04%	21.74%	14.43%	5.63%	8.94%
South		58.04%	28.39%	30.20%	10.33%	11.07%	14.58%	13.73%	31.19%
Southeast		50.30%	27.50%	21.49%	19.45%	21.13%	27.66%	-3.04%	22.02%
Southwest		36.15%	31.20%	21.15%	12.87%	5.84%	21.53%	11.65%	24.93%

Since the Percent Growth metric uses a percentage format with two decimal places, this formatting is applied by default when viewing this report as a graph.



Notice that the highlighted labels on the left use the numeric formatting of the Percent Growth metric.

If a graph does not apply the formatting of a metric's values by default, you can define the graph to inherit the numeric formatting. The steps below show you how to inherit numeric formatting for objects on a graph.

Prerequisites

- A graph report has been created.

- You are logged in to a project using MicroStrategy Developer.

To inherit the formatting of numeric values

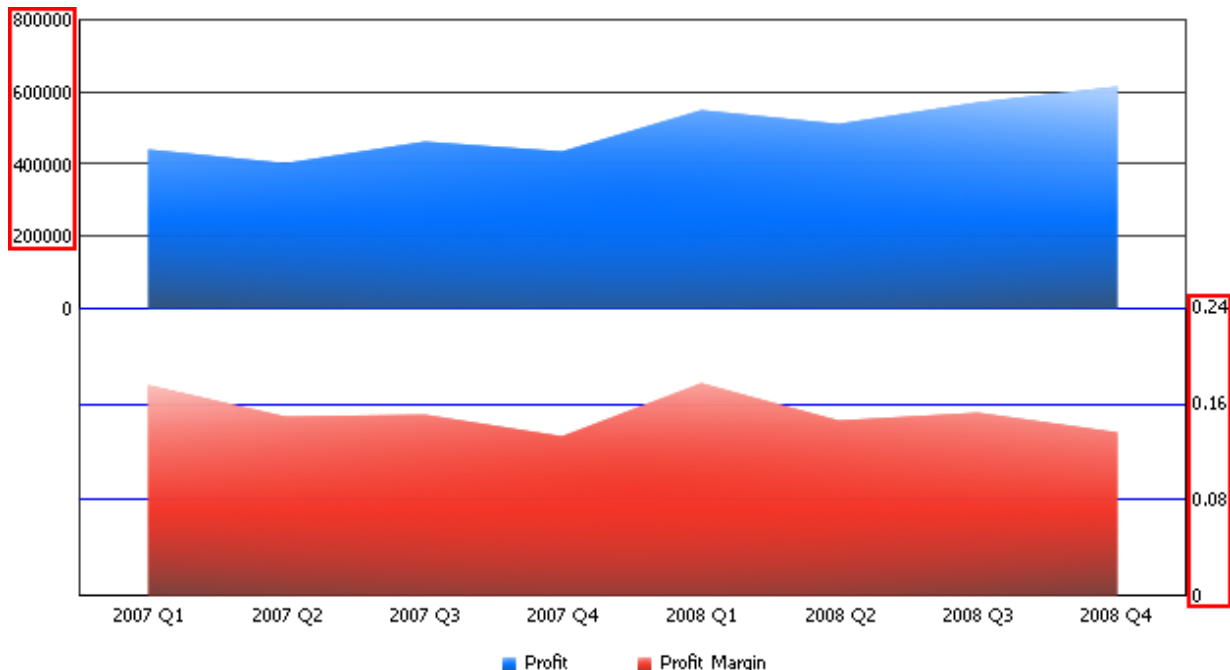
- 1 Open a report in Graph View.
- 2 From the **Graph** menu, select **Graph Options**. The Preferences dialog box opens.
- 3 Browse to the Number Formatting category for the graph object that you want to define the numeric formatting for:
 - Axis labels: To access the numeric formatting options for an axis label, expand the **Axes** category, expand the category of the axis to format (for example, **X** or **Y**), expand the **Axis Labels** category, and then select **Number Formatting**.
 - Series data labels: To access the numeric formatting options for a series data label, you must first choose to display series data labels (see [Displaying and formatting data labels, page 474](#)). Then expand the **Series** category, expand the **Labels and Values** category, and then select **Number Formatting**.
- 4 In the **Category** pane for the number formatting options, select the option to inherit numeric formatting:
 - If you are formatting values for axis labels, select **Automatic**. This applies the numeric formatting automatically retrieved for the graph.
 - If you are formatting values for series data labels, select **From Grid**. This applies the numeric formatting that is shown for the object on a MicroStrategy grid report.
- 5 Click **Apply** to accept and apply your changes.
- 6 Click **OK** to return to the graph.

Manually defining the formatting of numeric values

Graphs cannot always correctly inherit the formatting of metric values and other numeric values. For example, the report shown below includes metrics with two different numeric value formats.

Quarter	Metrics	Profit	Profit Margin
		Profit	Profit Margin
2007 Q1		\$440,716	17.64%
2007 Q2		\$402,254	14.98%
2007 Q3		\$462,211	15.07%
2007 Q4		\$434,904	13.31%
2008 Q1		\$549,010	17.64%
2008 Q2		\$512,794	14.63%
2008 Q3		\$571,770	15.33%
2008 Q4		\$615,823	13.65%

Since there is more than one numeric value format used in the grid report, the graph uses a generic numeric format by default. This is highlighted in the Area graph shown below.



While these values do not automatically represent the numeric formatting of the metric values, you can define the formatting for these values. The steps below show you how to define numeric formatting for objects on a graph.

Prerequisites

- A graph report has been created.
- You are logged in to a project using MicroStrategy Developer.

To manually define the formatting of numeric values

- 1 Open a report in Graph View.
- 2 From the **Graph** menu, select **Graph Options**. The Preferences dialog box opens.
- 3 Browse to the Number Formatting category for the graph object that you want to define the numeric formatting for:
 - **Axis labels:** To access the numeric formatting options for an axis label, expand the **Axes** category, expand the category of the axis to format (for example, **X** or **Y**), expand the **Axis Labels** category, and then select **Number Formatting**.
 - **Series data labels:** To access the numeric formatting options for a series data label, you must first choose to display series data labels (see [Displaying and formatting data labels, page 474](#)). Then expand the **Series** category, expand the **Labels and Values** category, and then select **Number Formatting**.
- 4 In the **Category** pane for the number formatting options, select the desired numeric formatting option:
 - **General:** Select this option to display numbers without any punctuation or other formatting symbols or syntax. This is often used as a default when a numeric format cannot be determined by default.
 - **Fixed:** Select this option to define the number of decimal places to show, whether to include a thousands separator, and whether to use an abbreviation such as thousands, millions, or billions.
 - **Currency:** Select this option to display number with various formatting options that reflect monetary values. You can choose from many different currency symbols, as well as how many decimal places to display.
 - **Date:** Select this option to display numbers in a format that represents the day of the year. A list of available formats to choose from is provided.

- **Time:** Select this option to display numbers in a format that represents the time of day. A list of available formats to choose from is provided.
- **Percent:** Select this option to display numbers in a percentage format. For example, a value of .275 is displayed at 27.50% when displayed as a percent. You can also define how many decimal places to display.
- **Fraction:** Select this option to display numbers as fractions. A list of available formats to choose from is provided.
- **Scientific:** Select this option to display number in scientific notation. Scientific format displays numbers as decimal numbers, with only one digit to the left of the decimal point. E notation is used to describe the true location of the decimal point.

For example, the number 123.456 in scientific format is 1.23456E+002. The E+002 means that the decimal point's true location is two digits to the right of its scientific value position. Numbers that are less than zero use a slightly different format. For example, the number .012345 in scientific format is 1.2345E-002. The E-002 means that the decimal point's true location is two digits to the left of its scientific value position.

- **Special:** Select this option to choose from various numeric formatting options. Some of the special formatting options available include:
 - Zip code
 - Extended zip code
 - Phone number
 - Social security number
- **Automatic** or **From Grid**, see [Inheriting the formatting of numeric values, page 469](#).

5 Click **Apply** to accept and apply your changes.

6 Click **OK** to return to the graph.

Displaying and formatting data labels

It is often convenient to display the numeric values for each data point on a graph. For example, you may want to see individual numeric values along the lines in your Line graph or the bars within your Bar graph.

If you choose to display data labels, you can also format them manually or allow the system to apply an automatic format style.

To display data labels on a graph report

- 1 Open a report in Graph View.
- 2 On the **Graph** menu, point to **Display**, and then choose **Data Label**. The data labels are displayed on your graph report.
- 3 The data labels and axes numbers are automatically formatted based on the formatting of the metrics in Grid View, as long as all metrics are formatted in the same manner. To apply different numeric formatting to the data labels, see [Formatting numeric values on a graph, page 468](#).

You can also manually format the data labels and axes and numbers by following the steps below.

To manually format data labels and axes numbers on a graph

- 1 In MicroStrategy Web, open a report in Graph View.
- 2 On the **Format** menu, select **Graph**. The Format:Graph dialog box opens.
- 3 You can define the number formatting for all data labels, or for each data label associated with each series in a graph:
 - To define the number formatting for all data labels, click **Number**. From the **Target** drop-down list, select **All Data Values**. From the Number area, choose the required category for formatting.
 - To define the number formatting for an individual data label, click **Number**. From the **Target** drop-down list, select **Y Axis Values**. From the Number area, choose the required category for formatting.

For detailed information on each numeric formatting option, see [Manually defining the formatting of numeric values, page 472](#).

- 4 Click **Apply** to accept and apply your changes.
- 5 Click **OK** to return to the graph.
- 6 Save your report to save the new data label or axis formatting.

Displaying and formatting a graph legend

The legend of a graph reflects the data displayed in the graph's Y-axis, also called the graph series. This is the data that comes from the columns of the corresponding grid report, and usually represents metrics.

A graph legend generally appears as a box to the right or left of your graph. The box contains small samples of each color on the graph as well as a short description of what each color means. A graph legend is a common component of any graph report, because it helps the analyst understand what the colors and shapes in the graph mean in terms of your data.

As you format your graph, you may find that you need to make a graph legend visible or format it. Steps are below for displaying, hiding, and formatting a graph legend in both MicroStrategy Developer and MicroStrategy Web.

Displaying and formatting a graph legend in MicroStrategy Developer

If a graph legend does not already appear on your graph report, display it using the following steps.

To display a graph legend on the graph in MicroStrategy Developer

- 1 Make sure you are viewing your report in Graph or Grid Graph View.
- 2 From the **Graph** menu, select **Display**, then **Legend**. A graph legend is displayed on your graph.

To remove the legend from the graph, select **Legend** again.



Some graph types such as Histogram and 3D Surface do not use graph legends. Selecting to display a legend for these types of graphs has no effect on the graph. These types of graphs use other titles and labels to provide context for data.

To define display options for a graph legend

You can define various display options for a graph legend.

- 3 From the **Graph** menu, select **Graph Options**. The Preferences dialog box opens.

- 4 Expand the category for the graph style you are using. For example, if you are formatting the legend of a vertical Bar graph, expand the **Vertical Bar Options** category.
- 5 Select the **Display** category.
- 6 You can define the following display options for a graph legend:
 - **Markers and Text:** Allows you to select from the following options to position the markers in relation to the text in the legend:
 - Markers to Left of Text
 - Markers to Right of Text
 - Text Centered on Markers
 - Markers Above Text
 - Markers Below Text
 - **Layout:** Allows you to select from the following options to define where the legend is displayed within the graph frame:
 - Legend on Right Side
 - Legend on Left Side
 - Legend Below Chart
 - **Box Style:** Allows you to select from the following options to define the type of frame to surround the text in the legend:
 - No Frame
 - Single Line Frame
 - Double Line Frame
 - Beveled Frame
 - Reverse Beveled Frame
- 7 Once your graph's legend is visible on your graph, you can choose to format it in a variety of ways. For example, you can edit the color series markers in the legend and therefore the colors of the series on your graph, or format the size and font of text elements in the legend. To format a graph legend, see [Formatting a graph legend in MicroStrategy Developer](#) below.

Formatting a graph legend in MicroStrategy Developer

You can format several aspects of a graph legend, including the text.

Prerequisites

- You have created a graph and selected to display the graph legend. For steps to display a graph legend, see [Displaying and formatting a graph legend in MicroStrategy Developer, page 476](#).

To format a graph legend in MicroStrategy Developer

- 1 View your report in Graph or Grid Graph View.
- 2 From the **Graph** menu, point to **Formatting**, and select **Legend**. The Preferences dialog box opens.
- 3 The Legend category is expanded. Select from the Fill, Border, and Font formatting categories to modify the legend formatting. For information on the options available for each formatting category, see [Manually formatting graph series and objects, page 451](#).
- 4 Click **Apply** to accept and apply your changes.
- 5 Click **OK** to return to the graph.

Displaying or formatting a graph legend in MicroStrategy Web

The steps below show you how to display and format a graph legend in MicroStrategy Web.



The procedure below describes how to format a graph using the Format: Graph dialog box in MicroStrategy Web. This dialog box contains all formatting options for a graph. As an alternative, you can quickly format a graph by right-clicking the graph component and selecting a formatting option. You can also format the graph using the Formatting toolbar, which is available from the View menu.

To display and format a graph legend in MicroStrategy Web

- 1 Run a report, and display it in either Graph View or Grid and Graph View.
- 2 From the **Format** menu, select **Graph**. The Format: Graph dialog box opens.
 - If DHTML is disabled, select **Graph**, then click **Go**. The Format: Graph panel is displayed.
- 3 Select the **Format** tab. If DHTML is disabled, select the **Format** link.
- 4 From the left-most drop-down list at the top, select **Format**. From the next drop-down list, select **Legend**.
 - If DHTML is disabled: From the drop-down list, under **Format**, select **Legend**. Click the check mark icon.
- 5 To ensure that the legend is displayed on the graph, select the **Show** check box, if it is not already selected.
- 6 From the **Position** drop-down list, specify whether the legend is positioned to the **Right**, **Left**, or at the **Bottom** of the graph.
- 7 Specify the font type, size, color, and effects to apply to the legend.
- 8 From the **Fill Color** drop-down list, specify the background color for the legend. You can also apply several effects to the color of the series, as described below.



If DHTML is disabled, you cannot create custom colors in the Fill Color drop-down list. Also, you cannot apply gradients, transparency effects, or bevel effects.

- Apply a color gradient, which is a combination of two colors, to the legend. To do so, in the Fill area, click the arrow on the right side of the color picker, and then select **Gradients**. In the Gradients dialog box, select the two colors to use for the gradient. Then, select a shading style in which to display the gradient to determine the direction in which the two colors are blended together.
- Adjust the transparency of the color of the legend using the **Transparency** scrolling bar. Click and drag the scrolling bar until you reach the desired color transparency. As you scroll, the percentage of transparency is displayed in a tooltip. When you set the transparency to a low percentage (by dragging the bar to the left), the colors of the

object appear brighter and more solid. When you set the transparency to a high percentage (by dragging the bar to the right), the colors of the object appear lighter and less solid.

- Select from several bevel effects from the **Effects** drop-down list, such as Chiseled Edge or Sphere, to apply to the legend.
- 9 From the **Line Color** drop-down list, specify a color for the border lines of the legend.
 - 10 From the right-most drop-down list on the **Format** tab, select **Legend Area** to access formatting options for the entire background area of the legend.
 - If DHTML is disabled: From the drop-down list, under **Format**, select **Legend Area** to access formatting options for the entire background area of the legend. Click the check mark icon.
 - 11 Apply formatting changes to the legend area as described above.
 - 12 Click **Apply** to apply the changes to the graph, or click **OK** to apply the changes and close the dialog box/panel.

Formatting the labels on a graph's axes

You can access label formatting settings for the different axes on your graph. These settings include, but are not limited to the following:

- **Category axis (the axis that commonly contains the attributes):**
 - Location and positioning of labels along the axis
 - Order of categories
 - Include and format of axis labels
 - Appearance of grid lines and graph tick marks
-  For information about what categories on a graph represent, see the table in [Moving objects on a graph report, page 378](#).
- **Y-1 axis**
 - Format and location of axis labels
 - The range of numbers in labels

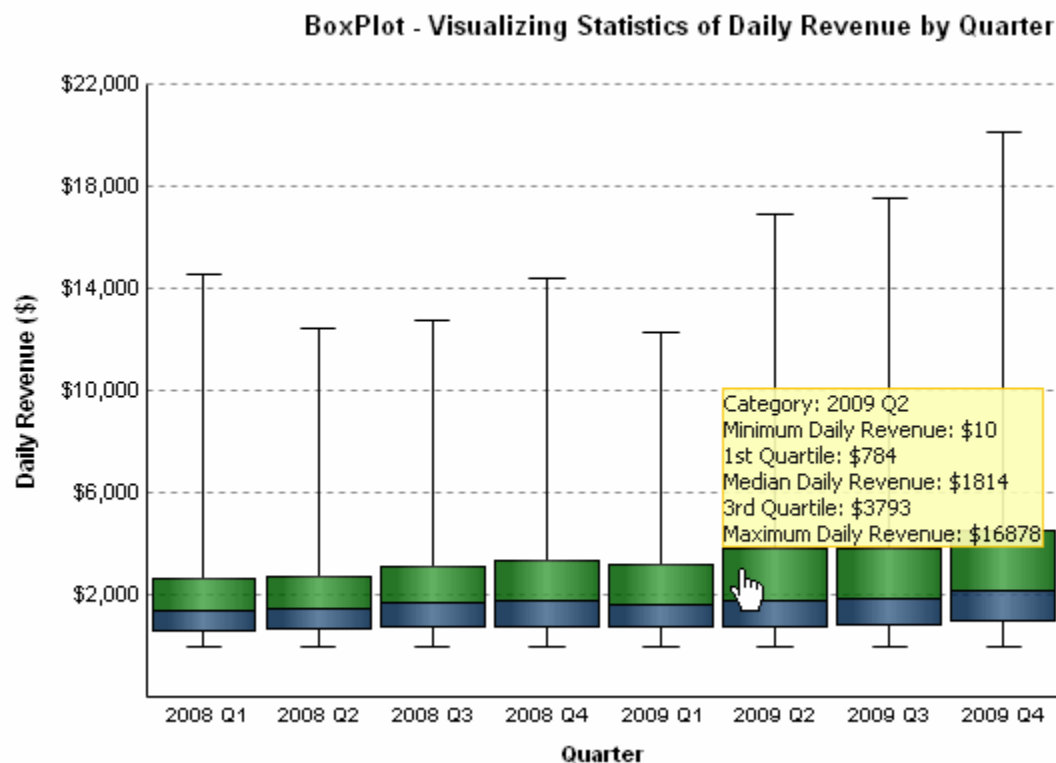
- Number format (for example, currency or percentage)
- Appearance of grid lines and graph tick marks

Depending on the type of graph you are formatting, you may also have access to scaling options for the X-axis, Z-axis, or Y-2 axis as well.

Click **Help** in MicroStrategy Developer for steps to format labels for each of these axes.

Creating tooltips in a graph

A tooltip is short, descriptive text that appears when you mouse over a graph item in MicroStrategy Web. Tooltips provide information such as the value of a specific data point, the name of a group, or the label of a graph's X-axis. An example of the default tooltip for a Boxplot in MicroStrategy Developer is shown below:



Tooltips are automatically added to every graph for display in MicroStrategy Web. These default tooltips display relevant data for a graph based on the graph type. For example, the default tooltips for Boxplots display the values of each quartile, as shown in the image above.

Tooltips are only visible in Web. However, both MicroStrategy Developer and Web allow you to modify the tooltips displayed in a graph by using macros. These macros are dynamic text that is replaced with graph information when the tooltip is displayed in Graph View. While the default macro used is { &TOOLTIP }, you can use different macros to customize what information appears in a given graph. You can use multiple macros in the same tooltip or combine static text and macros.

Some tooltip macros may not be available for use in specific types of graphs. For example, { &XVALUE } can only be used in Bubble charts or Scatter plots. If a macro is unavailable for the given graph type, no tooltip is displayed in the graph when viewed in Web.

The following table describes the macros you can use to customize graph tooltips in both MicroStrategy Developer and MicroStrategy Web:

Macro	Information Displayed
{ &TOOLTIP }	All relevant labels and values associated with a graph item.
{ &GROUPLABEL }	Name of the graph item's category. This value is often the graph item's attribute element information, as attributes are commonly used as the categories of graph reports.
{ &SERIESLABEL }	Name of the graph item's series. This value is often the graph item's metric name information, as metrics are commonly used as the series of graph reports.
{ &VALUE }	The value of a given data point.
{ &XVALUE }	The X-value of a data point. Only applicable to Bubble charts and Scatter plots.
{ &YVALUE }	The Y-value of a data point. Only applicable to Bubble charts and Scatter plots.
{ &ZVALUE }	The Z-value of a data point. Only applicable to Bubble charts and Scatter plots.
{ &XLABEL }	The label of the X-axis. Only applicable to Bubble charts and Scatter plots.
{ &YLABEL }	The label of the Y-axis. Only applicable to Bubble charts and Scatter plots.
{ &ZLABEL }	The label of the Z-axis. Only applicable to Bubble charts and Scatter plots.

For information on tooltips in documents, see the *Report Services Document Creation Guide*.

Steps are below to apply tooltips in both MicroStrategy Developer and MicroStrategy Web.

To apply a tooltip to a graph in MicroStrategy Developer

- 1 Open a report in Graph View.
- 2 On the **Graph** menu, choose **Preferences**. The Preferences dialog box opens.
- 3 Expand the **Series** category.
- 4 To apply the same tooltip to every item in the graph, select the **Labels and Values** category. Otherwise, to define tooltips for a single series, select the name of the series.
- 5 Ensure that the **Show Tooltip (Web Only)** check box is selected. If this box is cleared, no tooltips will be visible in the graph.
- 6 Type the text of the desired macro or your own static text.
- 7 Click **Apply** to apply changes.
- 8 Click **OK** to return to the graph.
- 9 On the **File** menu, select **Save** to save the new tooltip.

To apply a tooltip to a graph in MicroStrategy Web

- 1 Open a report and display it in either Graph View or Grid and Graph View.
- 2 From the **Format** menu, select **Graph**. The Format:Graph dialog box opens.
- 3 Select **Format**.
- 4 From the first drop-down list, select **Series Values**. The formatting options for series values appear.
- 5 Ensure that the **Tooltips** check box is selected. If this box is cleared, no tooltips will be visible in the graph.
- 6 Type the text of the desired macro or your own static text.

- 7 Click **Apply** to apply changes.
- 8 Click **OK** to return to the graph.

Adjusting the layout, size, and scale of a graph

You can determine how your graph is scaled, the overall layout of the graph, and how graph elements such as legends and titles are positioned on the graph.

Before you begin formatting different elements of your graph, you must first consider whether to use manual or automatic graph layout, as explained in the following section.

Pros and cons of manual and automatic graph layout

You can control the positioning of graph elements such as legends and titles manually or you can have the system automatically reposition these elements, which helps you avoid overlapping elements on your graph.

For example, suppose you add an attribute to your graph report. When the graph is re-executed, you may notice that your graph's overall layout has changed to accommodate the additional data. You have the option to enable or disable this automatic adjustment.

If you choose the automatic layout setting, any modifications that impact the layout, such as data changes, invoke the automatic positioning of elements such as legends and labels. For example, suppose you select a different page-by attribute on your grid, which results in a larger set of data with longer category names. With automatic layout the graph frame height may need to be adjusted so the axes labels fit. This adjustment is required because when the number of series increases, the height and width of the legend area may need to increase. This reduces the width of the graph's frame.

For automatic layout, if the recalculated legend size is larger than the current background frame, the size of the background frame can automatically expand. Since this option overrides other sizing options, it can lead to a frame size that is unexpected.

The automatic and manual layout options are in the Preferences dialog box, accessible from the Graph menu. By default, automatic layout is selected.

The following table provides a quick overview of the features of automatic and manual graph layout.

Layout Type	Pros	Cons
Automatic	- Graph elements are repositioned automatically as needed. For example, if you add report objects to the grid report, the graph report automatically resizes to accommodate the additional data. - Some objects such as graph legends can still be moved using other methods than manually moving the object on the graph. For information on displaying graph legends, see Displaying and formatting a graph legend in MicroStrategy Developer, page 476.	You cannot manually move any objects within the graph. Note: If you try to move an element manually, a dialog box opens. If you choose to stay in automatic layout mode, the element does not move. If you switch to manual layout, the element moves.
Manual	- You can position graph elements yourself. For example, you can drag and drop a graph, legend, title, data label, and so on to a new location. - Provides the most design flexibility. - Provides some automatic layout options such as automatically recalculating the size of the graph legend and repositioning the graph frame and legend.	You cannot benefit from the automatic resizing features offered by automatic layout.

Follow the procedure below to learn how to enable either automatic or manual graph layout mode.

To choose automatic or manual layout for graph reports

- 1 View a report in Graph View.
- 2 On the **Graph** menu, select **Preferences**. The Preferences dialog box is displayed.
- 3 Within the **Layout** category, select from the following options:
 - **Automatic layout:** Allows graph components to be repositioned automatically.
 - **Manual layout:** Allows you to position graph components yourself. If you select Manual layout, the following option becomes available:
 - **Re-calculate legend size automatically when data changes:** Automatically re-calculates the legend size when data changes. If this option is selected, you can also set the following option.
 - **Reposition chart frame and legend:** Re-positions the legends to a default location such as right or left when data changes. For

example, in a graph report, initially you placed the legend frame in the center of the graph report. If you select this check box, change the data, and re-execute the report, the legend frame is moved to one of the default locations on the graph. For more information about this setting, see [Example: Manually positioning a graph legend, page 487](#).

- **Expand background frame to accommodate recalculated legend sizes:** Automatically expands the size of the background frame if the recalculated legend size is larger than the current background frame. This setting is available only if the Automatic layout option or Manual layout: Re-calculate legend size automatically when data changes option is selected.



This option overrides other sizing options and can lead to an unexpected frame size.

- 4 Click **Apply** to accept and apply your changes.
- 5 Click **OK** to return to the graph.

Manually positioning graph elements

Sometimes you need to reposition graph objects such as the graph itself, the legend, or titles and subtitles. The extent to which you can manually control the position of graph elements is determined by the following:

- If automatic layout mode is enabled, graph elements are automatically repositioned when anything is resized or readjusted on the graph. For information about automatic layout, see [Pros and cons of manual and automatic graph layout, page 484](#).

Even if automatic layout mode is enabled, you can still choose to place graph elements in certain places on the graph. However, these locations are default locations and are not customizable. For example, with automatic layout mode enabled, you can place a graph's legend to the left, right, or below a graph by configuring options in the Graph Preferences dialog box (see [Displaying and formatting a graph legend in MicroStrategy Developer, page 476](#)). You cannot, however, drag and drop the legend onto another location.

- With manual layout mode enabled, you can manually drag and drop graph elements on your graph, as demonstrated in the following example. For information about manual layout, see [Pros and cons of manual and automatic graph layout, page 484](#).

Example: Manually positioning a graph legend

If you set your graph layout preference to manual layout mode (as described in [Pros and cons of manual and automatic graph layout, page 484](#)), you can select options to help you position the legend. The size of the legend can change automatically when the data changes. If you select manual layout mode, the legend can be repositioned to a default location, such as right or left, when the data changes.

For example, place the legend frame in the center of a graph report. Then, from the **Graph** menu, select **Preferences**. In the **Layout** category in the Preferences dialog box, select the **Re-calculate legend size automatically when data changes** check box, and then the **Reposition chart frame and legend** check box. Change the data of the report and re-execute it. The legend frame is moved to one of the default locations on the graph report.

To manually position a graph legend

- 1 View the report in Graph View.
- 2 On the **Graph** menu, select **Preferences**. The Preferences dialog box is displayed.
- 3 Within the **Layout** category, if manual layout is not already selected, select **Manual layout**.
 - To have the size of the legend change when the data changes, click **Re-calculate legend size automatically when data changes**.
 - If you selected the legend size recalculation option, you can also have the legend repositioned to a default location, such as right or left, when the data changes. To do this, select **Reposition chart frame and legend**. Click **OK** to return to the graph.
- 4 On the graph, click the background of the graph legend, outside of the legend text. A selection box with handles appears.

If boxes appear around the legend text, try clicking further outside of the legend text area until a selection box with handles appears.
- 5 Drag and drop the legend to the desired location.
- 6 Save the graph.

Resizing a graph

You can manually resize the elements on your graph, including the graph chart itself, if manual layout is enabled, as explained in [Pros and cons of manual and automatic graph layout, page 484](#).

The following procedure guides you in resizing the actual graph chart within your graph report.

To manually resize a graph

- 1 View a report in Graph View.
- 2 Make sure manual layout mode is enabled. See [Pros and cons of manual and automatic graph layout, page 484](#) for the necessary steps.
- 3 Click a section of the graph that is blank. For example, do not click the actual bars or lines of your graph; instead, click the graph's background. Handles appear around the graph frame.
- 4 Use the handles on each side and corner of the graph to manually resize the graph.

Resizing a 3D graph using Auto Arrange

When you set the font size for an axis label to a very large value in a non-3D graph, the graph shrinks to accommodate the new font size. However, in 3D graphs, after you resize the font to the original value, the graph does not expand to its original size. You can resize the 3D graph to its original size by using the Auto Arrange option. This option can be accessed from the Graph menu of the Report Editor.

Prerequisites

- A graph with a 3D graph style has been created.

To resize a 3D graph using Auto Arrange

- 1 View a report in Graph View, using one of the available 3D graph styles.
- 2 From the **Graph** menu, select **Auto Arrange**.

Scaling a graph

Configuring the scaling of your graph is one of the most important steps in creating a clear and useful graph report. A graph's scale determines the range of values that appears on your graph's axes, which directly affects how data is presented on your graph. Scaling differs from resizing in that scaling determines how data is measured along each axis. Resizing does not affect actual measurements of data or how that data is presented as a graph.

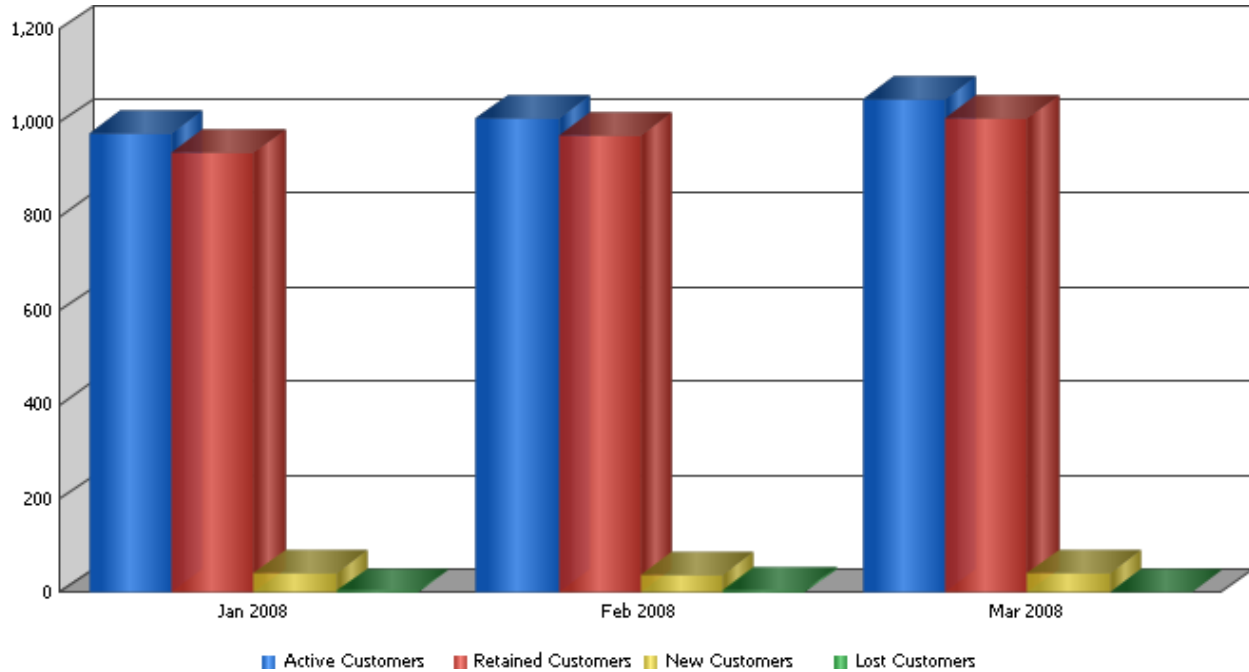
When you switch a report from Grid View to Graph View for the first time, the graph scaling is automatically set for you. For example, if revenue data does not exceed \$5,000 in your grid report, then \$5,000 may be set as the highest label on the Y-axis.

To scale your graph efficiently, analyze the data on your grid report and then base your scaling decisions on that data. For example, if profit data on your grid report does not exceed \$500,000, consider setting \$500,000 as the maximum value on the appropriate axis. This allows elements on your graph such as bars in a Bar graph to be rendered at a considerable size, making it easier for you to view and analyze the data on your graph.

Consider another example. In the report grid in the following image, notice there are no more than 1,049 active customers for any given month. Therefore, the labels on the Y-axis on the graph report do not exceed 1,200. When you switch a grid report to a graph, the scaling of the graph is automatically determined for you. In this case, MicroStrategy Developer produces a graph whose Y-axis does not exceed 1,200. This scaling allows the bars on the graph to appear at a considerable size, which makes the graph

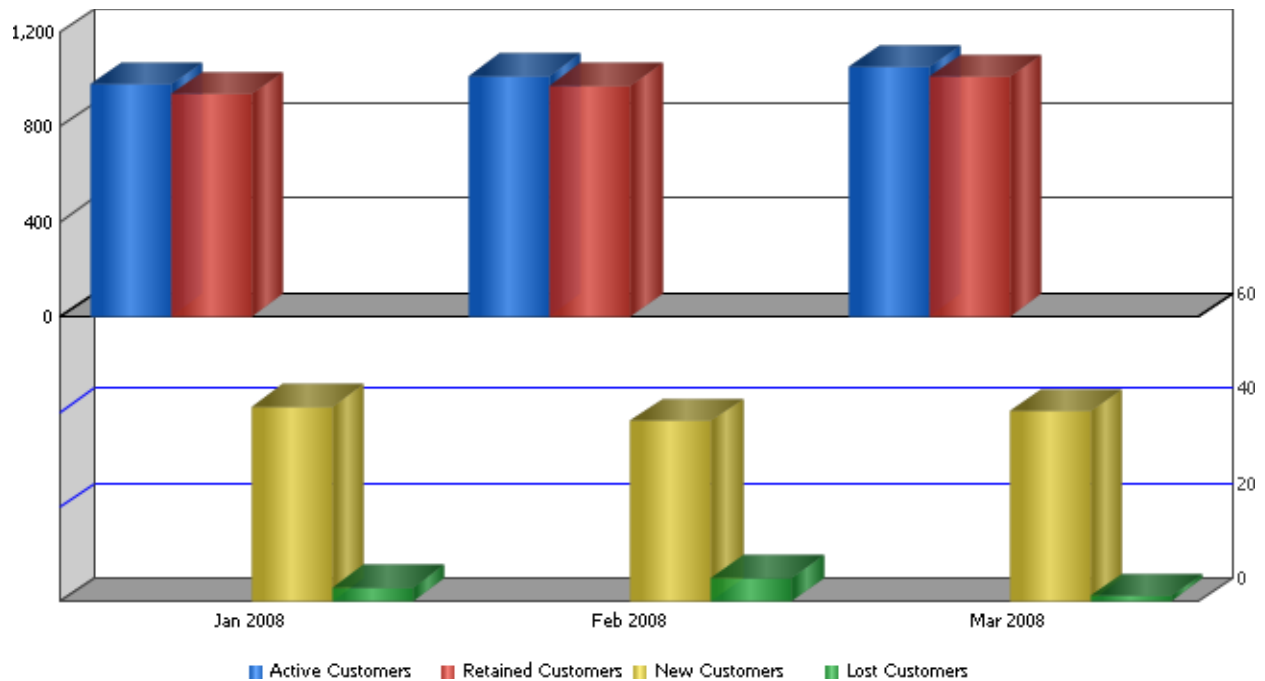
more attractive and usable. The grid report and resulting graph report are shown below.

	Metrics	Active Customers	Retained Customers	New Customers	Lost Customers
Month					
Jan 2008		977	936	41	3
Feb 2008		1,010	972	38	5
Mar 2008		1,049	1,009	40	1



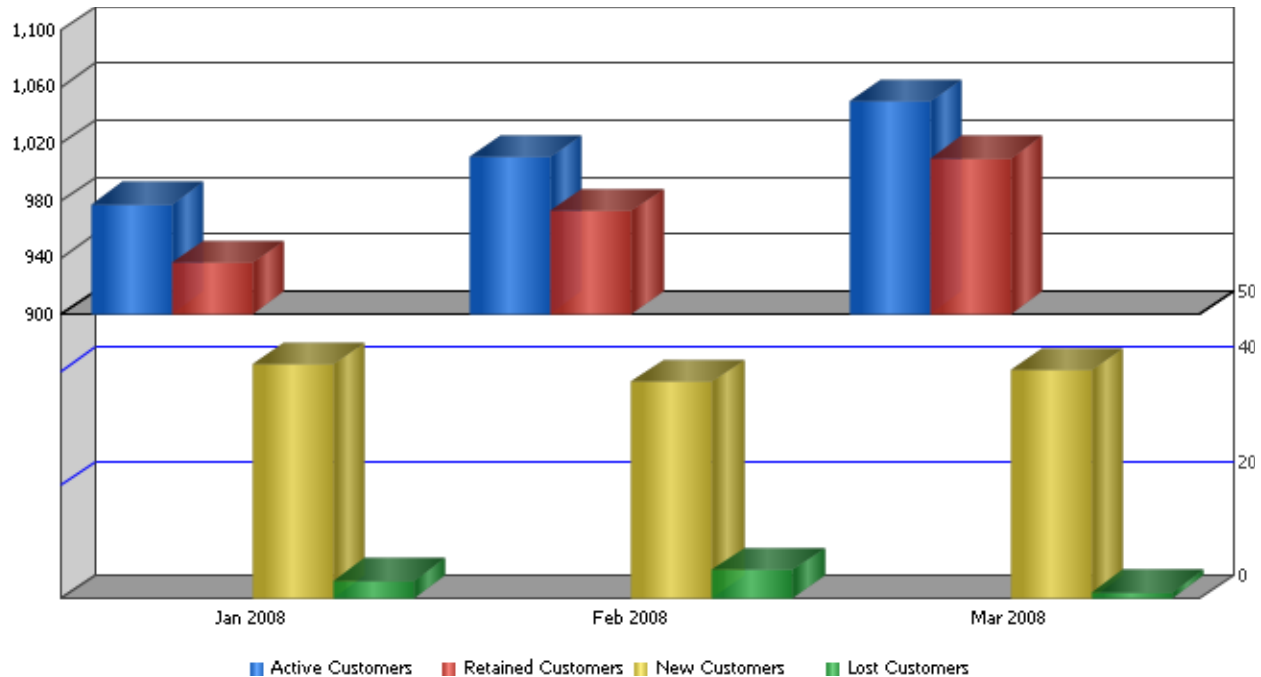
While the automatic scaling displays the active and retained customer data effectively, notice that the new and lost customer data is difficult to interpret from this graph. Another option to improve the scaling of a graph is to use a dual Y-axis. These two axes can then be used to display metrics with similar values when the relative scale of some metric data is far different than the scale of other metric data on the graph. For example, the same graph is displayed below with a dual Y-axis that displays Active Customers and

Retained Customers on one Y-axis, and New Customers and Lost Customers on another Y-axis.



Notice in the graph above that by putting the metrics on separate Y-axes, the scale is improved for the data and all data is easier to interpret and analyze. To further adjust the scale of the graph, you can change the top Y-axis to have a maximum of 1,100 and a minimum of 900. Additionally, you can change

the bottom Y-axis to have a maximum of 50. This modified graph is shown below.



Notice how defining a minimum of 900 for the top graph provides more clarity to the Active Customers and Retained Customers metric values. It is now much easier to see the differences in active and retained customers for each month. This shows that it is just as important to define a good minimum value for a graph as it is to define a good maximum value.

Also, defining a slightly smaller maximum of 50 for the bottom graph has provided a small improvement to the display of data.

The procedure below shows you how to manually define the scale of your graph. Steps are provided to adjust scaling in both MicroStrategy Developer and MicroStrategy Web.

Scaling a graph in MicroStrategy Developer

Prerequisites

- A graph report has been created.
- You are logged in to a project using MicroStrategy Developer.

To manually scale a graph

- 1 In MicroStrategy Developer, view a report in Graph View.
- 2 Select one of the data labels along any of the axes on your graph.
- 3 From the **Graph** menu, select **Grids and Scales**. The Preferences dialog box opens.
- 4 Underneath the **Axes** category, select one of the available Y-axis categories.

If the graph uses a dual Y-axis you can define the minimum and maximum values for the two Y-axes. To define a graph to use a dual Y-axis, see [Defining a graph to use a dual Y-axis in MicroStrategy Developer, page 494](#).

If the graph uses three-dimensional formatting, you can define the minimum and maximum values for the X-axis and the Y-axis.

- 5 In the **Use Manual Setting** area, select the **Maximum Value** and **Minimum Value** check boxes. You can then type in the maximum and minimum values to define for the Y-axis of the graph.



If you define a manual maximum and minimum value for a graph axis, the axis is not recalculated when modifying the objects or filtering for a graph. This can cause the axis to not adequately represent the graph data when the data displayed for the graph changes. If the data on a graph changes, you can clear the **Maximum Value** and **Minimum Value** check boxes and click **Apply** to allow the graph to automatically recalculate the size of the axis.

You should avoid defining manual maximum and minimum values for graphs that are frequently modified such as graphs that use prompts.

- 6 If the graph uses a dual Y-axis you can also define the minimum and maximum values for the second Y-axis. If the graph uses three-dimensional formatting, you can also define the minimum and maximum values for the X-axis.
- 7 Click **Apply** to accept and apply your changes.
- 8 Click **OK** to return to the graph.

Defining a graph to use a dual Y-axis in MicroStrategy Developer

The procedure below show you how to define a graph to use a dual Y-axis, which can improve the scale of the graph to match the available data.

Prerequisites

- A graph report has been created. It must be displayed as one of the following graph types which can support a dual Y-axis:
 - Area
 - Bar
 - Line
 - Scatter
 - Polar
 - Radar
 - Bubble
- Two or more objects (commonly metrics) are included in the series of the graph report.
- You are logged in to a project using MicroStrategy Developer.

To use a dual Y-axis in MicroStrategy Developer

- 1 In MicroStrategy Developer, view a report in Graph View.
- 2 From the **Graph** menu, select **Graph Options**. The Preferences dialog box opens.
- 3 Select the **Layout** category.
- 4 Select the **Dual Axis** check box. This displays each Y-axis overlaid on top of the other, with each axis using a different color.
- 5 If you want each Y-axis to be displayed separately, select the **Split Dual Axis** check box. Displaying a dual Y-axis in this manner can make the graph easier to analyze and interpret.
- 6 Select the **Dual Y Options** category To define which Y-axis each series object (commonly metrics) is displayed on.

- 7 Select series objects in the **Primary Axis** or **Secondary Axis** panes and use the arrow button to change the axis the series object is displayed on. You should include series objects on the same axis if they have similar data ranges. For example, review the reports described in [Scaling a graph, page 489](#) which use one Y-axis for metrics that have values between 900-1,000 and use a second Y-axis for metrics that have values between 0-50.



Graphs created in MicroStrategy 9 and later versions distribute series objects evenly between the Primary Axis and Secondary Axis by default. Graphs updated from pre-9 MicroStrategy releases may use a different default dual Y-axis order for the series objects. These default orders can be changed using the arrow buttons available.

- 8 If you selected to split each Y-axis you can also define the following options:
 - **Split Position:** Defines what percentage of the displayed graph is taken up by each axis. Use the slider to see a sample of how the display changes on the left.
 - **Show Axis Split:** Displays or hides the dual axes split line. This provides additional visual separation between each axis.
- 9 Click **Apply** to accept and apply your changes.
- 10 Click **OK** to return to the graph.

Scaling a graph in MicroStrategy Web

You can set whether Y-axis values display on the graph and, if so, what minimum and maximum values to use on the scale. You can also specify the interval or steps to use.

To scale a graph report in MicroStrategy Web

- 1 Click the name of a report to execute it. Display the report in either Graph View or Grid and Graph View.
- 2 From the **Format** menu, select **Graph**. The Format: Graph dialog box opens.

- 3 On the **Axes** tab, select from the following options. For details about each option, click **Help**.
 - To specify the scaling for the Y1-axis, type a maximum, minimum, and grid interval in the **Y1 Axis Values - Use Manual Setting** area.
 - To specify the scaling for the Y2-axis, type a maximum, minimum, and grid interval in the **Y2 Axis Values - Use Manual Setting** area.
 - To adjust the position of the Y1 and Y2 axes, change the settings in the **Dual Y Options** area.
- 4 Click **OK** to apply the changes to the graph.

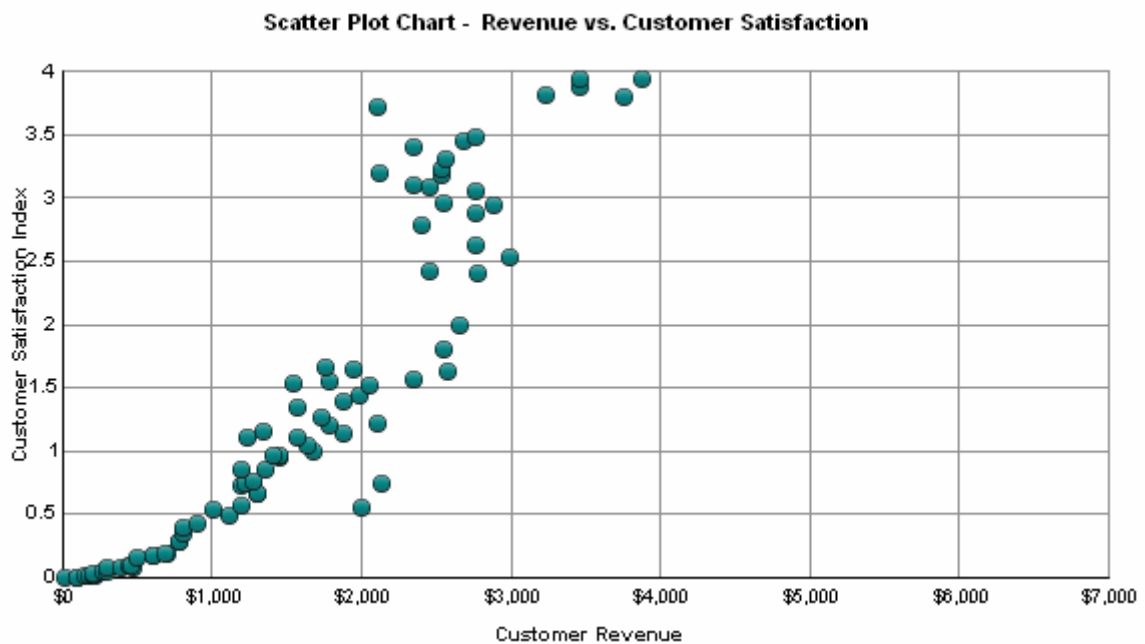
Displaying off-scale values in a graph

MicroStrategy Developer allows you to specify a range of values for which to display items in a graph. However, some graph items may fall outside a given range. When this occurs, you can use the Off-scale Value Display drop-down list to change how they are displayed. There is a separate setting for each axis in a graph.

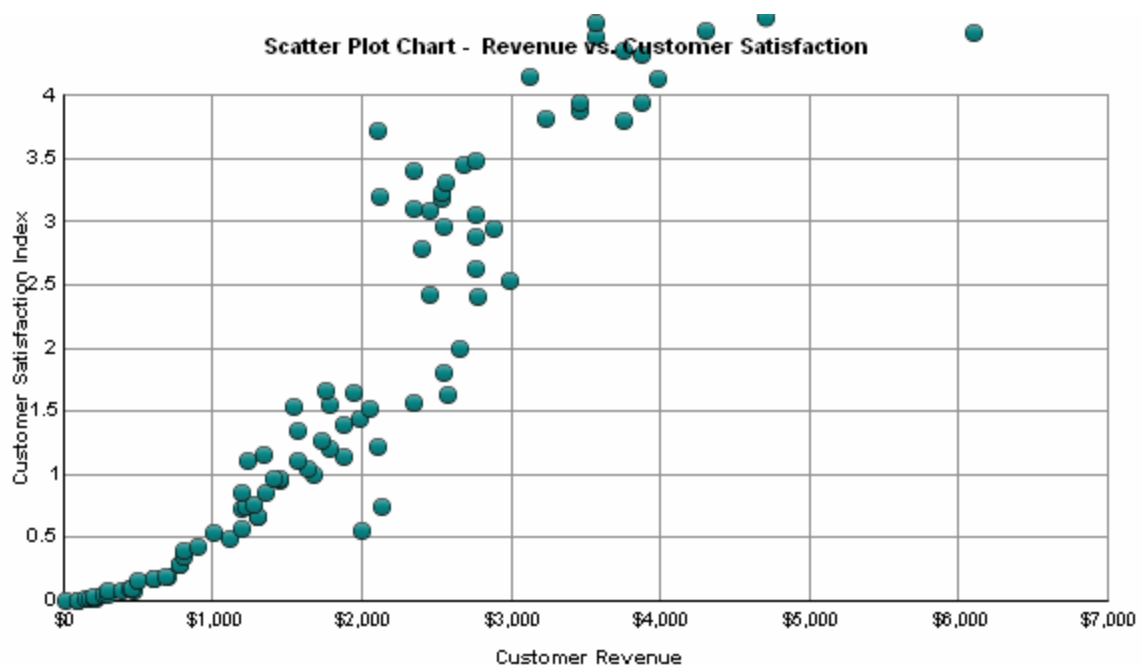
The display options are:

- **Hide Values:** This option hides all elements which fall outside the specified minimum or maximum range for a graph. In the graph below,

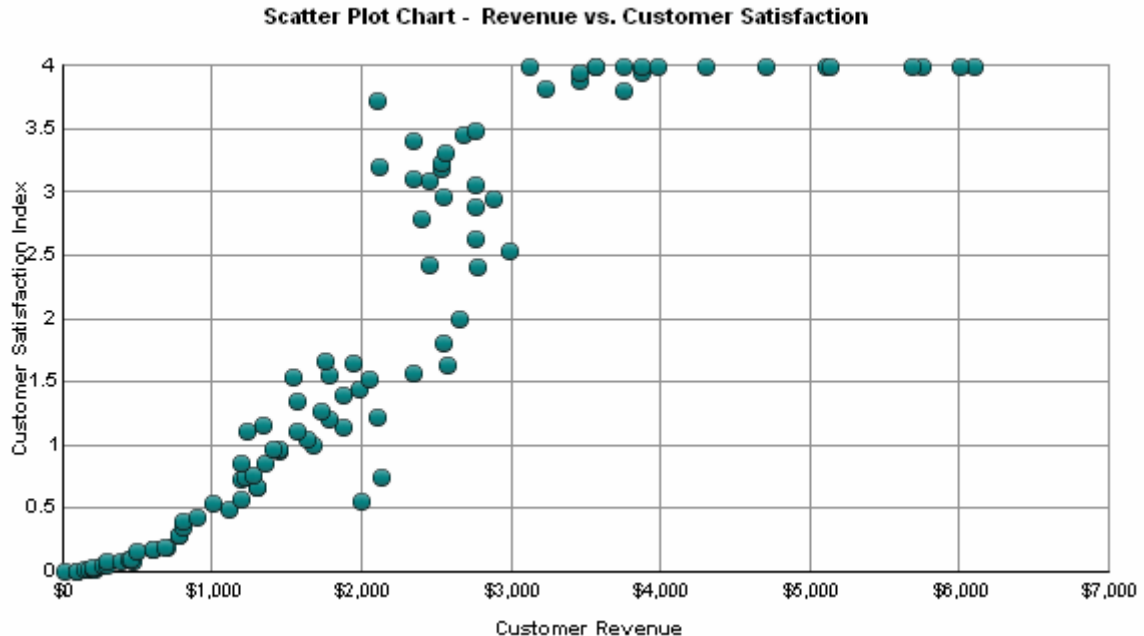
all data points representing a Customer Satisfaction Index of more than four have been removed from view.



- Show Values:** This option displays all elements which fall outside of the specified range. These elements continue off the specified axes and are only constrained by the size of the graph area. In the graph below, data points with a Customer Satisfaction Index of more than four are shown continuing off the top of the graph.



- **Clip Values:** This option shows elements that fall outside the specified range, but forces them to fit into the graph by clipping them to the side. In the graph below, data points with a Customer Satisfaction Index of more than four can be seen clipped to the top of the graph.



The procedure below details how to specify an off-scale display option for a graph.

Prerequisites

- You are logged in to a project using MicroStrategy Developer.
- A graph report has been created.
- A range of values has been specified for at least one axis. For more information on defining ranges for graphs, see [Scaling a graph, page 489](#).
- The graph includes graph elements which fall outside of the specified range. These steps can be completed without elements outside of the graph range, but the changes will not be visible.

To change the display of off-scale values

- 1 In MicroStrategy Developer, view a report in Graph View.

- 2 On the **Graph** menu, select **Grids and Scales**. The Preferences dialog box opens.
- 3 Expand the **Axes** category, and then expand the axis for which to modify the display of off-scale values.
- 4 Select the **Grids and Scales** category. The Grids and Scales options are displayed.
- 5 From the **Off-scale Value Display** drop-down list, select one of the following options:
 - **Hide Values**
 - **Show Values**
 - **Clip Values**
- 6 Click **Apply** to apply changes.
- 7 Click **OK** to exit the Preferences dialog box.
- 8 From the **File** menu, select **Save** to save the display change.

Saving and applying a custom graph style

After a graph has been created, you can save its formatting settings as a template file to apply to other graphs later. The template file is saved with a .3TF extension.

Suppose you format a graph report in a certain way and want other graph reports to have the same look and feel as that graph report. For example, you format a Bar graph so the bars are green and bevelled, while the data labels are colored orange. You can name and save this formatted report as a template file. The template file is essentially a new graph style.

When you apply the new graph style to other graphs, the graphs are automatically formatted according to your custom style. For example, if your custom style's formatting includes Pie charts in an Impact color scheme, the graph's formatting is updated to include Pie charts in that specific color scheme.

Applying custom graph styles to reports is especially convenient if you want several graph reports in your project to have the same look and feel. Instead

of manually adjusting the formatting elements of each graph, you can apply the same custom graph style to the reports.

The following procedures guide you in saving and then applying a custom graph style to a report.

For an example of how custom graph styles can be saved and applied, see [Example: Saving and applying a custom graph style, page 501](#).

To save a custom graph style

- 1 Select the report to use as a style template and display the report in Graph View.
- 2 From the **Graph** menu, select **Save Graph Style**. The Save Graph Style dialog box opens.
- 3 Select the name of an existing file or type a new name in the **File name** field. Notice that the file extension must be .3TF.
- 4 Click **Save** to save the graph's formatting as a separate template file.

Follow the procedure below to apply this graph style to another graph.



You cannot access the custom graph styles you create from the Graph Styles dialog box. You must follow the steps in this procedure to access and apply any custom graph styles.

To apply a custom graph style to a graph

- 1 Select the report to which you want to apply the custom graph style, and display the report in Graph View.
- 2 From the **Graph** menu, select **Apply Graph Style**. The Apply Graph Style dialog box opens.
- 3 Navigate to the custom graph style of your choice. The extension for custom graph styles is .3TF.
- 4 Select the file and click **Open**. Your graph is updated with the formatting that is part of the graph style template you selected.

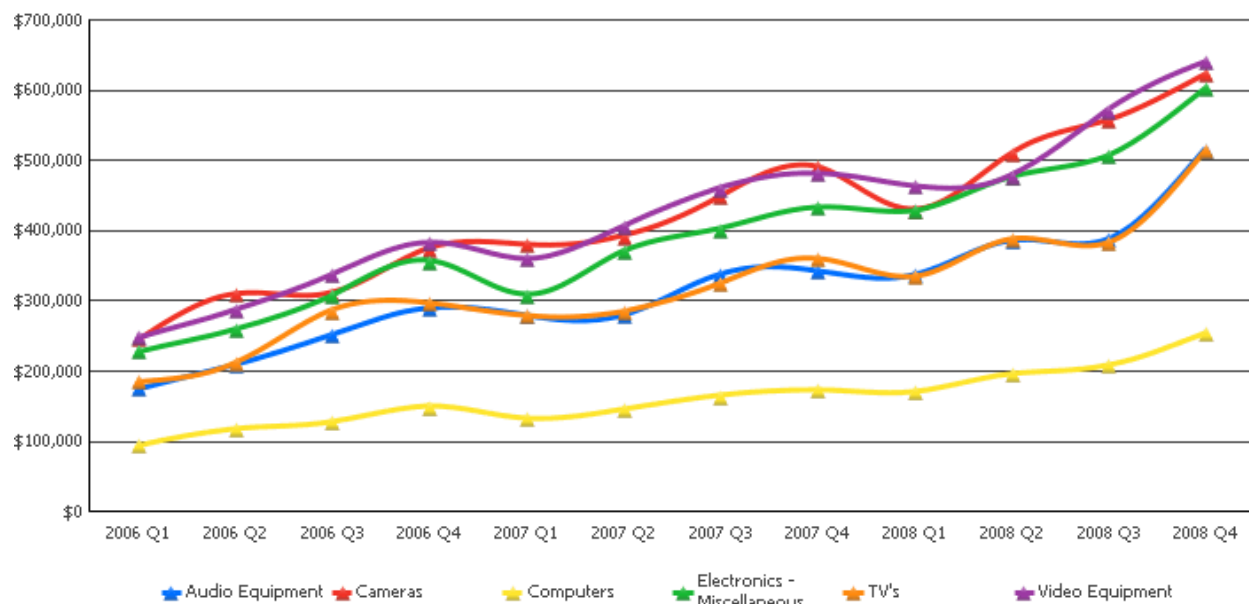
Example: Saving and applying a custom graph style

The following example demonstrates how you can save a formatted graph as a new, custom graph style, and then apply the new graph style to another graph.

Notice the following formatting qualities of the graph report below:

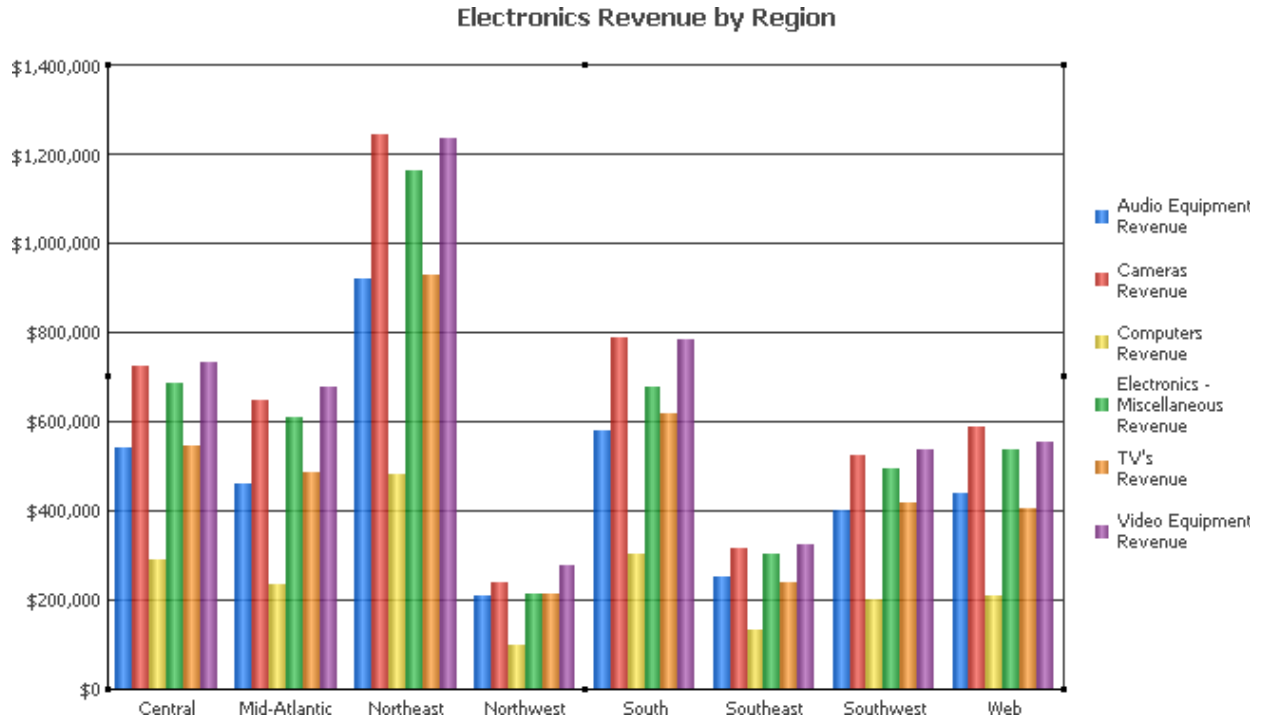
- The large font of the title
- The Line graph style that uses different colors
- Markers are displayed as small triangles for each data point
- The legend appears below the graph rather than on the right side
- The legend labels do not include the metric name Revenue by selecting the **Suppress last level labels** option in the Graph Preferences dialog box within the Options: General category

Electronics Revenue Over Time

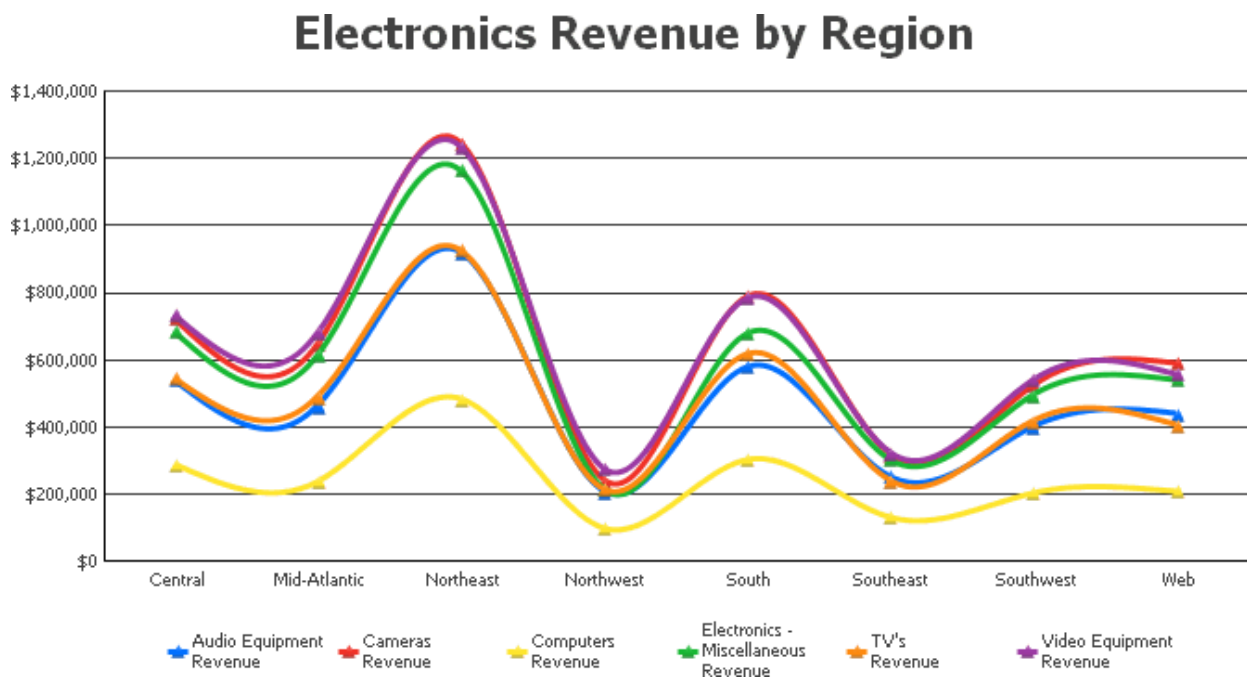


You save this specific graph formatting as a new graph style. For the steps to take to save graph formatting as a new graph style, see [Saving and applying a custom graph style, page 499](#).

Now that you have saved the new graph style, you are considering applying the style to the following graph report.



When you apply the custom graph style to the report above, the report's formatting changes according to your custom graph style, as shown in the image below.



The overall formatting of the Sales by Region report was updated according to the custom graph style you created from another report.

Choosing a graph image format for HTML documents and Web

When you create and save a graph in MicroStrategy Developer, you should consider whether users will access the graph in MicroStrategy Web, for example, in a report or HTML document. The image format assigned to a graph report directly affects how the graph is executed in graphs and HTML documents in MicroStrategy Web.



For background information on HTML documents, see the *MicroStrategy Developer help*.

The format you choose affects the quality of the graph and how quickly it is displayed in MicroStrategy Web, as described in the procedure below. For example, when a graph is saved as a GIF, it is loaded quickly in MicroStrategy Web, but does not present color gradients and other visual effects well.

You can determine the image file format in which to save the graphs in your graph reports in MicroStrategy Developer.

To select an image file format for a graph report in MicroStrategy Developer

- 1 Select the graph to modify and open it in Graph View in MicroStrategy Developer.
- 2 From the **Graph** menu, select **Preferences**. The Preferences dialog box opens.
- 3 Under **Options**, select **Graph Type**.
- 4 From the **Show graph in format** drop-down list, select an image format. The following section lists each image format and provides information on whether or not the format is recommended for use in HTML documents or MicroStrategy Web:
 - GIF: Produces the smallest file sizes and therefore allows for the fastest report execution. However, it is not recommended for HTML

documents and MicroStrategy Web because it may not display color gradients and some other color elements correctly.

- **PNG:** Produces the second largest file sizes. While it is generally recommended for HTML documents and MicroStrategy Web, it is not recommended for any graphs that include color gradients because gradients are not displayed correctly.
- **JPEG:** Produces the largest file sizes. It is recommended for HTML documents and MicroStrategy Web because it allows color features such as gradients to be displayed correctly.
- **BMP:** Generally not recommended for HTML documents and MicroStrategy Web because it may not display color gradients and some other color elements correctly.

5 Click **OK**. Save the graph report to apply the image format changes to it.

Exporting and sharing graphs

In addition to viewing graphs in MicroStrategy Developer or Web, graphs can also be exported as various file types or copied into third-party applications such as Microsoft PowerPoint or Excel. This allows you to provide graphs to your user community in various formats and file types.

Exporting graphs

You can export a MicroStrategy graph into file types such as HTML, PDF, Microsoft Word document, and Microsoft Excel document. This allows you to share MicroStrategy graphs as various file types that meet the needs of your user community. For example, you can export a graph to a PDF to share the graph with a user community that does not have access to MicroStrategy Developer or Web.

The procedure below shows you how to export a MicroStrategy graph.

To export MicroStrategy graphs

1 In MicroStrategy Developer or Web, log in to a MicroStrategy project.

- 2 Browse to a MicroStrategy graph and execute it as a graph.
- 3 Export the MicroStrategy graph to the desired file type, using one of the following procedures:
 - In MicroStrategy Developer, from the **Data** menu, point to **Export To**, and select from the following export options:
 - **MS Excel**
 - **MS Word**
 - **HTML File**
 - **PDF**

The MicroStrategy graph is created as the selected file type.

- In Web, from the **Home** menu, click **Export**, and select from the following export options:
 - **PDF**
 - **Excel with formatting**
 - **HTML**

If export options are displayed, select your desired export options and formatting. For information on the available export options, click **Help**.

Click **Export**. The MicroStrategy graph is created as the selected file type.

Copying graphs into third-party applications

You can copy MicroStrategy graphs from MicroStrategy Developer directly into third-party applications. This allows you to share MicroStrategy graphs through other applications that meet the needs of your user community. For example, while viewing a graph in MicroStrategy Developer, you can copy the graph display and paste it into a Microsoft PowerPoint presentation.

You can also copy graphs into other Microsoft Office applications such as Microsoft Word and Excel, as well as other document-authoring applications.



With MicroStrategy Office, you can fully integrate MicroStrategy reports and documents into your Microsoft Office applications.

MicroStrategy reports and documents can then be refreshed with the most recent data available in your MicroStrategy environment. For information on using and configuring MicroStrategy Office, see the *Office User Guide*.

The procedure below shows you how to copy a MicroStrategy graph into third-party applications.

To copy MicroStrategy graphs into third-party applications

- 1 In MicroStrategy Developer, log in to a MicroStrategy project.
- 2 Browse to a report, right-click the report and select **Run**. The Report Editor opens.
- 3 If the report is not displayed as a graph, from the **View** menu, select **Graph View**.
- 4 Modify and format the graph to fit your needs. When copying a graph, only the information that is currently displayed is copied. If data is on another page of the graph, you must copy that page separately or define the graph to fit all data on one page. For information on displaying data on a single page for a graph, see [Displaying graph data without scrollbars, page 380](#).
- 5 From the **Edit** menu, select **Copy**.
- 6 Open the third-party application to include the MicroStrategy graph in such as Microsoft PowerPoint or Excel.
- 7 With the application open, select an area that allows images to be displayed such as a PowerPoint slide or a cell in an Excel spreadsheet. Perform the action required to paste the MicroStrategy graph into the third-party application. A common technique to perform this action is to right-click the area to copy the MicroStrategy graph to, and select **Paste**. The MicroStrategy graph is included in the third-party application.

Using interactive graph visualizations in documents: Widgets

If you are interested in creating and using interactive, Flash-based graph visualizations in Report Services documents, you may want to consider using widgets in the documents. A widget is a type of Report Services control that presents data in a visual and interactive way. You can think of widgets as interactive Flash-only graphs that dynamically update when you select a new set of data to view. You can even use some types of widgets to manually select a set of data to analyze. There are a number of widget types available, such as the Gauge or Interactive Stacked Area widget. Although each type of widget looks different and is used in a unique way, the main purpose of widgets is the same: to provide document analysts with a visual and interactive representation of their data.

Widgets are not available in reports; they are available only in Report Services documents. For more information, including steps to add widgets to your Report Services documents, see the *Report Services Document Creation Guide*.

DISPLAYING INFORMATION ABOUT REPORTS: REPORT DETAILS

Introduction

You can determine what information appears in the Report Details pane for all reports viewed on your machine. You can configure the following report details:

- Filter details, which display the report filter and report limit by default, although other types of filters can be displayed
- Report limit details, which display the report limit
- Prompt details, which display the prompt information for all prompts in the report
- Report details, which display the complete report details, including report description, prompt details, filter details, and template details
- Template details, which display the complete template details, including attribute and metric details

The following image displays the report description, report filter, and report limits information in the Report Details pane in a report:

Report details						
Report Description: This report shows Revenues vs. Forecasted Revenues. Users can select the region, category, and time period they are interested in viewing. Thresholds in this report are built on the variance between Revenue and Forecast.						
Report Filter (Local Filter): (Category = Electronics) And (Quarter = 2006 Q1,2006 Q2,2006 Q3 or 2006 Q4) And (Region = Northeast or Mid-Atlantic)						
Report Limits: Filter is empty.						
Cache Used: No						
1 2 3						
Region	Category	Subcategory	Quarter Metrics	2006 Q1 Revenue Forecast	2006 Q1 Revenue	2006 Q2 Revenue Forecast
Northeast						
	Electronics					
		Audio Equipment		\$ 22,032	\$20,400	\$ 26,497
		Cameras		\$ 34,110	\$34,110	\$ 84,784
		Computers		\$ 8,501	\$7,523	\$ 9,491
		Electronics - Miscellaneous		\$ 8,830	\$9,598	\$ 15,532
		TV's		\$ 12,925	\$11,644	\$ 24,674
		Video Equipment		\$ 31,046	\$32,680	\$ 45,626
Mid-Atlantic						
	Electronics					
		Audio Equipment		\$ 22,935	\$25,770	\$ 54,798

You can configure different options for different types of report details. For example, you can select whether to include view filter information or the attribute name in the filter details. For report details, you can choose whether to include information on prompts or filters. For complete descriptions of all the report detail options, see [Configuring report details settings, page 513](#).

Displaying report details

The report details are displayed in the Report Details pane, which can be displayed when you open a report. By default, report details are enabled so that the Report Details pane is automatically displayed when you open a report. You can disable report details so that they do not display when the report is opened. For steps to enable or disable report details, see [Enabling and disabling report details, page 511](#).

Regardless of this setting, report details can be hidden or viewed after a report is opened.

To display or hide report details in a report

- 1 Open a report. You can edit it or execute it.
- 2 From the **View** menu, select **Report Details**.
 - If the Report Details pane was hidden, it appears at the top of the report.
 - If the Report Details pane was displayed, it is hidden.

Enabling and disabling report details

By default, report details are enabled so that you can view them in any report. You can disable report details so that they do not display when the report is opened. If you have disabled them, you can enable them again.

To enable or disable report details

- 1 In MicroStrategy Developer, from the **Tools** menu, select **Preferences**. The Developer Preferences dialog box opens.
- 2 Click the **Report** category to open it, and then click **Report Details**.
- 3 The **Show report details** check box governs whether the report details are displayed automatically:
 - To enable report details, select the check box.
 - To disable report details, clear the check box.
- 4 Click **OK** to save your changes.

Levels of report details configuration

You can configure:

- All the auto text codes in a specific text field in a document, from the Properties dialog box (the text field level)

- All the auto text codes in a specific document, from the Document Properties dialog box (the document level)



For information on configuring auto text codes in documents, see the *Report Services Document Creation Guide*.

- The report details in the Report Editor, from the Report Details Formatting option (the report level)
- All the report details in a project, using the Project Configuration Editor (the project level)

The list above also shows the order of precedence. The configuration of a particular text field in a document overrides the configuration at the document level, which overrides the configuration in the related dataset, which overrides the project configuration. For example, if an option is set one way for the project and another for the report, the report setting takes precedence.

The report details settings at the report and project levels are the same; the only differences are where you configure the settings and what they affect. For complete descriptions of all the report detail options, see [Configuring report details settings, page 513](#).

Interaction of report details preferences and report details for reports

The report details preferences specified in the Developer Preferences dialog box specify which types of report details are displayed. These preferences override some of the report details that are configured in the Report Details Properties dialog box in the Project Configuration Editor and the Report Editor. For example, you choose prompt details as a report details preference. When you open a prompted report, the name of each prompt and its prompt answers are displayed in the Report Details pane. You cannot remove the prompt details display from that report, unless you remove prompt details as a report details preference or disable report details preferences. In the report itself, however, you can configure how the prompt details are displayed, such as whether the title of the prompt is displayed.

The report details preferences specify whether the following types of report details are displayed:

- Report description
- Prompt details

- Filter details
- Definition of shortcut filters, as opposed to the filter name
- View filter details
- Template/metric details

You can change this behavior by disabling the report details preferences, which allows the report details settings to be used instead. You can disable the report details preferences when you save changes to the report details settings or by using the Developer Preferences dialog box. Steps to disable them using the Developer Preferences dialog box are below. For more information about the report details preferences in general, see [Enabling and customizing report details preferences, page 554](#).



The report details preferences do not affect auto text codes in documents; the preferences only impact reports.

To disable report details preferences

- 1 From the **Tools** menu in MicroStrategy Developer, select **Preferences**. The Developer Preferences dialog box opens.
- 2 Click the **Report** category to open it, and then click **Report Details**.
- 3 To use the report details settings instead of the report details preferences, clear the **Apply these Developer preferences to the Report Details** check box.
- 4 Click **OK** to save your changes.

Configuring report details settings

You can configure the following types of report details:

- **Filter details**, which display the report filter and report limit by default, although other types of filters can be displayed.
- **Report limit details**, which display the report limit.
- **Prompt details**, which display the prompt information for all prompts in the report.

- **Report details**, which display the complete report details, including report description, prompt details, filter details, and template details.
- **Template details**, which display the complete template details, including attribute details and metric details.

You can configure different options for different types of report details. For example, you can select whether to include view filter information or the attribute name in the filter details. For report details, you can choose whether to include information on prompts or filters. For a complete list of the report detail settings, see:

- [*Configuring the display of object name delimiters for report details, page 514*](#)
- [*Configuring report details, page 516*](#)
- [*Configuring the display of template details, page 520*](#)
- [*Configuring the prompt details, page 530*](#)
- [*Configuring the filter details, page 537*](#)

Configuring the display of object name delimiters for report details

Delimiters are the characters around attribute and metric names which set them off from other text. Braces { } are used as delimiters. You can apply the setting **Use delimiters around metadata object names** to achieve any of the following in the display of report details:

- Display delimiters for all metadata objects
- Omit delimiters for all metadata objects
- Display delimiters only for those objects that contain a special character (**Automatic**)



Special characters are characters other than a - z, A - Z, 0 - 9, #, _, and . (period).

For example, the following report filter shown in the Report Details pane displays delimiters:

Report details			
Report Filter: (Revenue) > 3000000 And (Region) (DESC) Begins with "South"			
Region	Metrics	Revenue	Profit
South		\$5,389,280	\$806,956
Southwest		\$3,694,132	\$561,331

You can also choose to inherit the setting instead, as described in [Levels of report details configuration, page 511](#).

To configure the display of object name delimiters for a project

- 1 In MicroStrategy Developer, right-click the project that you want to work with, and select **Project Configuration**. The Project Configuration Editor opens.
- 2 In the **Categories** list, expand **Project definition**, and then select **Document and Reports**.
- 3 Click **Report details properties**. The Report Details Properties dialog box opens to the General area.
- 4 From the **Use delimiters around report object names** drop-down list, select one of the following:
 - **No** to omit delimiters for all report objects
 - **Yes** to display delimiters for all report objects
 - **Automatic** to display delimiters only for those objects that contain a special character
- 5 Click **OK** to return to the Project Configuration Editor.

To configure the display of object name delimiters for a report

- 1 Open the report. You can either edit it or execute it.

- 2 From the **Data** menu, select **Report Details Properties**.
- 3 Click **General**.
- 4 From the **Use delimiters around metadata object names** drop-down menu, select one of the following options:
 - **No** to omit delimiters for all metadata objects
 - **Yes** to display delimiters for all metadata objects
 - **Automatic** to display delimiters only for those objects that contain a special character

 By default, this setting inherits the project setting, as described in [Levels of report details configuration, page 511](#). You can return to this default by selecting **<default>** from the drop-down list.
- 5 Click **OK** to return to the report.
- 6 If report details preferences are enabled, a message opens, indicating that some of your changes may not be applied to the report. Object name delimiters are not affected by report details preferences, so click **No** to keep report details preferences enabled. For more information about report details preferences in general, see [Enabling and customizing report details preferences, page 554](#).

Configuring report details

Report details display information about the report, including the report description, prompt details, filter details, and template details.

You can choose whether to include or omit the following:

- **Report Description** (the short description of the report)
- **Prompt Details** (the prompts on the report)
- **Filter Details** (the report filter, view filter, and report limits)
- **Template Details** (the objects on the report and the metric definitions)

You can also choose to inherit the setting instead, as described in [Levels of report details configuration, page 511](#).

Report details: Examples

The following report sample shows the report details, as well as a portion of the executed report. Notice that the report description, report filter, report limits, view filter, and template information are displayed. The report limits information is blank because the report does not include a report limit.

Report details

Report Description:Revenue, profit, & cost by region and employee

Report Filter:Books: Category=Books

Report Limits:Empty Filter

View Filter:Revenue > 75000

Template:

Region

Employee

Metrics:

Revenue

Revenue

{~+}

Cost

Sum(Cost)

{~+}

Profit

Sum(Profit)

{~+}

Region	Employee		Metrics	Revenue	Cost	Profit
Central	Gale	Loren		\$124,966	\$97,939	\$27,027
	Torrison	Mary		\$126,804	\$99,392	\$27,412
Mid-Atlantic	Bernstein	Lawrence		\$76,358	\$60,036	\$16,322
	Folks	Adrienne		\$82,076	\$64,368	\$17,708
	Hollywood	Robert		\$77,154	\$60,461	\$16,693

The template information may not be valuable to you, but you do need to know how the report is filtered, how the metrics are calculated, and which employees are included on the report. In addition, the report description is not very informational in this case. The image below shows the same report,

but the report details have been configured to display only the information that you need.

Report details						
Report Filter:Books: Category=Books						
Report Limits:Empty Filter						
View Filter:Revenue > 75000						
Region	Employee		Metrics	Revenue	Cost	Profit
Central	Gale	Loren		\$124,966	\$97,939	\$27,027
	Torrison	Mary		\$126,804	\$99,392	\$27,412
Mid-Atlantic	Bernstein	Lawrence		\$76,358	\$60,036	\$16,322
	Folks	Adrienne		\$82,076	\$64,368	\$17,708
	Hollywood	Robert		\$77,154	\$60,461	\$16,693

Interaction of report details preferences and report details for reports

To allow report details settings to be used instead of report details preferences, ensure that the report details preferences are disabled. For steps to set report details preferences, see [Enabling and customizing report details preferences, page 554](#).

If you do not disable the report details preferences, they override some of the report details settings that affect the content of the report details. These can include the following:

- Report description
- Prompt details
- Filter details
- Template/metric details

To configure the report details for a project

- 1 Disable the report details preferences, if you have not already done so. For steps, see [To disable report details preferences, page 513](#).

- 2 In MicroStrategy Developer, right-click the project that you want to work with, and select **Project Configuration**. The Project Configuration Editor opens.
- 3 In the **Categories** list, expand **Project definition**, and then select **Document and Reports**.
- 4 Click **Report details properties**. The Report Details Properties dialog box opens.
- 5 To include the report's short description, select the **Report Description** check box. Clear the check box if you do not want to include the short description.
- 6 To include details of prompts, select the **Prompt Details** check box. Clear the check box if you do not want to include prompts.
- 7 To include details of the report's filter, any view filter, and report limits, select the **Filter Details** check box. Clear the check box if you do not want to include these.
- 8 To include object and metric definitions, select the **Template Details** check box. Clear the check box if you do not want to see these.

To configure the report details for a report

- 1 Disable the report details preferences, if you have not already done so. For steps, see [To disable report details preferences, page 513](#).
- 2 Open the report. You can either edit it or execute it.
- 3 From the **Data** menu, select **Report Details Properties**.
- 4 Click the **General** tab.



By default, each setting on this tab inherits the project setting, as described in [Levels of report details configuration, page 511](#). You can return to this default by selecting **<default>** from the drop-down list.
- 5 To determine whether to include the report's short description, select **Yes** or **No** from the **Report Description** drop-down list.
- 6 To determine whether to include the prompts, select **Yes** or **No** from the **Prompt Details** drop-down list.

- 7 To determine whether to include the report filter, view filter, and report limits, select **Yes** or **No** from the **Filter Details** drop-down list.
- 8 To determine whether to include the objects and the metric definitions, select **Yes** or **No** from the **Template Details** drop-down list.
- 9 Click **OK** to close the Report Details Properties dialog box.
- 10 If report details preferences are enabled, a message opens, indicating that some of your changes may not be applied to the report, as described in [Interaction of report details preferences and report details for reports, page 518](#).
 - To disable report details preferences, click **Yes**. The changes that you made affect the display of report details in this report, and report details preferences are disabled for all reports.
 - To keep report details preferences enabled, click **No**. The report details preferences override the changes that you made.

Configuring the display of template details

The template of a report contains:

- The group of objects (attribute, metrics, custom groups, and so on) that defines the columns of data to be included in the report
- The layout and format of those objects

The template details in the Report Details pane display information about the objects on the report, but do not include layout and format information.

To configure the display of template details, you can:

- Choose whether to include the report objects from the base report or from the view.

A view report is created when you move objects from the grid of a report to the Report Objects, use a view filter, or create a derived metric. The report before you made these kinds of modifications is referred to as the base report. The template details can show all the objects on the report, regardless of whether they are on the report grid, if you select **Base**. If you select **View**, only those objects on the report grid are included in the template details. See [Including objects from the base or view report, page 521](#) for a more detailed description and examples.

- Determine whether the template name of the report is displayed. If you choose to display the template name, you can also select whether or not to identify an embedded template as a local template. An embedded template is one that is part of the report, as opposed to a template saved as a stand-alone object. See [Displaying the base template name, page 523](#) for a more detailed description and examples.
- Determine whether the short description of the template is included. If the template is embedded or does not have a description, the template description line is not displayed.
- Determine whether report objects other than metrics (such as attributes and consolidations) are listed.
- Determine whether the metrics on the report are listed.
- If metrics are displayed, select whether or not to display each of the following:
 - Conditional metrics only
 - Metric formulas
 - Metric dimensionality
 - Metric conditionality
 - Metric transformation



If you choose to inherit whether metrics are displayed, these settings are also inherited.

For steps to configure the template details, see [To configure the template details for a report, page 528](#).

Including objects from the base or view report

You create a view report if you do any of the following:

- Move objects from the grid of a report to the Report Objects
- Use a view filter
- Create a derived metric

The report before you made these kinds of modifications is referred to as the base report. The template details can show all the objects on the report, regardless of whether they are on the report grid, if you select **Base**. If you

select **View**, only those objects on the report grid are included in the template details.

Base or view report setting: Examples

For example, a report contains the metrics Revenue, Cost, and Profit, but Profit is not on the report grid. Profit is only displayed in the Report Objects.

- If you set **Units from View or Base** to **Base**, all three metrics show in the Report Details pane, as shown below:

Report objects		Report details	
Name	Type	Template:	
 Region	Attribute	Region	
 Cost	Metric	Metrics:	
 Profit	Metric	Revenue	
 Revenue	Metric	Revenue	
		{~+}	
		Cost	
		Sum(Cost)	
		{~+}	
		Profit	
		Sum(Profit)	
		{~+}	

Region	Metrics	Revenue	Cost
Central		\$5,029,366	\$4,265,043
Mid-Atlantic		\$4,452,615	\$3,779,531
Northeast		\$8,554,415	\$7,253,683
Northwest		\$1,761,187	\$1,494,202
South		\$5,389,280	\$4,582,324
Southeast		\$2,239,951	\$1,903,276
Southwest		\$3,694,132	\$3,132,800
Web		\$3,902,762	\$3,319,225

- If you set **Units from View or Base** to **View**, only Revenue and Cost are shown in the Report Details pane, as shown below:

Report objects		Report details	
Name	Type		
Region	Attribute	Template:	
Cost	Metric	Region	
Profit	Metric	Metrics:	
Revenue	Metric	Revenue	
		Revenue	
		{~+}	
		Cost	
		Sum(Cost)	
		{~+}	
		Metrics	Revenue
Region			Cost
Central			\$5,029,366 \$4,265,043
Mid-Atlantic			\$4,452,615 \$3,779,531
Northeast			\$8,554,415 \$7,253,683
Northwest			\$1,761,187 \$1,494,202
South			\$5,389,280 \$4,582,324
Southeast			\$2,239,951 \$1,903,276
Southwest			\$3,694,132 \$3,132,800
Web			\$3,902,762 \$3,319,225

Displaying the base template name

When you save a report that you created using a stand-alone template, you can either:

- Retain the shortcut to the template, so that changes made to the template are propagated to the report.
- Create a local copy of the template, which embeds the template in the report. Changes made to the template are not propagated to the report.

You can use the **Base Template name** setting to indicate whether or not a report uses a stand-alone template, and to display a stand-alone template's name.

- To determine whether a report uses a stand-alone template, set the **Base Template name** setting to **Automatic**.
 - A stand-alone template displays the template's name.
 - An embedded template does not display a name.
 - A report that does not use a template does not display a name.

- To always display a template name, set the **Base Template name** option to **Yes**.
 - A stand-alone template displays the template's name.
 - An embedded template displays as Local Template.
 - A report that does not use a template displays as Local Template.
- To hide the template name, set the **Base Template name** option to **No**. Nothing is displayed.

Base template name setting: Examples

For example, create a template containing the Region attribute and the Revenue metric. Save it as Regional Revenue Template. Create a filter for the Books category. Create a report with the Regional Revenue Template and the Books filter. Save the report as Template + Filter. When prompted, choose to retain the shortcut to the template, so that the template is not embedded in the report.

In the Report Details Properties dialog box, display template details and set **Base Template name** to **Automatic**. Because the template is a stand-alone template, its name is displayed in the Report Details pane, as shown below:

Report details		
Template (Regional Revenue Template):		
Region		
Metrics:		
Revenue		
Revenue		
{~+}		

Region	Metrics	Revenue
Central		\$376,836
Mid-Atlantic		\$337,656
Northeast		\$646,421
Northwest		\$129,175
South		\$406,110
Southeast		\$170,445
Southwest		\$280,796
Web		\$292,655

Save the report as Embedded Template + Filter. When prompted, create a local copy of the template. The template is now embedded in the report. If you change the Regional Revenue Template, the Template + Filter report changes accordingly but the Embedded Template + Filter report does not.

Close and re-open the report. Since the template is embedded and no longer stand-alone, it no longer has a name, as shown below.

Report details		
Template:		
Region		
Metrics:		
Revenue		
Revenue		
{~+}		
Region	Metrics	Revenue
Central		\$376,836
Mid-Atlantic		\$337,656
Northeast		\$646,421
Northwest		\$129,175
South		\$406,110
Southeast		\$170,445
Southwest		\$280,796
Web		\$292,655

Change the **Base Template name** setting to **Yes**, which forces a template name to be displayed. Since the embedded template does not have a name, it is displayed as a local template, as shown below:

Report details		
Template (Local Template):		
Region		
Metrics:		
Revenue		
Revenue		
{~+}		
Region	Metrics	Revenue
Central		\$376,836
Mid-Atlantic		\$337,656
Northeast		\$646,421
Northwest		\$129,175
South		\$406,110
Southeast		\$170,445
Southwest		\$280,796
Web		\$292,655

Create a third report. Do not use the template, but add Region, the Revenue metric, and the Books filter to it. In the Report Details Properties dialog box, display template details and set **Base Template name** to **Automatic**. Because no template was used to create the report, the template name is blank. Change **Base Template name** to **Yes**. The template name is displayed as Local Template.

You can also change the **Base Template name** setting to **No** so that the template name is not displayed, regardless of whether the report has an embedded template, a stand-alone template, or no template.

Interaction of report details preferences and template details for reports

To allow report details settings to be used instead of report details preferences, ensure that the report details preferences are disabled. For more information on report details preferences, see [Enabling and customizing report details preferences, page 554](#).

If you do not disable the report details preferences, they override some of the report details settings that display the report details. These include whether the template/metric details are displayed.

To configure the template details for a project

- 1 Disable the report details preferences, if you have not already done so. For steps, see [To disable report details preferences, page 513](#).
- 2 In MicroStrategy Developer, right-click the project that you want to work with, and select **Project Configuration**. The Project Configuration Editor opens.
- 3 In the **Categories** list, expand **Project definition**, and then select **Document and Reports**.
- 4 Click **Report details and properties**. The Report Details Properties dialog box opens.
- 5 Select the **Template Details** check box.
- 6 On the left, select **Template Details**.
- 7 To select whether the base or view report is used, choose one of the following from the **Units from View or Base** drop-down list:
 - To include the report objects from the base report (which includes all the objects that make up the report's definition, regardless of whether they are on the report grid), select **Base**.

- To include the report objects from the view report (which includes only the objects on the report grid), select **View**.

See [Including objects from the base or view report, page 521](#) for a detailed description and examples.

- 8 To select what is displayed for the template name, choose one of the following from the **Base Template name** drop-down list:
 - To display the template name for a stand-alone template and “Local Template” for other reports, select **Yes**.
 - To display the template name for a stand-alone template, but leave it blank for other reports, select **Automatic**.
 - To omit the template name for all types of reports, choose **No**.

See [Displaying the base template name, page 523](#) for a detailed description and examples.

- 9 To include the short description of the template, select the **Template Description** check box. Clear the check box if you do not want to include the short description.
- 10 To include descriptions of report objects other than metrics (such as attributes, consolidations, and custom groups), select the **Non_Metric template units** check box. Clear the check box if you do not want to include these.
- 11 To include metrics, select the **Metrics** check box. Clear the check box if you do not want to include metrics.
- 12 If metrics are displayed, select or clear the check box for each of the following, depending on whether you want to display them:
 - **Conditional Metrics only**
 - **Formula**
 - **Dimensionality**
 - **Conditionality**
 - **Transformation**



If you choose to inherit whether metrics are displayed, these settings are also inherited.

- 13 Click **OK** to close the Report Details Properties dialog box.

To configure the template details for a report

- 1 Disable the report details preferences, if you have not already done so. For steps, see [To disable report details preferences, page 513](#).
- 2 Open the report. You can either edit it or execute it.
- 3 From the **Data** menu, select **Report Details Properties**.
- 4 Click the **General** tab.
- 5 Ensure that template details are displayed by setting **Template Details** to **Yes**.
- 6 Click the **Template Details** tab.



By default, each setting on this tab inherits the project setting, as described in [Levels of report details configuration, page 511](#). You can return to this default by selecting **<default>** from the drop-down list.

- 7 To select whether the base or view report is used, choose one of the following from the **Units from View or Base** drop-down list:
 - To include the report objects from the base report, which includes all the objects in the report's definition, regardless of whether they are on the report grid, select **Base**.
 - To include the report objects from the view report, which includes only the objects on the report grid, select **View**.

See [Including objects from the base or view report, page 521](#) for a more detailed description and examples.

- 8 To select what is displayed for the template name, choose one of the following from the **Base Template name** drop-down list:
 - To display the template name for a stand-alone template and "Local Template" for other reports, select **Yes**.
 - To display the template name for a stand-alone template, but leave it blank for other reports, select **Automatic**.
 - To omit the template name for all types of reports, choose **No**.

See [Displaying the base template name, page 523](#) for a more detailed description and examples.

- 9 To determine whether or not the short description of the template is included, select **Yes** or **No** from the **Template Description** drop-down list.



If the template is embedded or does not have a description, the template description line is not displayed.

- 10 To determine whether or not report objects other than metrics (such as attributes, consolidations, and custom groups) are listed, select **Yes** or **No** from the **Non-metric template units** drop-down list.
- 11 To determine whether or not the metrics on the report are listed, select **Yes** or **No** from the **Metrics** drop-down list.

- 12 If metrics are displayed, select whether or not to display each of the following:

- **Only Conditional Metrics**
- **Formula**
- **Dimensionality**
- **Conditionality**
- **Transformation**



If you choose to inherit whether metrics are displayed, these settings are also inherited.

- 13 Click **OK** to close the Report Details Properties dialog box.
- 14 If report details preferences are enabled, a message opens, indicating that some of your changes may not be applied to the report, as described in [Interaction of report details preferences and template details for reports, page 526](#).
 - To disable report details preferences, click **Yes**. The changes that you made affect the display of template details in this report, and report details preferences are disabled for all reports.
 - To keep report details preferences enabled, click **No**. The report details preferences override the changes that you made.

Configuring the prompt details

The prompt details display the prompt information for all prompts in the report. You can configure:

- Whether the prompt title and index (a number indicating the order of the prompts in the report) are displayed.
- The text to display when a prompt is unanswered. The options are:
 - Display nothing
 - Display “Prompt Not Answered”
 - Display “No Selection”
 - Display “All/None”

Whether the word “All” or “None” displays depends on the type of prompt. For example, an unanswered object prompt displays as “None”, because no objects are selected. An unanswered filter definition prompt displays as “All” because the report is not filtered and therefore all the objects appear on the report.

- Whether and how to display the attribute name for the attribute element list prompts in the report. The options are:
 - Display the attribute name (for example, Region)
 - Omit the attribute name
 - Repeat the attribute name for each prompt answer (for example, Region = North, Region = South)



The browse form of the attribute, which is displayed when a user answers the prompt, is used to display the attribute elements in the prompt details. For information on browse forms, see the *Project Design Guide*.

- Whether to include unused prompts. An unused prompt occurs when you drill on a report that contains a prompt. The resulting child report can display or omit the prompt details from the parent.

Prompt details: Examples

The Regional Revenue by Category report contains the Category and Region attributes, and the Revenue metric. It is prompted for Category and Region. The prompt details are configured to display the prompt titles and index. The prompt title is specified when the prompt is created (Category prompt and Region prompt in the example below), and the index is a number indicating the order of the prompts in the report (Prompt 1 and Prompt 2 below).

Report details			
Prompt 1: Category prompt			
Books			
Prompt 2: Region prompt			
Northwest, Northeast			
Category	Region	Metrics	Revenue
Books	Northeast		\$646,421
	Northwest		\$129,175

If you change the prompt details to omit the prompt titles and index, only the prompt answers are displayed, as shown below:

Report details			
Books			
Northwest, Northeast			
Category	Region	Metrics	Revenue
Books	Northeast		\$646,421
	Northwest		\$129,175

Both the Category and Region prompts are attribute element list prompts, so you can also specify whether and how to display the attribute name. In the example above, the attribute names are not displayed; only the selected attribute elements are listed. In the example below, the attribute names (Category and Region) are displayed.

Report details			
Category=Books			
Region=Northwest, Region=Northeast			
Category	Region	Metrics	Revenue
Books	Northeast		\$646,421
	Northwest		\$129,175

You can select whether to display the attribute name once for each prompt, or, as shown above, repeated for each prompt answer.

The Category and Region prompts were answered in the previous examples, but sometimes users do not answer all the prompts. You can choose what to display when a prompt is not answered. You can select pre-defined text, or you can choose to display nothing at all. The pre-defined text includes:

- Prompt Not Answered
- No Selection
- All/None

Whether the word “All” or “None” displays depends on the type of prompt. For example, an unanswered object prompt displays as “None”, because no objects are selected. An unanswered filter definition prompt displays as “All” because the report is not filtered and therefore all the objects appear on the report.

The following report sample is the same report as before, except that the Region prompt has been answered and the Category prompt has not. The prompt details for the Category prompt display the word “All”, meaning that all the categories are included on the report.

Report details			
Prompt 1: Category prompt			
All			
Prompt 2: Region prompt			
Region=Northwest, Region=Northeast			
Category	Region	Metrics	Revenue
Books	Northeast		\$646,421
	Northwest		\$129,175
Electronics	Northeast		\$5,962,709
	Northwest		\$1,234,850
Movies	Northeast		\$1,001,561
	Northwest		\$200,894
Music	Northeast		\$943,724
	Northwest		\$196,269

Finally, you can choose whether to include unused prompts. An unused prompt occurs when you drill on a report that contains a prompt. The resulting child report can display or omit the prompt details from the parent.

For example, drill down from Region to Call Center on the previous report. The drilled-to report, as shown below, displays the prompt details of the drilled-from report (the original report). The unused prompts are included.

If the original report instead did not include unused prompts, the Report Details pane would be blank.

Report details				
Prompt 1: Category prompt All				
Prompt 2: Region prompt Region=Northwest, Region=Northeast				
Category	Region	Call Center	Metrics	Revenue
Books	Northeast	Boston		\$112,962
		New York		\$533,459
	Northwest	San Francisco		\$74,735
		Seattle		\$54,440
Electronics	Northeast	Boston		\$1,034,067
		New York		\$4,928,642
	Northwest	San Francisco		\$715,269
		Seattle		\$519,581
Movies	Northeast	Boston		\$177,096
		New York		\$824,466
	Northwest	San Francisco		\$116,542
		Seattle		\$84,352
Music	Northeast	Boston		\$163,812
		New York		\$779,912
	Northwest	San Francisco		\$114,901
		Seattle		\$81,368

Interaction of report details preferences and prompt details for reports

Ensure that the report details preferences are disabled, which allows the report details settings to be used instead. For more information on report details preferences, see [Enabling and customizing report details preferences, page 554](#).



You can disable the report details preferences when you save changes to the report details settings (as described in the following procedure) or by using the Developer Preferences dialog box (for steps, see [To disable report details preferences, page 513](#)).

If you do not disable the report details preferences, they override some of the report details settings that deal with the content of the report details. These include whether the prompt details are displayed.

If the report details preferences specify that the prompt details are not displayed, you cannot display them using the Report Details Properties dialog box. If the report details preferences specify that the prompt details are displayed, you cannot hide them using the Report Details Properties dialog box. You can use the Report Details Properties dialog box to configure how they are displayed, however.

To configure the prompt details for a project

- 1 Disable the report details preferences, if you have not already done so. For steps, see [To disable report details preferences, page 513](#).
- 2 In MicroStrategy Developer, right-click the project that you want to work with, and select **Project Configuration**. The Project Configuration Editor opens.
- 3 In the **Categories** list, expand **Project definition**, and then select **Document and Reports**.
- 4 Click **Report details and properties**. The Report Details Properties dialog box opens.
- 5 Select the **Prompt Details** check box.
- 6 To determine whether or not to include the prompt titles and index, select **Title and Index** or **No Title or Index** from the **Include Prompt Titles** drop-down list.

The prompt title is defined when the prompt is created, and the index is a number indicating the order of the prompts in the report.

- 7 To specify the text to display when a prompt is unanswered, select one of the following from the **Replacement string for unanswered prompts** drop-down list:
 - **Default**
 - **Blank**
 - **Prompt Not Answered**
 - **No Selection**
 - **All/None**

- 8 To specify whether and how to display the attribute name for any attribute element list prompts in the report, select one of the following options from the **Show attribute name for Attribute Element Prompts** drop-down list:
 - To show the attribute names, select **Yes**.
 - To omit the attribute names, select **No**.
 - To repeat the attribute name for each prompt answer, select **Repeated**.
- 9 Select the **Include unused prompts** check box to include unused prompts. Clear the check box if you do not want to include unused prompts. An unused prompt occurs when you drill on a report that contains a prompt. The drilled-to report can display or omit the prompt details from the drilled-from report.
- 10 Click **OK** to close the Report Details Properties dialog box.

To configure the prompt details for a report

- 1 Disable the report details preferences, if you have not already done so. For steps, see [To disable report details preferences, page 513](#).
- 2 Open the report. You can either edit it or execute it.
- 3 From the **Data** menu, select **Report Details Properties**.
- 4 Click the **General** tab.
- 5 Ensure that prompt details are displayed by setting **Prompt Details** to **Yes**.



By default, each setting on this tab inherits the project setting, as described in [Levels of report details configuration, page 511](#). You can return to this default by selecting **<default>** from the drop-down list.

- 6 To determine whether or not to include the prompt titles and index, select **Title and Index** or **No Title or Index** from the **Include Prompt Titles** drop-down list.

The prompt title is defined when the prompt is created, and the index is a number indicating the order of the prompts in the report.

- 7 To specify the text to display when a prompt is unanswered, select one of the following from the **Replacement string for unanswered prompts** drop-down list:
 - **Blank**
 - **Prompt Not Answered**
 - **No Selection**
 - **All/None**
- 8 To specify whether and how to display the attribute name for any attribute element list prompts in the report, select one of the following options from the **Show attribute name for Attribute Element Prompts** drop-down list:
 - To show the attribute names, select **Yes**.
 - To omit the attribute names, select **No**.
 - To repeat the attribute name for each prompt answer, select **Repeated**.
- 9 To specify whether or to not include unused prompts, select **Yes** or **No** from the **Include unused prompts** drop-down list. An unused prompt occurs when you drill on a report that contains a prompt. The resulting child report can display or omit the prompt details from the parent.
- 10 Click **OK** to close the Report Details Properties dialog box.
- 11 If report details preferences are enabled, a message opens, indicating that some of your changes may not be applied to the report, as described in [Interaction of report details preferences and prompt details for reports, page 533](#).
 - To disable report details preferences, click **Yes**. The changes that you made affect the display of prompt details in this report, and report details preferences are disabled for all reports.
 - To keep report details preferences enabled, click **No**. The report details preferences override the changes that you made.

Configuring the filter details



This section assumes that you are familiar with filters, especially the different types of filters, stand-alone vs. embedded filters, and the components of filters. For background information, see the *Basic Reporting Guide*.

The filter details display information about the report filter and report limit used in the report by default, although other types of filters can be displayed. The filter details have many configuration settings, which can be grouped into the following categories. This list provides a summary of the settings.

- You can configure the content of the filter details, by selecting which filter types to include (report filter, view filter, report limits, and so on) and how the filters are displayed. For example, you can select whether to include the name of the filter type and whether to display the report limits before or after view filters. See [Configuring the content of the filter details, page 538](#) for descriptions of all these options.
- For filters that contain attribute element list qualifications, you can configure how those lists are displayed. For example, you can specify whether to show the attribute name (such as Region or Category) and the separator between attribute names. See [Configuring how attribute element list qualifications are displayed, page 544](#) for descriptions of all these options.
- You can configure how attribute form and set qualifications in filters are displayed. For example, you can select whether to use names or symbols for the operators. See [Configuring how attribute form and set qualifications are displayed, page 546](#) for descriptions of all these options.
- You can configure how logical operators, which join multiple qualifications or filters, are displayed. For example, you can configure whether to include parentheses around conditions. See [Configuring how logical operators are displayed, page 548](#) for descriptions of all these options.
- You can configure whether aliases replace object names in the filter details. See [Configuring whether to use aliases in the filter details, page 551](#) for a detailed description of this option.

For the procedure to configure the filter details, see [Configuring the display of filter details: Procedure, page 551](#).

Interaction of report details preferences and filter details for reports

Ensure that the report details preferences are disabled, which allows the report details settings to be used instead. For more information on report details preferences, see [Enabling and customizing report details preferences, page 554](#).



You can disable the report details preferences when you save changes to the report details settings (as described in [Configuring the display of filter details: Procedure, page 551](#)) or by using the Developer Preferences dialog box (for steps, see [To disable report details preferences, page 513](#)).

If you do not disable the report details preferences, they override some of the report details settings that deal with the content of the report details. These include whether the filter details are displayed.

If the report details preferences specify that the filter details are not displayed, you cannot display them using the Report Details Properties dialog box. If the report details preferences specify that the filter details are displayed, you cannot hide them using the Report Details Properties dialog box. You can use the Report Details Properties dialog box to configure how they are displayed, however.

Configuring the content of the filter details

You can choose whether or not to include each type of filter. The filter types are listed below:

- **Report Filter**

If the report filter is displayed, you can select whether or not to display the following:

- **Report Filter Name.** You can also select another option, **Automatic**. If **Automatic** is chosen, the report filter name is displayed for a stand-alone filter, but is omitted for an embedded filter.
- **Report Filter Description.**

- **Report Limits**

- **View Filter**

If the view filter is displayed, you can select whether or not to display the following:

▫ **Metric Qualification in the View Filter**

• **Drill Filter**

When the report is drilled on, determines whether or not the attribute element(s) drilled on are displayed in the report details of the drill report. For examples, see [Drill filter: Examples, page 542](#).

• **Security Filter**

For each option, you can also choose to inherit the setting instead, as described in [Levels of report details configuration, page 511](#).



Steps to configure the filter details are provided in the *MicroStrategy Developer help*.

Filter types: Examples

The following report displays some of the different filter types.

Report details				
Report Filter: Category=Books, Movies				
Report Limits: Empty Filter				
View Filter: Empty Filter				
Security Filter: Region=Northeast, Mid-Atlantic, Southeast				
Category	Region	Call Center	Metrics	Revenue
Books	Mid-Atlantic	Washington, DC		\$235,588
		Charleston		\$102,068
	Northeast	Boston		\$112,962
		New York		\$533,459
	Southeast	Atlanta		\$80,516
		Miami		\$89,929
Movies	Mid-Atlantic	Washington, DC		\$363,602
		Charleston		\$155,367
	Northeast	Boston		\$177,096
		New York		\$824,466
	Southeast	Atlanta		\$122,223
		Miami		\$142,063

The report filter is embedded, and the **Report Filter Name** option is set to **Automatic**, therefore no name is displayed for the report filter. If the **Report Filter Name** option was set to **Yes** instead, the report details would display as:

```
Report Filter (Local Filter)
```

```
Category = Books, Movies
```

If the report filter was a stand-alone filter (not a filter embedded in that report), the name displays if the **Report Filter Name** setting is set to **Automatic** or **Yes**. For example, if the report was filtered by the stand-alone filter named Southern Regions, the report details display as:

```
Report Filter (Southern Regions)
```

```
Region = South, Southeast, Southwest
```

The content of the filter types: Examples

You can configure how the filters are displayed in the filter details by specifying the following settings:

- Whether to include the names of the filter types, such as Report Filter, View Filter, and so on.

In the report sample above, the names are displayed. If the names are omitted, the filter details display as:

```
(Local Filter): Category=Books, Movies
```

```
Region=Northeast, Mid-Atlantic, Southeast
```

- If the filter type name is included, specify whether to include empty expressions. An empty expression is a filter type that is not included on the report.

For example, the report sample above lists report limits and view filter, even though the report does not contain either one. The filter details could instead display only those filter types that the report contains, such as:

```
Report Filter (Local Filter):
```

```
Category = Books, Movies
```

```
Security Filter:
```

```
Region=Northeast, Mid-Atlantic, Southeast
```

- Whether to add a new line after each filter type name, before the actual definition of the filter.

For example, the report sample above includes a new line after each filter type name. If the line is omitted, the filter details display as:

```
Report Filter (Local Filter): Category= Books, Movies
Security Filter: Region=Northeast, Mid-Atlantic,
Southeast
```

- Whether to add a new line between the different filter types to help differentiate between them.

For example, the report sample above includes the new line. If the line is omitted, the results are:

```
Report Filter (Local Filter): Category= Books, Movies
Security Filter: Region= Northeast, Mid-Atlantic,
Southeast
```



The report sample above does not have a new line between the filter types or after the filter type name.

- Whether to show the report limits before or after the view filter.
- Whether to expand shortcut filters, which are stand-alone filters used in the report filter. The options are:
 - **Show filter name**
 - **Show filter definition**
 - **Show name and definition**

In all cases, the definition of embedded filters is displayed. (An embedded, or local filter, does not have a name.)

For example, Filter 1 is defined as Region = Northeast. A report contains the Category and Region attributes, as well as the Revenue metric. A local filter (also called an embedded filter) is created in the report, filtering on Books. Filter 1 is added to the report filter. If the filter name (Filter 1 in this case) is shown, the filter details look like the following:

```
Report Filter:
(Category = Books) And {Filter 1}
```

If the filter definition (Region = Northeast in this case) is shown, the filter details look like the following:

Report Filter:

(Category = Books) And (Region = Northeast)

If both are shown, the filter details look like the following:

Report Filter:

(Category = Books) And ({Filter 1}: Region= Northeast)

Drill filter: Examples

The drill filter is the attribute element(s) that you selected when you drilled on the report. The drill filter is displayed in the drilled-to report.

For example, a report contains the Category and Region attributes, as well as the Revenue and Profit metrics. A portion of the report is shown below. Notice that the Report Details pane is blank—although filter details have been included, the only kind of filter that is displayed is the drill filter. Since this report is not a drill report, nothing is displayed.

Report details				
Category	Region	Metrics	Revenue	Profit
Books	Central		\$276,826	\$94,234
	Mid-Atlanti	Drill		Other directions
	Northeast	Attribute Forms		Down
	Northwest		\$129,175	Up
	South		\$406,110	\$86,806
	Southeast		\$170,445	\$36,619
	Southwest		\$280,796	\$60,697
Electronics	Web		\$292,655	\$63,070
	Central		\$3,506,062	\$619,714
	Mid-Atlantic		\$3,106,940	\$545,693
	Northeast		\$5,962,709	\$1,053,305
	Northwest		\$1,234,850	\$217,311

Drill on the report from Books/Central down to Call Center, as shown above. Now the Report Details pane displays the drill filter information, that Books and Central were selected when the drill was performed.

Report details					
Drill Filter: (Category=Books) And (Region=Central)					
Category	Region	Call Center	Metrics	Revenue	Profit
Books	Central	Milwaukee		\$313,177	\$67,702
		Fargo		\$63,659	\$13,629

If the drill filter was turned off, the Report Details pane would be blank in the drill report.

If you drill from an attribute (as opposed to selecting elements), a drill filter is not created. The report is not filtered; it is just displayed at a different level. For instance, select the Region attribute instead of the Central region (an element of the Region attribute) in the original report, as shown below.

Report details					
Category	Region	Metrics	Revenue	Profit	
Books	Central	Drill			
	Mid-A	Sort			
	North	Attribute Forms			
	North	Move			
	South	Remove from Grid			
	South	Formatting			
	Web	Lock Object			
	Central	Edit... Ctrl+E			
	Mid-A	Remove from Report			
	North	Properties			
Electronics	North	Edit Links...			
	South				
	Southeast		\$1,352,007	\$271,833	
	Southwest		\$2,562,060	\$453,973	

The resulting drill report is displayed below. The report is the same as the original, but at the Call Center level rather than Region. All regions and all categories are still displayed, unlike the first drill report that was filtered for Books and Central. In other words, the report is not filtered, so the drill filter

details are not displayed and the Report Details pane remains blank, because a drill filter was not created.

Report details					
Category	Region	Call Center	Metrics	Revenue	Profit
Books	Central	Milwaukee		\$313,177	\$67,702
		Fargo		\$63,659	\$13,629
	Mid-Atlantic	Washington, DC		\$235,588	\$50,723
		Charleston		\$102,068	\$22,070
	Northeast	Boston		\$112,962	\$24,519
		New York		\$533,459	\$115,413
	Northwest	San Francisco		\$74,735	\$16,310
		Seattle		\$54,440	\$11,720
	South	New Orleans		\$249,057	\$53,804
		Memphis		\$157,053	\$33,002
	Southeast	Atlanta		\$80,516	\$17,285
		Miami		\$89,929	\$19,334
	Southwest	San Diego		\$225,515	\$48,756
		Salt Lake City		\$55,282	\$11,941
	Web	Web		\$292,655	\$63,070
Electronics	Central	Milwaukee		\$2,916,573	\$516,822
		Fargo		\$589,489	\$102,892
	Mid-Atlantic	Washington, DC		\$2,193,709	\$384,658
		Charleston		\$913,231	\$161,035
	Northeast	Boston		\$1,034,067	\$180,972
		New York		\$4,928,642	\$872,333
	Northwest	San Francisco		\$715,269	\$127,092
		Seattle		\$519,581	\$90,219

Configuring how attribute element list qualifications are displayed

A filter can contain an attribute element list qualification, which qualifies on a list of attribute elements. For example, you can use an attribute element list qualification on the attribute Customer, in a report, to return data only for those customers that you specify in your list. For more information on attribute element list qualifications, see the *Basic Reporting Guide*.

For these filters, you can configure how the lists are displayed using the following settings. Examples are included below the list, on [page 545](#).

- **Show attribute name for In List conditions:** Determines whether or not the name of the attribute in the filter's attribute element list is displayed. The name can also be repeated for each attribute element (for example, Region = Northeast, Region = Mid-Atlantic).
- **Separator after attribute name:** Specifies the characters that separate the attribute name from the attribute element.



In Developer, clear the **Inherit** check box to enable the **Separator after attribute name** field.

- **New line after attribute name:** Determines whether or not the attribute name and its element display on separate lines.
- **Separator between last two elements:** Specifies the text that separates the last two attribute elements in the list. The choices are:
 - **or**
 - **and**
 - **comma** (the character is used, not the text)
 - **custom** (in the **Custom separator** field, type the characters to use as the separator)
- **New line between elements:** Determines whether or not each attribute element displays on a separate line.
- **Trim elements:** Determines whether extra spaces in the attribute elements are deleted. For example, an element of an account attribute is PSI2415 : 10 : COMMERCIAL. If **Trim elements** is enabled, the attribute is displayed as PSI2415:10:COMMERCIAL, omitting the extra spaces.

You can also choose to inherit the setting instead, as described in [Levels of report details configuration, page 511](#).

In the report sample below:

- **Show attribute name for In List conditions** = Yes
- **New line after attribute name** = Yes
- **Separator between last two elements** = custom
- **Custom separator** = , and (note that a space is included after the word “and”)

- **New line between elements** = No

Report details		
Report Filter (East Coast): Region = Northeast, Mid-Atlantic, and Southeast		
Region	Metrics	Revenue
Mid-Atlantic		\$4,452,615
Northeast		\$8,554,415
Southeast		\$2,239,951

In the following report sample:

- **Show attribute name for In List conditions** = Repeated
- **New line after attribute name** = No
- **Separator between last two elements** = comma
- **New line between elements** = Yes

Report details		
Report Filter (East Coast): Region = Northeast Region = Mid-Atlantic Region = Southeast		
Region	Metrics	Revenue
Mid-Atlantic		\$4,452,615
Northeast		\$8,554,415
Southeast		\$2,239,951

Configuring how attribute form and set qualifications are displayed

A filter can contain:

- An attribute form qualification, which is based on attribute forms (such as First Name and Last Name for the Customer attribute)
- A metric set qualification, which is based on metric value or rank
- A relationship set qualification, which is based on relationships between attributes

For more information on these types of filter qualifications, see the *Basic Reporting Guide*.

For these types of filters, you can configure how the qualifications are displayed using the following settings:

- **Use names or symbols for operators:** Determines whether names (such as Equals or Greater Than) or symbols (such as = or >) are displayed.

For example, the filter details displayed below use a symbol to indicate the operator:

```
Revenue > 6000000
```

The filter details displayed below use a name:

```
Revenue Greater than 6000000
```

- **Include attribute form names in qualification conditions:** For conditions in attribute qualification filters, determines whether or not to display attribute form names (such as DESC or ID).

For example, the filter details displayed below include the attribute form name (DESC):

```
(Region (DESC) Begins with "North")
```

These filter details do not display the attribute form name:

```
(Region Begins with "North")
```

- **Dynamic dates:** Determines whether dynamic dates are shown as the date or as the expression that calculates the date.

For example, the filter details displayed below includes the date:

```
Day = 4/24/2008
```

The filter details shown below display the dynamic date expression:

```
Day = Today minus 12 Month(s)
```

For all these options, you can also choose to inherit the setting instead, as described in [Levels of report details configuration, page 511](#).

Configuring how logical operators are displayed

Filters can have multiple qualifications, which are the conditions (such as Revenue > 6,000,000) that the data must meet to be included in a report. A report can also contain multiple filters in its report filter. For example, a report filter contains two filters, East Coast Regions and Revenue > \$6M. The filters are joined by a logical operator, which is AND in this case.

Use the following settings to configure how to display the logical operators that join multiple conditions:

- **New line between conditions:** Determines whether or not each condition is placed on a separate line. You can also select **Automatic**, which inserts a line only when conditions are joined by different logical operators.

For example, a report is filtered for revenue greater than \$5 million and profit greater than \$800,000 and the Northeast region. The filter details use the **Automatic** option for **New line between conditions**. Because the conditions are joined by the same logical operator (and), the conditions are displayed on the same line.

Report details			
Report Filter: (Revenue > 5000000) And (Profit > 800000) And (Region = Northeast)			
	Metrics	Revenue	Profit
Region			
Northeast		\$8,554,415	\$1,300,732

A second report uses the same filter conditions, except the second logical operator is replaced by OR; the report is filtered for revenue greater than \$5 million and either profit greater than \$800,000 or the Northeast region. The **New line between conditions** option is still set to **Automatic**. A new line is started at the logical operator AND, and again

before the second condition. Using the new lines helps to distinguish the conditions.

Report details		
Report Filter: (Revenue > 5000000) And ((Profit > 800000) Or (Region = Northeast))		
	Metrics	
Region	Revenue	Profit
Northeast	\$8,554,415	\$1,300,732
South	\$5,389,280	\$806,956

- If you select **Yes** or **Automatic**, you can specify whether to **Single space** or **Double space** the conditions. Both examples above are single-spaced.
- Parentheses around conditions:** Determines whether or not parentheses are placed around each condition, such as (Region = Northeast). If a new line is inserted between conditions, you can often omit the parentheses since the different conditions are already differentiated from each other.

You can also select **Automatic**, which displays parentheses only when they resolve ambiguity in the expression. Parentheses are not included around conditions that are joined by the same logical operator. When conditions are joined by different operators, the parentheses are necessary to ensure that the conditions are grouped correctly.

For example, a report is filtered for revenue greater than \$5 million, and either profit greater than \$800,000 or the Northeast region. The filter details use the **Automatic** option for **Parentheses around conditions**. Parentheses are displayed because they ensure that the conditions are grouped and read correctly.

Report details		
Report Filter: Revenue > 5000000 And (Profit > 800000 Or Region = Northeast)		
	Metrics	
Region	Revenue	Profit
Northeast	\$8,554,415	\$1,300,732
South	\$5,389,280	\$806,956

A second report uses the same filter conditions, except the second logical operator is replaced by AND; the report is filtered for Revenue greater

than \$5 million and Profit greater than \$800,000 and the Northeast region. The **New line between conditions** option is still set to **Automatic**. The filter details use the **Automatic** option for **Parentheses around conditions**. Because the conditions are joined by the same logical operator (AND), and there is no ambiguity about the order, parentheses are not used.

Report details			
Report Filter: Revenue > 5000000 And Profit > 800000 And Region = Northeast			
	Metrics	Revenue	Profit
Region			
Northeast		\$8,554,415	\$1,300,732

- **Logical operator between conditions:** Specifies whether or not to display the logical operator between conditions. The options are:
 - **Yes** to display all operators
 - **No** to omit all operators
 - **AND only** to display only the AND operator
 - **OR only** to display only the OR operator

All the previous report samples showed all the operators, while the report sample below displays AND only. Notice that the OR before the Region condition is replaced by a comma.

Report details			
Report Filter: Revenue > 5000000 And (Profit > 800000,Region = Northeast)			
	Metrics	Revenue	Profit
Region			
Northeast		\$8,554,415	\$1,300,732
South		\$5,389,280	\$806,956

You can also choose to inherit these settings instead, as described in [Levels of report details configuration, page 511](#).

Configuring whether to use aliases in the filter details

You can rename an object on a report, to display a more meaningful description in the context of that particular report, for example. An alias does not change the name of the object, only the name displayed on the report. A filter still uses the name of the object, not the alias. You can determine whether aliases replace object names in the filter details.



For more information on creating aliases, see the *Basic Reporting Guide*.

For example, the Revenue metric has been aliased as Sales in a report. The filter details below display the metric name, not its alias.

Report details		
Report Filter: (Region = Northeast, Mid-Atlantic, Southeast) And (Revenue > 4000000)		
	Metrics	Sales
Region		
Mid-Atlantic		\$4,452,615
Northeast		\$8,554,415

The filter details shown below display the object's alias instead.

Report details		
Report Filter: (Region = Northeast, Mid-Atlantic, Southeast) And (Sales > 4000000)		
	Metrics	Sales
Region		
Mid-Atlantic		\$4,452,615
Northeast		\$8,554,415

You can also choose to inherit the setting instead, as described in [Levels of report details configuration, page 511](#).

Configuring the display of filter details: Procedure

Use the steps below to configure the display of filter details.

To configure the filter details for a project

- 1 Disable the report details preferences, if you have not already done so. For steps, see [To disable report details preferences, page 513](#).
- 2 In MicroStrategy Developer, right-click the project that you want to work with, and select **Project Configuration**. The Project Configuration Editor opens.
- 3 In the **Categories** list, expand **Project definition**, and then select **Document and Reports**.
- 4 Click **Report details and properties**. The Report Details Properties dialog box opens.
- 5 Select the **Filter Details** check box.
- 6 On the left, select **Filter Details - Contents**.
- 7 Select which filter types to display (report filter, view filter, report limits, and so on) and how the filters are displayed. For example, you can select whether to include the name of the filter type and whether to display the report limits before or after view filters. See [Configuring the content of the filter details, page 538](#) for details on all these options.
- 8 On the left, select **Filter Details - Other**.
- 9 For filters that contain attribute element list qualifications, configure how those lists are displayed. For example, you can specify whether to show the attribute name (such as Region or Category) and the separator between attribute names. See [Configuring how attribute element list qualifications are displayed, page 544](#) for details on all these options.
- 10 For filters that contain attribute form and set qualifications, configure how the qualifications are displayed. For example, you can select whether to use names or symbols for the operators. See [Configuring how attribute form and set qualifications are displayed, page 546](#) for details on all these options.
- 11 Configure how logical operators, which join multiple qualifications or filters, are displayed. For example, you can select whether to include parentheses around conditions. See [Configuring how logical operators are displayed, page 548](#) for details on all these options.

12 Click **OK** to save your changes.

To configure the filter details for a report

- 1** Disable the report details preferences, if you have not already done so. For steps, see [To disable report details preferences, page 513](#).
- 2** Open the report. You can either edit it or execute it.
- 3** From the **Data** menu, select **Report Details Properties**.



By default, all the filter details settings inherit the project setting, as described in [Levels of report details configuration, page 511](#). You can return to this default by selecting **<default>** from the drop-down list.
- 4** Click the **General** tab.
- 5** To ensure that the filter details are displayed in the report, select **Yes** from the **Filter Details** drop-down list.
- 6** To determine whether to display aliases or object names in the filter details, choose one of the following from the **Use aliases in Filter Details** drop-down list:
 - To use aliases, select **Yes**.
 - To use object names, select **No**.

For more information, including examples, see [Configuring whether to use aliases in the filter details, page 551](#).
- 7** Click the **Filter Details - Contents** tab to specify the content of the filter details, by selecting which filter types to include (report filter, view filter, report limits, and so on) and how the filters are displayed. For example, you can select whether to include the name of the filter type and whether to display the report limits before or after view filters. See [Configuring the content of the filter details, page 538](#) for descriptions of all these options.
- 8** Click the **Filter Details - Other** tab.
- 9** For filters that contain attribute element list qualifications, you can configure how those lists are displayed. For example, you can specify whether to show the attribute name (such as Region or Category) and the

separator between attribute names. See [Configuring how attribute element list qualifications are displayed, page 544](#) for descriptions of all these options.

- 10 For filters that contain attribute form and set qualifications, you can configure how the qualifications are displayed. For example, you can select whether to use names or symbols for the operators. See [Configuring how attribute form and set qualifications are displayed, page 546](#) for descriptions of all these options.
- 11 You can configure how logical operators, which join multiple qualifications or filters, are displayed. For example, you can select whether to include parentheses around conditions. See [Configuring how logical operators are displayed, page 548](#) for descriptions of all these options.
- 12 Click **OK** to save your changes.
- 13 If report details preferences are enabled, a message opens, indicating that some of your changes may not be applied to the report, as described in [Interaction of report details preferences and filter details for reports, page 538](#).
 - To disable report details preferences, click **Yes**. The changes that you made affect the display of filter details in this report, and report details preferences are disabled for all reports.
 - To keep report details preferences enabled, click **No**. The report details preferences override the changes that you made.

Enabling and customizing report details preferences

These report details preferences override some of the report details settings that are configured in the Report Details Properties dialog box in the Project Configuration Editor and the Report Editor. For example, you choose prompt details as a report details preference. When you open a prompted report, the name of each prompt and its prompt answers are displayed in the Report Details pane. You cannot remove the prompt details display from that report, unless you remove prompt details as a report details preference. In the report itself, however, you can configure how the prompt details are displayed, such as whether the title of the prompt is displayed.

You can also choose to use the report details settings instead of these report details preferences. These settings are available at the project and report level.

To use the report details settings, disable the report details preferences by clearing the **Apply these Developer preferences to the Report Details** check box in the Developer Preferences dialog box. For steps, see [To disable report details preferences, page 513](#).

If the report details preferences are used, you can specify the content of the report details, which can include any of the following:

- **Report description:** Displays the description of the report written by the creator or owner of the report.
- **Prompt details:** Display the name of each prompt on the report, as well as the prompt answer that the user chose before the report was executed.
- **Metric details:** Display the names of the objects on the report and the definitions of the metrics on the report.
- **Cache status:** Displays whether the report used a cached set of data for its results.
- **Filter details:** Display the name of any shortcut filter, the definition of each report filter, and the definition of each report limit used on the report.

A shortcut filter is a stand-alone filter that can be used on multiple reports. In contrast, a report filter is created within the report and cannot be used in another report.

If filter details are included, you can select either, both, or neither of the following:

- **Show definition of shortcut filters:** Displays the definition, instead of the filter name, of any stand-alone filters that were added to the report as a shortcut.
- **View filter details:** Displays the view filter, if one is applied to the report. View filters are not named, so the definition is displayed.

MicroStrategy OLAP Services allows view filters, which are a filter that can be applied by the user dynamically after the report has been executed and displayed.

The following report sample displays all of the options, including the view filter, in the report details. The definition of the shortcut filter (East Coast) is displayed.

Report details				
Prompt 1: Category Prompt Books, Movies				
Report Filter (East Coast): Region=Northeast, Southeast, Mid-Atlantic And Category=Books, Movies				
Report Limits: Empty Filter				
View Filter: Profit Between 35000 and 1000000				
Template: Metrics: Revenue Revenue {~+} Profit Sum(Profit) {~+}				
Report Cache Used: No				
Category	Region	Metrics	Revenue	Profit
Books	Mid-Atlantic		\$337,656	\$72,794
	Northeast		\$646,421	\$139,932
	Southeast		\$170,445	\$36,619
Movies	Northeast		\$1,001,561	\$63,071

To enable and customize report details preferences

- 1 From the **Tools** menu in MicroStrategy Developer, select **Preferences**. The Developer Preferences dialog box opens.
- 2 Expand the **Report** category, and then select **Report Details**.
- 3 By default, report details are displayed when a report is opened. To hide them, clear the **Show report details** check box.



If you clear the **Show report details** check box, the rest of the options on this interface become unavailable.

- 4 To define the size, color, or type of font for report detail display, click **Font Options**. The Font dialog box opens. Once you have formatted the font, click **OK** to return to the Developer Preferences dialog box.
- 5 By default, these preferences override some of the report details settings that are configured in the Report Details Properties dialog box. To use the report details settings instead, disable the report details preferences by clearing the **Apply these Developer preferences to the Report Details** check box.



If you clear the **Apply these Developer preferences to the Report Details** check box, the rest of the options on this interface become unavailable.

- 6 To display the short description of the report, select the **Include report description** check box.
- 7 To display prompt information for all prompts in the report, select the **Include prompt details** check box. This information includes the name of each prompt, as well as the prompt answer that the user chose before the report was executed.
- 8 To display filter information, select the **Include filter details** check box. This information includes the name of each shortcut filter, the definition of each report filter, and the definition of each report limit.
- 9 To display the definition, instead of the filter name, of any stand-alone filters that were added to the report as a shortcut, select the **Show definition of shortcut filters** check box.
- 10 To display view filter information, select the **Include view filter details** check box. View filters are not named, so the definition is displayed.
- 11 To display information about metrics, select the **Include metric details** check box. This information includes the names of the objects on the report and the definitions of the metrics on the report.
- 12 To display whether the report used a cached set of data for its results, select the **Include cache status** check box. The cache status is displayed as **Yes** when the report retrieved its results from a saved cache of data on the Intelligence Server. The cache status is displayed as **No** when the report retrieved its data from your data warehouse.
- 13 Click **OK** to save your changes.

LINKING REPORTS AND DOCUMENTS

Introduction

MicroStrategy offers many different ways for users to interact with their data, discover patterns, and draw conclusions. The most guided of these options are simple report manipulations such as pivoting or displaying and hiding subtotals. The most flexible option is report creation, which allows a user to access the entire schema and to use attributes and metrics from any part of the model. Drilling is a middle ground between these extremes, allowing users to navigate from the data that they are viewing to different levels of aggregation as defined by project hierarchies and drill maps. Drilling is sometimes described as enabling investigative workflows.

Linking reports and documents

Linking reports and documents is another tool to present investigative workflows, one that is more flexible than traditional drilling but more constrained than full report creation. The key feature that distinguishes links from drilling is the ability to use an object (such as an attribute element) on a report to trigger the execution of another report or a document that is substantially different from the original report. For example, a user viewing

an Employee detail report can click a link to execute a Regional Sales Breakdown report, or click a link on a Profit and Loss report to view a Departmental Summary document.

A link allows context to be passed to any prompted report or document, whether the destination report is related to the original. In contrast, drilling always implies that at least the filter of the drilled-to report is closely connected to the original report. Even the templates of the original and drilled-to reports are usually related, unless the drill path type is set to template. Template drill path types replace the template of the original report with a completely different destination template.

 For more information on using drilling in reports, see the *Answering Questions about Data* chapter in the *Basic Reporting Guide*. For information on creating drill maps for reports, including template drill path types, see [Chapter 5, Drill Maps](#).

This chapter will explore the many options exposed by links and how they can be used to create new opportunities for guided analysis.

 This chapter contains information about linking from reports, although linking from documents is similar. For specific information about linking from documents, as well as about documents in general, see the *Report Services Document Creation Guide*. You must have MicroStrategy Report Services to create documents. If you do not have it, contact your MicroStrategy sales representative for information on this add-on product.

Linking reports: Examples

 The examples in this chapter are based on data from the MicroStrategy Tutorial.

In short, links allow an analyst to explore additional, related data by executing another document or report from within a report, in either MicroStrategy Developer or MicroStrategy Web. For example, a **Regional Revenue** report contains links to another report called **Revenue by Category**. In MicroStrategy Developer, a user right-clicks a region (Central,

in the example shown below), points to **Link**, and then selects the link (Revenue by Category in this example) from the list.

Region	Metrics	Revenue
Central		\$5,000,000
Mid-Atlantic		
Northeast		
Northwest		
South		\$5,389,280
Southeast		\$2,239,951
Southwest		\$3,694,132
Web		\$3,902,762

Drill	▶
Link	▶
Attribute Forms	▶

Revenue by Category

The **Revenue by Category** report (the target of the link) is executed, as shown below.

Region	Category	Metrics	Revenue
Central	Books		\$376,836
	Electronics		\$3,506,062
	Movies		\$589,357
	Music		\$557,112

A link allows context to be passed to any prompted report or document, whether that destination is related to the original report. This is done by specifying how any prompts that are in the target are answered. For example, the region that a user clicks in the original report can be passed to the target report. If the user clicks Mid-Atlantic instead of Central in the **Regional Revenue** report, the same target report is executed, but with a different prompt answer, as shown below:

Region	Category	Metrics	Revenue
Mid-Atlantic	Books		\$337,656
	Electronics		\$3,106,940
	Movies		\$518,969
	Music		\$489,049

In MicroStrategy Web, the links are underlined. When you hover the mouse over the link, the link name is displayed, as shown below.

RegionMetrics Revenue	
<u>Central</u>	\$5,029,366
<u>Mid-Atlantic</u>	\$4,452,615
<u>North</u>	Revenue by Category
<u>Northwest</u>	\$1,761,187
<u>South</u>	\$5,389,280
<u>Southeast</u>	\$2,239,951
<u>Southwest</u>	\$3,694,132
<u>Web</u>	\$3,902,762

When you click Central, the **Revenue by Category** report is executed for the Central region, as shown below:

Region	Category	Metrics	Revenue
<u>Central</u>	<u>Books</u>		\$376,836
	<u>Electronics</u>		\$3,506,062
	<u>Movies</u>		\$589,357
	<u>Music</u>		\$557,112

Steps for creating this example are provided in [Creating linked reports that answer prompts with the selected objects, page 589](#).

A report can link to a document instead of another report, as shown in the example above. You can also create links in documents. In documents, links can be added to objects in a Grid/Graph, and they can link to another document or to a report. For documents, links are functional in Interactive Mode and Editable Mode in MicroStrategy Web. For examples, see the *Linking from Documents* chapter in the *Report Services Document Creation Guide*.

Prerequisites for working with links

This chapter assumes that you are familiar with creating reports, documents, and prompts. For resource material, see the following:

- For information on creating reports and prompts, including descriptions of the different prompt types, see the *Basic Reporting Guide*.
- For information on creating documents, see the *Report Services Document Creation Guide*.
- For information on hierarchies and how they are structured, see the *Answering Questions about Data* chapter of the *Project Design Guide*.

About links

A link is a connection in one report or document to another report or document. A link lets an analyst execute another document or report (the *target*) from a document or report (the *source*), and to pass parameters to answer any prompts that are in the target. For example, if a user is viewing a report containing regional sales, the user can click a particular region to execute another report that displays sales for the stores in that region. The source report could also link to a document that displays profit and cost values as well as the revenue in an interactive dashboard.

Links can:

- Be used from a report (the source)
- Be used from a Grid/Graph in a document (the source)
- Be created on attributes, metrics, hierarchies, or object prompts
- Execute a report or a document (the target)
- Open the target in either a new window or the same window (replacing the source) in MicroStrategy Web



In MicroStrategy Developer, all links open in a new window.

- Answer prompts in the target in a variety of ways, including using existing prompt answers from the source, using the objects selected in the source, running the prompts, sending a list of elements, and others

You can create multiple links for the same object. For example, the Region attribute in the **Regional Revenue** report can link to the **Revenue by Category** report, which displays data for the selected region only (as shown in the example in [Linking reports: Examples, page 560](#)). The same attribute can also link to a document that provides additional information about the region, such as revenue forecasts, inventory figures, store listings, employee schedules, and so on. For an example of multiple links, see [Specifying the default link when multiple links are available, page 565](#).

Different objects on a report can have different links. For example, the Region attribute in the **Regional Revenue** report can link to the **Revenue by Category** report, while the Revenue metric can link to a document with revenue forecasts and other key performance indicators.

Links in documents are similar to links in reports. The similarities and differences are described in the following table.

Links in Documents	Links in Reports
Can be attributes, metrics, hierarchies, and object prompts in Grid/Graphs displayed as grids	Can be attributes, metrics, hierarchies, and object prompts
Can link to documents and reports	Can link to documents and reports
Are not functional in MicroStrategy Developer	Are functional in executed reports in MicroStrategy Developer
Are functional in Interactive Mode and Editable Mode in MicroStrategy Web	Are functional in executed reports in MicroStrategy Web

Components of a link

When you create a link, you specify the following components:

- The **name** of the link. By default, the first link is named Link1, the second Link2, and so on, but you can change the names. Since the name of the link appears in both MicroStrategy Developer and MicroStrategy Web, it should be descriptive and informative to help users identify the target of the link.
- The **target** of the link. The target is the report or document that is executed when a user clicks the link.
- The **prompt to be answered and how to apply prompt answers to it**, if the target contains prompts. A list of the prompts in the target is provided. For each prompt, select a **prompt answer method**. For example, a prompt can use the existing prompt answers from the source, can be run so that the user must answer it, or can be answered with the object selected in the source. Other prompt answer methods exist; see [Specifying how prompts are answered in the target, page 567](#) for a full listing, with descriptions.

The list of prompts contains an option named **Any other prompt**, which controls any prompts that are not in the target when the link is created. These can be either:

- Prompts added to the target after the link is created
- Prompts that are created as the result of an answer to one of the original prompts in the target, such as a prompt-in-prompt answer

For more information about the **Any other prompt** option, see [Specifying prompt answers for any other prompts not listed, page 581](#).

If you create multiple links for the same object, you specify the *default link*, which is used when a user clicks the object. To choose any of the other links, the user must right-click the object. See [Specifying the default link when multiple links are available, page 565](#) for more details on how the other links are displayed. For an example, see [Specifying the default link when multiple links are available, page 565](#).

For all the links on an object, you can specify whether the target opens in the same window as the source (replacing the source) or in a new window. This applies to MicroStrategy Web only; links open in a new window in MicroStrategy Developer regardless of this setting. By default, the target opens in the same window as the source, effectively replacing it. You can instead choose to open the target in a new window, which allows both the source and the target to be visible simultaneously.

Specifying the default link when multiple links are available

You can create multiple links for a single object in the source, which provides additional report and document selections to analysts. For example, the report shown below contains two links:

Region	Metrics	Revenue
Central		\$5,029,366
Mid-	Drill	2,615
Nor	Link	Revenue Rank within Region
Nor	Attribute Forms	Regional Report by Category
South		\$5,389,280
Southeast		\$2,239,951
Southwest		\$3,694,132
Web		\$3,902,762

The top link, **Revenue Rank within Region**, is bold, indicating that it is the default link. When a user double-clicks a region, the default link is executed.

To access all other links, the user right-clicks the object and selects **Link**, as shown above. A list of all the links for that object is displayed. If the user selects the **Revenue Rank within Region** link, the **Revenue Rank**

Prompted for Region report executes, as shown below. Only the selected region is included.

Region	Employee	Metrics	Revenue	Revenue Rank
Central	Ellerkamp	Nancy	\$847,227	2
	Gale	Loren	\$1,669,290	3
	Torrison	Mary	\$1,690,350	4
	Zemlicka	George	\$822,500	1

If the user selects the **Regional Report by Category** link, the **Revenue Data by Category Prompted for Region** report is executed, as shown below.

Category	Metrics	Revenue	Profit	Cost
Books		\$376,836	\$81,331	\$295,505
Electronics		\$3,506,062	\$619,714	\$2,886,348
Movies		\$589,357	\$37,009	\$552,348
Music		\$557,112	\$26,269	\$530,843

In MicroStrategy Web, when a user hovers over the link in either a report or a document, only the default link is displayed in a pop-up. The same report used above is shown below in MicroStrategy Web.

Region	Metrics	Revenue
Central		\$5,029,366
Mid Atlantic		\$4,452,615
Northwest		\$1,761,187
South		\$5,389,280
Southeast		\$2,239,951
Southwest		\$3,694,132
Web		\$3,902,762

When a user clicks the object, the default link is executed. The user can right-click the object to display and choose from a list of all the links, as shown below:

Region	Metrics	Revenue
Central		\$5,029,366
Mid Atlantic		\$4,452,615
Northwest		\$1,761,187
South		\$5,389,280
Southeast		\$2,239,951
Southwest		\$3,694,132
Web		\$3,902,762

Specifying how prompts are answered in the target

For each prompt in the target (the report or document being executed from the link), you must select a **prompt answer method**, which is how to answer the prompt. You also specify how to answer any other prompts that are not listed. These are prompts that are created as the result of an answer to one of the original prompts in the target, such as a prompt-in-prompt answer. They can also be prompts added to the target later, after the link is created. These prompts are listed as the **Any other prompt** option in the list of prompts in the interface.

The prompt answer methods are briefly described in the table below. An expanded description of each, with an example, follows the table.

Prompt Answer Method	Requirements	Prompts in the Target Are Answered By...
Answer with the same prompt	Same prompt in the target and source	The prompt answer from the source
Prompt user	None	The user (prompts are displayed when the target is executed)
Default answer	None	The default prompt answer for the target prompt
Dynamically	- Attribute element prompt in target - Value prompt in target (available only in MicroStrategy Web)	The object selected in the source (for example, the attribute element or metric value that the user clicked on)
Empty answer	Target prompt must not be required	Nothing (no prompt answer is provided from the source and the user is not prompted)
Static element list	Attribute element prompt in target	List of elements created by the link designer Used to pass attribute elements from conditional metrics to the target
Current unit	Hierarchy prompt in target	The object selected in the source (for example, the attribute element that the user clicked on)
All valid units	Hierarchy prompt in target	Any object to the left of or above the user selection in the source



For a list of prompt types, see the *Building Query Objects and Queries* chapter in the *Basic Reporting Guide*. For a more extensive description of each prompt type from an analyst's perspective, with images of each prompt as it appears to users, see the *Answering Prompts and Refreshing Data* chapter of the *Basic Reporting Guide*.

- **Answer with the same prompt.** The same prompt answers that were used to execute the source are used in the target.

For example, the **Prompted Regional Revenue** report links regions to another report called **Revenue by Category**. Both reports contain the same prompt, which prompts the user to select the regions to display on the report. A user executes the **Prompted Regional Revenue** report and selects Mid-Atlantic, Northeast, and Southeast when prompted. He right-clicks Mid-Atlantic and selects the link to the **Revenue by Category** report. The **Revenue by Category** report displays data for all three regions, although only one region was selected in the link. For an expanded description of this example, with sample reports, see [Answering target prompts with the source prompt answers, page 570](#).

- **Prompt user.** When the target is executed, the user is prompted to provide answers manually.

For example, the **Regional Revenue** report links regions to another report called **Revenue by Category**, which is prompted for regions. A user right-clicks Mid-Atlantic on the **Regional Revenue** report, and selects the link to the **Revenue by Category** report. The regional prompt is displayed, although a region was selected on the source report. The **Prompt user** prompt method does not pass information to the target report, so the user must answer the prompts manually.

- **Default answer.** The prompt is answered by the default prompt answer for the prompt in the target. If the target prompt does not have a default answer, the **Empty answer** method is used. The prompt is not answered unless it is required, in which case the user is prompted.

For example, continue with the same **Regional Revenue** and **Revenue by Category** reports described above. This time, however, the **Default answer** prompt method is used in the link, and the regional prompt has a default answer of Central. When a user right-clicks Mid-Atlantic on the **Regional Revenue** report, the **Revenue by Category** report displays data for Central.

- **Dynamically.** The object selected in the source is passed to the prompt in the target. If this object does not answer the target prompt, the **Empty answer** method is used. The prompt is not answered unless it is required, in which case the user is prompted.

Available for attribute element prompts and value prompts only.



To select the dynamic prompt answer method for value prompts, you must use MicroStrategy Web. For steps, see the *MicroStrategy Web Help*.

For example, continue with the same **Regional Revenue** and **Revenue by Category** reports described above. This time, the link on Region in the source uses the **Dynamic** method. If a user clicks the Central region to run the link, Central is passed to the target to answer the prompt. The **Revenue by Category** report displays data for the Central region only. If the user clicks Mid-Atlantic instead, Mid-Atlantic is passed to the target and the **Revenue by Category** report displays data for the Mid-Atlantic region only. [Linking reports: Examples, page 560](#) shows samples for this example of the **Dynamic** prompt answer method.

- **Empty answer.** The prompt in the target is ignored, which means that the prompt is not answered. No prompt answer is provided from the source and the user is not prompted to provide answers.



The prompt must not be required, otherwise, after the user clicks the link, the prompt is executed and must be answered before the target will execute.

The Empty prompt answer method, when used in conjunction with the dynamic prompt answer method (described above), allows a source report to answer one prompt in a target report with the user selection, while ignoring any other prompts.

For example, a source report contains Region in the rows and Category in the columns. Both attributes link to the same target report, which prompts for Region and Category. If a user clicks Central in the source, the target report is executed, with data for all categories for the Central region. If a user clicks Books in the source, the target is executed, with data for all regions for the Books category. The prompt for the selected attribute is answered using the dynamic prompt answer method, while the prompt for the other attribute is ignored, using the Empty answer prompt method. No prompt answer is provided to the target, but the user is not prompted, either. For a more detailed explanation of the Empty answer prompt answer method, including samples, see [Ignoring target prompts: Using the Empty answer method, page 571](#).

- **Static element list.** This method answers the target prompt with a list of elements. This allows you to pass fixed conditions to the target.

Once you select this method when you are creating the link, the Element List area becomes available, allowing you to select the elements that will be passed as answers to the element list prompt in the target.

Available for attribute element prompts only.

Use Static element list when you link from a conditional metric. Conditional metrics are filtered by attribute elements, and this prompt method allows you to pass those elements to the target. For an example,

and more details on this method, see [Passing metric conditions: Using the Static element list answer prompt method, page 574](#).

This method also allows you to pass characteristics and key figures from MDX cubes to the target; for more information on using links with MDX cubes, see [Passing MDX cube data: Using the Static element list answer prompt method, page 576](#).

The prompt answer methods listed below are available for hierarchy prompts only. The source report used in these examples contains Category, Region, and Revenue. The source report links Region to the target report, which contains the Geography hierarchy and Revenue, with a hierarchy prompt in the report filter.

- **Current unit.** Similar to the **Dynamic** prompt answer method, where the prompt is answered using the object selected in the source. If the user selects an attribute header rather than a specific attribute element, the **Empty answer** method is used. The prompt is not answered unless it is required, in which case the user is prompted.

For example, if the link uses this prompt answer method, when the user clicks Central in the target report, the target displays all the revenue for the Central region. The report is filtered on Region only.

- **All valid units.** Any object to the left of or above the user selection in the source is used as the prompt answer for the target. In other words, this method passes all the selections made on the source, rather than just the selection made for the link. If the user does not select any valid objects (for example, the user selects an attribute header rather than a specific attribute element), the **Empty answer** method is used. The prompt is not answered unless it is required, in which case the user is prompted.

For example, if a user selects Central in the target report, a category is also selected. When the user clicks Central in the source, if the link uses this prompt answer method, the target report displays the revenue from the Central region, but only for the category that was selected in the target. The report is filtered on Region and Category.

For a more detailed comparison of these two methods, and report samples, see [Prompt answer methods for hierarchy prompts, page 577](#).

Answering target prompts with the source prompt answers

When you select the **Answer with the same prompt** method, the existing prompt answers from the source are used in the target. To use this method, the target and the source must use the same prompt. If the same prompt

does not exist in the source and in the target, the user is prompted when the target is executed.

For example, the regions in the **Prompted Regional Revenue** report have links to another report called **Revenue by Category**. Both reports contain the same prompt, which prompts the user to select the regions to display on the report. Execute the **Prompted Regional Revenue** report and select Mid-Atlantic, Northeast, and Southeast when prompted. The results are shown below, in MicroStrategy Developer.

Region	Metrics	Revenue
Mid-Atlantic		\$4,459,815
North		415
South		

Right-click Mid-Atlantic and select the link to the **Revenue by Category** report. The **Revenue by Category** report displays data for all three regions, as shown below, although only one region was selected in the link.

Region	Category	Metrics	Revenue
Mid-Atlantic	Books		\$337,656
	Electronics		\$3,106,940
	Movies		\$518,969
	Music		\$489,049
Northeast	Books		\$646,421
	Electronics		\$5,962,709
	Movies		\$1,001,561
	Music		\$943,724
Southeast	Books		\$170,445
	Electronics		\$1,552,007
	Movies		\$264,286
	Music		\$253,213

Ignoring target prompts: Using the Empty answer method

When you select the **Empty answer** method, the prompt in the target report or document is ignored. This means that the prompt is not answered—not by the selections made on the source, by the default prompt answer, or by prompting the user. You must ensure that the prompt is not required, however, because in that case the user is prompted when the target is executed from the link in the source. The Empty prompt answer method, when used in conjunction with the dynamic prompt answer method, allows a

source report to answer one prompt in a target report with the user selection, while ignoring any other prompts.

For example, the following report contains Region on the rows, and Category and Revenue in the columns. Region contains a link, as shown in the following image:

Region	Metrics Category	Revenue			
		Books	Electronics	Movies	Music
Central		\$251,279	\$5,307,467	\$241,405	\$1,112,782
Mid-Atlantic	Drill	\$210,320	\$22,565,830	\$208,237	\$194,436
North	Link				\$368,269
North	Attribute Forms	\$88,988	\$9,165,295	\$82,912	\$78,666
South		\$989,021	\$4,224,822	\$190,283	\$178,414
Southeast		\$115,667	\$2,380,031	\$544,283	\$100,336
Southwest		\$187,364	\$3,926,387	\$898,909	\$166,645
Web		\$132,770	\$2,704,890	\$627,471	\$116,147

When the selected link is run, the following report is displayed:

Region	Category	Metrics	Revenue	Revenue per Employee
Central	Books		\$251,279	\$62,820
	Electronics		\$5,307,467	\$1,326,867
	Movies		\$241,405	\$60,351
	Music		\$1,112,782	\$278,196

Note that only the selected region (Central) is displayed in the target. All categories are also displayed.

Return to the source report, which contains a link on Category as well, as shown in the following image:

Region	Metrics Category	Revenue			
		Books	Electronics	Movies	Music
Central		\$251,279	\$5,307,467	\$241,405	\$1,112,782
Mid-Atlantic	Drill	\$210,320	\$22,565,830	\$208,237	\$194,436
Northeast	Link				\$368,269
Northwest	Attribute Forms	\$88,988	\$9,165,295	\$82,912	\$78,666
South		\$989,021	\$4,224,822	\$190,283	\$178,414
Southeast		\$115,667	\$2,380,031	\$544,283	\$100,336
Southwest		\$187,364	\$3,926,387	\$898,909	\$166,645
Web		\$132,770	\$2,704,890	\$627,471	\$116,147

When the selected link is run, the following report is displayed:

Region	Category	Metrics	Revenue	Revenue per Employee
Central	Books		\$251,279	\$62,820
Mid-Atlantic	Books		\$210,320	\$26,290
Northeast	Books		\$2,048,826	\$341,471
Northwest	Books		\$88,988	\$29,663
South	Books		\$989,021	\$329,674
Southeast	Books		\$115,667	\$28,917
Southwest	Books		\$187,364	\$37,473
Web	Books		\$132,770	\$132,770

This is the same target report as the other link, however, only the selected category (Books) is displayed. All the regions are displayed. How does this work?

The target report contains two prompts, one for Region and one for Category, but the user makes only one selection (a region or a category), yet is not prompted for the other. This is because the prompts in the link definition use different prompt answer methods. The link on region uses the following prompt answer methods:

- Region prompt: Dynamic
- Category prompt: Empty answer

This passes the selected region to the target, to answer the region prompt, and ignores the category prompt. Likewise, the link on category uses the following prompt answer methods:

- Region prompt: Empty answer
- Category prompt: Dynamic method

This passes the selected category to the target, to answer the category prompt, and ignores the region prompt.



Prompt answers cannot be required for either of these prompts when the Empty answer prompt answer method is used. If they are, the prompts must be answered by the user when the target report is executed, even if the prompt answer method is Empty Answer.

Passing metric conditions: Using the Static element list answer prompt method

When you select the **Static element list** method, you select the elements that are passed as answers to an element list prompt in the target. This allows you to pass the elements from a conditional metric in the source to the target.

For example, a source report contains Region, Customer, and metrics for Books Revenue, Movies Revenue, and Music Revenue. The report links from the conditional metrics to a target report, as shown below. Note that only a section of the report is shown; the report contains data for all regions and all customers.

Region	Customer		Metrics	Books Revenue	Movies Revenue	Music Revenue
Central	Aadland	Warner		\$282	\$339	\$469
	Aadland	Constant				\$240
	Aagesen	Bink				
	Aamodt	Stacy				
	Aaron	Ferrell				\$453
	Abadilla	Lennie		\$200	\$185	\$170
	Abajian	Lorin		\$104	\$188	\$188
	Aharra	Hugh		\$404	\$397	\$432

The name of the link implies that the condition from the selected metric (in this case, Category = Books) is passed to the target. The target report is executed and displays as shown below. Notice that the report is filtered for the Books category. You can confirm that by comparing the revenue amounts. Revenue for Warner Aadland in the target is \$282, which is the same as the Revenue amount in the source report for the same customer.

Report Filter: Category = Books								
Region	Customer		Metrics	Revenue	Cost	Profit	Units Sold	Revenue per Employee
Central	Aadland	Warner		\$282	\$218	\$64	17	\$71
	Aadland	Constant		\$69	\$55	\$14	8	\$17
	Aagesen	Bink		\$445	\$368	\$78	30	\$111
	Aamodt	Stacy		\$90	\$74	\$16	10	\$22
	Aaron	Ferrell		\$201	\$155	\$46	16	\$50
	Abadilla	Lennie		\$200	\$153	\$47	14	\$50
	Abajian	Lorin		\$104	\$80	\$24	6	\$26
	Aharra	Hugh		\$404	\$321	\$83	28	\$101

Note that once again the complete report is not shown; it contains data for all regions and all customers.

If the link from the Movies Revenue metric on the source report is chosen instead, the same target report is executed, but with different results, as shown below:

Report Filter: Category = Movies								
Region	Customer		Metrics	Revenue	Cost	Profit	Units Sold	Revenue per Employee
Central	Aadland	Warner		\$339	\$309	\$29	24	\$85
	Aadland	Constant		\$330	\$312	\$18	25	\$82
	Aagesen	Bink		\$400	\$379	\$21	30	\$100
	Aamodt	Stacy		\$148	\$147	\$1	9	\$37
	Aaron	Ferrell		\$311	\$284	\$27	22	\$78
	Abadilla	Lennie		\$185	\$168	\$17	10	\$46
	Abajian	Lorin		\$188	\$175	\$14	12	\$47
	Abarna	Hugh		\$397	\$390	\$7	29	\$99

The target report is filtered by the condition on the conditional metric (Category = Movies), and the data has changed accordingly. Again, only a portion of the report is shown in the sample.

How does this work? You cannot pass the elements in the condition of a conditional metric directly to the target; you instead add those elements to a static element list in the link.



If you do not add any elements to the element list, the **Empty answer** method is used. The prompt is not answered unless it is required, in which case the user is prompted.

In this example, the target report is prompted for Category. Each conditional metric in the source report contains a link to the target. In the link, the Category prompt uses the Static element list prompt answer method. For the link from the Books Revenue metric, the list contains Books. For the link from the Movies Revenue metric, the list contains Movies, and so on.

The target report also contains prompts for Region and Customer. In the links in the example above, those prompts use the **Empty answer** prompt method. The prompts are not answered; no data about Region or Customer is passed from the source to the target. The region and customer selections can be passed to the target if the **Dynamic** prompt method is used instead. The following source report is based on the previous example, but the Region and Customer prompts now use the **Dynamic** prompt method instead of **Empty answer**. Notice that the link name has changed as well to reflect this

change; the link name is very important to help a user understand what the target report will contain.

Region	Customer		Metrics	Books Revenue	Movies Revenue	Music Revenue
Central	Aadland	Warner		\$282	\$339	\$469
	Aadland	Constant	Drill		30	\$240
	Aagesen	Bink	Link	Books Region and Customer Revenue Data		
	Aamodt	Stacy	Sort rows by this column			
	Aaron	Ferrell			11	\$453
	Abadilla	Lennie		\$200	\$185	\$170
	Abajian	Lorin		\$104	\$188	\$188
	Abercrombie	Mark		\$104	\$307	\$137

Note that only a portion of the source report is shown; all regions and customers are included on the report.

When the link is run, the following target report is displayed. This report is shown in its entirety.

Report Filter: (Category = Books) And (Region = Central) And (Customer = Aadland:Warner)							
Region	Customer	Metrics	Revenue	Cost	Profit	Units Sold	Revenue per Employee
Central	Aadland	Warner	\$282	\$218	\$64	17	\$71

Only the Region (Central) and Customer (Warner Aadland) selected in the source report are included on this target report. The report is also filtered by Category, as before.

Steps for creating this example are provided in [Creating linked reports that answer prompts using metric conditions, page 592](#).

The **Static element list** method also allows you to pass characteristics and key figures from MDX cubes to the target; for more information on using links with MDX cubes, see [Passing MDX cube data: Using the Static element list answer prompt method](#) below.

Passing MDX cube data: Using the Static element list answer prompt method

You can use an MDX cube report as a source report, and pass data from the MDX cube to the target, just as you can with any other report. The difference

is that an MDX cube report uses data from an MDX cube source, and this data must be passed to the target using static element lists.

The data from the MDX cube can include characteristics and key figures.

- **Characteristics** are “translated” into dimensions in MicroStrategy, and characteristic values into attribute elements. An example of a characteristic is Sales Region, with characteristic values of North, Central, and South.
- **Key figures** are numeric and time-related data, such as revenue figures, months, and years.



For a more detailed discussion of MDX cube terminology, including characteristics and key figures, and information on creating MDX cube reports, see the *MDX Cube Reporting Guide*.

Before you create a link from an MDX cube report, add the necessary prompts to the target, just as you would with any other report. When you create the link, select the target prompts, specify the Static element list prompt answer method, and choose the elements.

Prompt answer methods for hierarchy prompts

Links can pass only a single element of any attribute, but they can pass elements from different attributes at the same time. For example, with the links between the **Regional Revenue** and **Revenue by Category** reports described throughout this chapter, the selected region is passed to the target, when the dynamic prompt answer method is used. (See [Linking reports: Examples, page 560](#) for the report samples.) Only the one region is passed, regardless of any other selections on the source report, because that answers the attribute element prompt in the target report.

In contrast, a hierarchy prompt allows users to select prompt answers from one or more attribute elements from one or more attributes. This prompt gives users the largest number of attribute elements to choose from when they answer the prompt to define their filtering criteria. The **All valid units** prompt answer method passes selections made on the source report, rather than just the selection made for the link, to the target report. To restrict the prompt answer to just the selected attribute element, use the **Current unit** prompt answer method. Like the Dynamic method, only the attribute element selected is passed to the target.

For example, the following report is used as the source. It contains Year, Region, and Revenue. As shown below, Region contains two links, one that

passes the current region only (using the **Current unit** prompt answer method) and another that passes the selected year and region (using the **All valid units** prompt answer method).

Year	Region	Metrics	Revenue
2004	Central		\$1,823,715
	Mid-Atlantic		\$1,026,100
	Northeast		\$1,026,100
	Northwest		\$1,026,100
	South		\$2,011,972
	Southeast		\$836,399
	Southwest		\$1,418,189
	Web		\$720,560
2005	Central		\$2,182,632
	Mid-Atlantic		\$7,662,786
	Northeast		\$3,668,765
	Northwest		\$3,189,483
	South		\$1,901,549
	Southeast		\$1,036,970
	Southwest		\$1,625,727
	Web		\$1,133,901
2006	Central		\$2,906,587
	Mid-Atlantic		\$9,384,211
	Northeast		\$4,748,571
	Northwest		\$3,653,040
	South		\$1,669,019
	Southeast		\$1,266,947
	Southwest		\$2,135,390
	Web		\$1,726,816

The target report contains the Geography hierarchy and the Revenue metric. It is filtered by a hierarchy prompt on Geography.

Notice that 2004 and Central are selected in the image above. If you click the Current Year and Region link, 2004 and Central are passed to the target as the prompt answers. This is confirmed in the report filter details, as shown in the report sample below. Only Central is returned, with a revenue amount (\$1,823,715) that matches the revenue amount in the source report.

Report Filter (Local Filter): (Year = 2004) And (Region = Central)		
Region	Metrics	Revenue
Central		\$1,823,715

If you click the Current Region Only link instead, although 2004 and Central are still selected, only Central is passed as a prompt answer to the target report. The report is filtered by Central only; all categories are included. The revenue amount is therefore much higher (\$6,912,934 instead of \$1,823,715), as shown below.

Report Filter (Local Filter): Region = Central		
Region	Metrics	Revenue
Central		\$6,912,934

The valid units for the **All valid units** prompt are any elements that are to the left of or above the user selection in the source report. For example, add Call Center and Employee to the right of Region in the source report, as shown below. (The links have been renamed to the prompt answer methods.)

Year	Region	Call Center	Employee	Metrics	Revenue
	Central	Milwaukee	Gale Loren		\$591,808
			Torrison Mary		\$608,257
			Zemlicka George		\$298,169
		Fargo	Ellerkamp Nancy		\$325,481
	Mid-Atlantic	Charleston	Bernstein Lawrence		\$1,433,104
			Adrienne		\$1,504,537
			Robert		\$1,398,420
			Brown Vernon		\$473,569
			Corcoran Peter		\$461,761
			Ingles Walter		\$254,414
			Smith Thomas		\$290,049
			Young Sarah		\$315,972

Notice that 2004, Central, the Milwaukee and Fargo Call Centers, and a number of employees are selected. When the Current Unit link is clicked, only Central is displayed on the target report, with the same revenue amount (\$6,912,934) as in the target report from the previous example.

Report Filter (Local Filter): Region = Central		
Region	Metrics	Revenue
Central		\$6,912,934

If the All Valid Units link is clicked instead, Central is returned, with 2004 revenue only, as shown below.

Report Filter (Local Filter): (Year = 2004) And (Region = Central)		
Region	Metrics	Revenue
Central		\$1,823,715

The link could be on Call Center instead, as shown in the source report below.

Year	Region	Call Center	Employee	Metrics	Revenue
	Central	Milwaukee	Gale Loren		\$591,808
			Torrison Mary		\$608,257
			Zandbergen George		\$298,169
	Mid-Atlantic	Fargo	Maney		\$325,481
		Washin	All valid units		\$1,433,104
			Adrienne		\$1,504,537
		Charleston	Hollywood Robert		\$1,398,420
			Brown Vernon		\$473,569
			Corcoran Peter		\$461,761
			Ingles Walter		\$254,414
			Smith Thomas		\$290,049
			Young Sarah		\$315,972

If the All Valid Units link is clicked, the target report is filtered by Year, Region, and Call Center (the selection and the attributes to the left of it), as shown below. Note that the revenue amount (\$1,498,233) is less than the amount (\$1,823,715) displayed in the previous report that filtered by 2004 and Central. The difference of \$325,481 is Fargo's 2004 revenue contribution, as shown in the report above.

Report Filter (Local Filter): (Year = 2004) And (Region = Central) And (Call Center = Milwaukee)		
Call Center	Metrics	Revenue
Milwaukee		\$1,498,233

Note that throughout this section, the target report has not changed, although the last report shown above displays Call Center, while all the other target report samples displayed Region. This is because the target report contains the Geography hierarchy; the lowest attribute from that hierarchy

that is passed in the prompt is displayed on the report. In this case it is Call Center; in the others it was Region.



For more information on hierarchy prompts in general, see the *Basic Reporting Guide*.

Specifying prompt answers for any other prompts not listed

Any other prompts are those prompts that are not in the target when you are creating the link. These can be either:

- Prompts added to the target after the link is created
- Prompts that are created as the result of an answer to one of the original prompts in the target, such as a prompt-in-prompt answer

These prompts are listed as the **Any other prompt** option in the list of prompts in the interface. You can choose any of the following prompt answer methods for the **Any other prompt** option:

- Answer with the same prompt from the source
- Prompt user
- Default answer
- Empty answer

By default, the **Any other prompt** option uses the **Prompt user** method. For more information on each method, see [Specifying how prompts are answered in the target](#).

For example, the **Regional Revenue** report contains a link to another report called **Revenue by Category** (the target). When the link was created, **Revenue by Category** contained only a prompt for Region. The link uses the **Dynamic prompt answer** mode to answer that prompt, so the region that the user selects to access the link answers the prompt. The user is not prompted when the target is executed. This scenario is shown in [Linking reports: Examples, page 560](#). However, after the link was created, a second prompt, for Customer Region, is added to the **Revenue by Category** report.

Now a user selects Central in the **Regional Revenue** report, and clicks the link to **Revenue by Category**. The report does not execute immediately, but instead the Customer Region prompt appears. The user must select a customer region to continue or, since the prompt is not required, can include

all categories by clicking **Finish**. In the sample shown below, the user answered the prompt with the Central and Mid-Atlantic customer regions.

Report Filter: Region=Central And {Customer Region}=Central, Mid-Atlantic			
Region	Category	Metrics	Revenue
Central	Books		\$363,125
	Electronics		\$3,371,784
	Movies		\$568,170
	Music		\$536,446

Because the Customer Region prompt was added after the link was created, the prompt uses the prompt answer method assigned to **Any other prompt**. Since the creator of the link did not change that method, it is still defined as the default of **Prompt user**.

If you edit the Revenue by Category link in the **Regional Revenue** report now, the Customer Region prompt is displayed in the list of prompts. Its prompt answer mode is defined as **Prompt user**, although you can change it. You can also select a different prompt answer mode for **Any other prompt**.

Change the Customer Region prompt to **Empty answer**. Change the **Any other prompt** option to **Default answer**. Create a prompt on Customer State, and add the following as default answers:

- Illinois
- Indiana
- Iowa
- Kansas
- Kentucky

Add the new Customer State prompt to the **Revenue by Category** report.

Re-execute the **Regional Revenue** report. Right-click **Central**, point to **Link**, and then select **Revenue by Category**. You are not prompted at all.

The target is filtered by Region and Customer State, but not Customer Region, as shown below.

Report Filter: Region=Central And {Customer State}=Illinois, Indiana, Iowa, Kansas, Kentucky			
Region	Category	Metrics	Revenue
Central	Books		\$189,354
	Electronics		\$1,778,060
	Movies		\$296,820
	Music		\$280,143

- The Region prompt is answered dynamically (by your selection of Central).
- The Customer State prompt uses the default answers defined in the prompt (Illinois, Indiana, Iowa, Kansas, and Kentucky).
- The Category prompt is ignored and therefore does not show up in the report filter.

Creating links

Prerequisites

Before creating links, you should:

- Create the source and target documents and reports.
- Know what types of prompts the targets require and how they will be answered. The following table describes prerequisites for different prompt answer methods.

Prompt Answer Method	Prerequisite
Answer with the same prompt	The same prompt is used in the target and the source.
Default answer	The target prompt contains a default answer.

Prompt Answer Method	Prerequisite
Dynamically	The target prompt is an attribute element prompt or value prompt. The object that the user selects to execute the target answers the target prompt. Note: To select Dynamically for value prompts, you must use MicroStrategy Web. For steps, see the <i>MicroStrategy Web Help</i> .
Static element list	The target prompt is an attribute element prompt.
Empty answer	The prompt should not be required.
Current unit	The target prompt is a hierarchy prompt. The object that the user selects to execute the target answers the target prompt.
All valid units	The target prompt is a hierarchy prompt. The object(s) that the user selects to execute the target answers the target prompt.

To create a link

- 1 In MicroStrategy Web, open the report. You can edit the report or execute it.
- 2 Right-click the object to create a link from, and select **Edit Links**. The Links dialog box opens.

You can create links from the following objects:

 - Attributes
 - Hierarchies
 - Metrics
 - Object prompts
- 3 If this is not the first link created for the object, click **New**. If it is, continue with the next step.

Define the link

- 4 Click ... (**Browse**) next to **Run this report or document** to find and select the target report or document.

Apply prompt answers

- 5 Select a target prompt from the box.
- 6 Select a prompt answer method from the drop-down list (unless otherwise indicated, each method is available for all prompt types):
 - **Answer with the same prompt from the source.** The same prompt answers that were used to execute the source are used in the target. To use this method, the target and the source must use the same prompt.
 - **Prompt user.** The user is prompted to provide answers manually when the target is executed.
 - **Use default answer.** The prompt is answered by the default prompt answer for the prompt in the target report.
 - **Answer with an empty answer.** The prompt in the target is ignored, which means that the prompt is not answered. No prompt answer is provided from the source and the user is not prompted to provide answers.

 If the prompt is required, the prompt is still executed when a user clicks this link, since a prompt answer must be provided.
 - **Answer Dynamically.** The prompt is answered using the object selected in the source.

Available for attribute element prompts and value prompts only.
 - **Answer using static element list.** This method overrides the prompt answers from the source with a list of elements. Once you select this method when you are creating the link, the Element List area becomes available, allowing you to select the elements that will be passed as answers to the element list prompt in the target.

Available for attribute element prompts only.
 - **Answer using current unit.** Similar to the dynamic prompt answer method, where the prompt is answered using the object selected in the source.

Available for hierarchy prompts only.
 - **Answer using all valid units.** Any object to the left of or above the user selection in the source is used as the prompt answer for the target. In other words, this method passes all the selections made on the source, rather than just the selection made for the link.

Available for hierarchy prompts only.



For examples of each of the prompt answer methods, see [Specifying how prompts are answered in the target, page 567](#). For a list of prompt types, see the *Building Query Objects and Queries* chapter in the *Basic Reporting Guide*.

- 7 Repeat the above steps, starting with [Apply prompt answers](#), for each prompt in the target.

Specify the prompt answer method for any other prompts

Any other prompts are those prompts that are not in the target when you are creating the link, such as prompts added to the target later. By default, the Prompt user method is selected for these prompts, but you can change the method. For more information on how these prompts occur, including examples, see [Specifying prompt answers for any other prompts not listed, page 581](#).

- 8 Select **Any other prompt** in the **Target prompts** list.
- 9 Select a **Prompt Answer Method** from the following list; only these methods are available for the Any other prompt option:
 - **Answer with the same prompt from the source**
 - **Prompt user**
 - **Use default answer**
 - **Answer with an empty answer**



See the previous list of prompt answer methods for brief descriptions of each.

Rename the link

- 10 By default, links are named Link1, Link2, and so on. To rename it, click **Rename**, and replace the name with new text.

You can help identify the link for analysts by using the name of the target and/or describe how its prompts are answered.

Create another link on the same object

To create additional links from the same object, continue with the steps below. Otherwise, continue with [Specify whether a new window is used in MicroStrategy Web, page 587](#). For an example of multiple links and

how they work from a source report, see [Specifying the default link when multiple links are available, page 565](#).

- 11 Click **New**, and define the link as described above, starting with [Define the link, page 584](#).
- 12 By default, the first link that is created is set as the default link, which is used when a user clicks the object in MicroStrategy Web or double-clicks it in MicroStrategy Developer. To set a different link as the default, select it in the list at the top of the dialog box, and click **Set as default**. For details on default links, see [Specifying the default link when multiple links are available, page 565](#).
- 13 You can set the order that the links are displayed in the right-click menu. Select a link in the list at the top of the dialog box, and click **Move Up** or **Move Down** to re-order the links.

Specify whether a new window is used in MicroStrategy Web

For all the links on an object, you can specify whether the target opens in the same window as the source (replacing the source) or in a new window when a user clicks the link in MicroStrategy Web.

- 14 By default, in MicroStrategy Web, the target report or document opens in the same window as the source. To have the target open in a new window, which allows both the source and the target to be visible simultaneously, select the **Open in a new window** check box.



This setting applies to MicroStrategy Web only; links open in a new window in MicroStrategy Developer regardless of this setting.

- 15 Click **OK** to return to the source report.

Copying links to other objects in the report

To use the same link (or a similar one) on a different report object, copy the link. This saves time, and ensures that the links are exactly the same. You can copy only one link or multiple links.

For example, you want to create three links on Region and the same three links on Call Center. Create the links on Region, and then copy them to Call Center. If one of the Call Center links needs to be slightly different, you can edit it after copying it.

Links may already have been defined for the object that you are copying the links to. In this case, the following rules apply:

- Any existing links are not removed or overwritten, even if they point to the same target.
- If a copied link has the same name as an existing link, the copied link is renamed as “Copy of *Name*”.
- If a copied link is a default link, it is copied to the object as the default link, unless that object already has a default link defined. If it does, the link is copied, but it is not set as the default.

To copy a link



This procedure assumes you have already created the link that you want to copy. For steps, see [To create a link, page 584](#).

- 1 In MicroStrategy Developer, open the report. You can edit the report or execute it.
- 2 Right-click the object that contains the link to be copied, and select **Edit Links**. The Link Editor opens.
- 3 Click **Copy Links**. The Copy Links dialog box opens.
- 4 In the list under **Select the links to be copied**, select the link or links to be copied.
- 5 In the list under **Select the units to which the selected links will be copied**, select the object or objects to copy the links to.
- 6 You can choose whether or not to copy the **Open in a new window** setting to the new link. This setting determines whether the target opens in the same window as the source (replacing the source) or in a new window (simultaneously displaying the target and the source) in MicroStrategy Web.

By default, the **Open in a new window** setting is not copied. To change this, select the **Copy ‘Open in a new window’ setting to the target units** check box.
- 7 Click **Copy**. You are returned to the Link Editor.
- 8 Click **OK** to return to the source report.

- 9 You can modify copied links, as necessary.
 - Right-click the object that contains the new link, and select **Edit Links**. The Link Editor opens.
 - Make any necessary changes.
 - Click **OK** to save the changes and return to the source.

Creating linked reports

These examples include steps to create the links. They assume you are familiar with creating prompts and reports. For details on creating reports and prompts, see the *Basic Reporting Guide*.

These examples use data from the MicroStrategy Tutorial. The reports created in this example should be saved in the `Public Objects\Reports` folder or a subdirectory so that they are available when you log into MicroStrategy Web.

Creating linked reports that answer prompts with the selected objects

The following steps re-create the example shown in [Linking reports: Examples, page 560](#). The **Regional Revenue** report contains a link to the **Revenue by Category** report. When a user clicks a region in MicroStrategy Web or double-clicks a region in MicroStrategy Developer, the **Revenue by Category** report is executed, and data for only the selected region is shown.

The high-level steps to create this example are listed below:

- 1 [Create the prompt for region, page 590](#)
- 2 [Create the prompted target report \(Revenue by Category\), page 590](#)
- 3 [Create the source report \(Regional Revenue\), page 591](#)

Create the prompt for region

This prompt will be used on the target report to determine the region to display. When the link from the source report to the target is used, this prompt is answered by the region selected in the source.

This is an attribute element list prompt, which allows the link in the source report to use the dynamic prompt answer method. This method passes the selected region to the target report.

To create an attribute element list prompt

- 1 In MicroStrategy Web, log in to a project.
- 2 Open any folder page (for example, click the Shared Folders icon on the home page).
- 3 Click the **Create Prompt** icon . The Create Prompt page opens.
- 4 Click **Attribute Element List**. The New Prompt page opens.
- 5 Click **Select Attribute**. The Select an Object dialog box opens.
- 6 Select **List all elements**.
- 7 Click the **General** tab.
- 8 In the **Title** field, type **Region**.
- 9 Clear the **Prompt answer is required** check box.
- 10 Save the prompt.

Create the prompted target report (Revenue by Category)

The target report is the report that is executed from the original report (the source). The target report is filtered by the prompt that you created in the previous procedure.

To create a prompted report

- 1 In MicroStrategy Web, log in to a project.
- 2 Click the MicroStrategy icon  at the upper left of any page and select **Create Report**.
- 3 Add **Region**, **Category**, and the **Revenue** metric to the report.
- 4 Add the **Region** prompt that you created in the previous procedure to the report filter.
- 5 Save and close the report, naming it **Revenue by Category**.

Create the source report (Regional Revenue)

The source report is the report that contains the link to the target. The link in this source report passes the selected region to answer the prompt in the target.

To create a report that links to another report

- 1 In MicroStrategy Web, log in to a project.
 - 2 Click the MicroStrategy icon  at the upper left of any page and select **Create Report**.
 - 3 Add **Region** and the **Revenue** metric to the report.
- Create the link**
- 4 Right-click **Region** and select **Edit Links**. The Links dialog box opens.
 - 5 Click ... (**Browse**) next to **Run this report or document**. The Select Target dialog box opens.
 - 6 Navigate to the **Revenue by Category** report, select it, and click **Open**. This report is the target of the link.
 - 7 Select **Region** in the **Target prompts** list. This prompt on the Revenue by Category report filters the target report by region.

- 8 Choose **Answer Dynamically** from the drop-down list under **Select Prompt Answer Method**. This prompt answer method answers the prompt using the object selected in the source (in this case, region).



If **Answer Dynamically** is not an option in the list, the Region prompt is not an attribute element prompt.

- 9 By default, the link is named Link1. To help analysts identify the link, rename it. Click **Rename**, and type **Revenue by Category** over **Link1**.
- 10 Click **OK** to return to the report.

Save the report

- 11 Save the report, naming it **Regional Revenue**.

To use the link

- 1 To use the link, do one of the following:
 - In MicroStrategy Developer, execute the **Regional Revenue** report, then double-click a region.
 - Log in to MicroStrategy Web, execute the **Regional Revenue** report, and then click a region.

The Revenue by Category report is executed, and displays only the data for the selected region.

Creating linked reports that answer prompts using metric conditions

The following steps re-create the example shown in *Passing metric conditions: Using the Static element list answer prompt method, page 574*. The **Category Revenue by Region and Customer** report contains multiple links to the **Revenue Data** report. When a user clicks one of the revenue metrics in MicroStrategy Web or double-clicks a revenue metric in MicroStrategy Developer, the **Revenue Data** report is executed, and data for only the category of the metric is shown. This is the default link, which is called **Category Revenue Data**, where *Category* is replaced by the category of the metric.

The revenue metrics also have a second link to the same report. This link, which is called **Category Region and Customer Revenue Data**, passes the selected region and customer to the source, as well as the category of the metric. For example, if a user selects Books Revenue for Warner Aadland in the Central region, and uses the Books Region and Customer Revenue Data link, the target report displays data for Warner Aadland, Central, and Books only.



Although the source report in the example contains three revenue metrics, these steps use only two, to simplify the process. Also, both links are created at once, while the example replaces the Category Revenue Data link with the **Category Region and Customer Revenue Data** link.

The high-level steps to create this example are listed below:

- 1 [Create the filters for Books and Movies, page 593](#)
- 2 [Create the conditional metrics, page 594](#)
- 3 [Create the prompts for category, region, and customer, page 595](#)
- 4 [Create the prompted target report \(Revenue Data\), page 596](#)
- 5 [Create the source report \(Category Revenue by Region and Customer\), page 597](#)

Create the filters for Books and Movies

Each conditional metric (Books Revenue and Movies Revenue) contains a filter for the appropriate category.

To create the filters for Books and Movies

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then select **Filter**. The Filter Editor opens.

If the New Filter dialog box is displayed, click the **Empty Filter** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 In the Object Browser, double-click **Products**, then double-click **Category**.
- 3 Drag **Books** from the Object Browser to the Filter definition area, which then displays **Category in List (Books)**.
- 4 Click **Save and Close** on the toolbar.
- 5 Name the filter **Books**.
- 6 Repeat the steps above, replacing **Books** with **Movies** in steps 3 and 5.

Create the conditional metrics

Next, create the conditional metrics. A conditional metric contains a filter; in this example, they are the filters on category (Books and Movies) created in the previous procedure.

To create conditional metrics for category revenue

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then select **Metric**. The Metric Editor opens.

If the New Metric dialog box is displayed, click the **Empty Metric** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).
- 2 In the Object Browser, double-click **Revenue** (the fact).
- 3 In the top part of the metric definition (under **Metric New Metric is defined as**), click **Condition**. The Condition definition pane is displayed below the Metric component pane.
- 4 In the Object Browser, find the **Books** filter. Drag and drop it in the Condition definition pane.
- 5 Click **Save and Close** on the toolbar.
- 6 Name the metric **Books Revenue**.

- 7 Repeat the steps above, replacing the Books filter with the **Movies** filter in step 4 and naming the metric **Movies Revenue**.

Create the prompts for category, region, and customer

These prompts will be used on the target report to determine the category, region, and customer to display. When the **Category Revenue Data** link from the source report to the target is used, the category prompt is answered by the category in the static element list of the link. The region and customer prompt is ignored in that link.

For the **Category Region and Customer Revenue Data** link, the category prompt is answered in the same way, but the region and customer selected in the source report are passed to the target.



In the names of the links, *Category* is replaced by the category of the metric.

These are attribute element list prompts, to allow attribute elements to be passed to the target report.

To create attribute element list prompts

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then choose **Prompt**. The Prompt Generation Wizard opens.
- 2 Select the **Filter definition prompt** option. Then, in the list of filter definition prompts, select **Choose from an attribute element list**, and click **Next**.
- 3 Type **Category** in the **Attribute** field and click **Next**.
- 4 Select **List all elements** and click **Next**.
- 5 In the **Title** field, type **Category**.
- 6 Clear the **Prompt answer required** check box.
- 7 Click **Finish**. The Save As dialog box opens.
- 8 Click **Save**.

- 9 Repeat the steps above twice.
 - The first time, replace **Category** with **Region** in steps 3 and 5.
 - The second time, replace **Category** with **Customer** in steps 3 and 5.

Create the prompted target report (Revenue Data)

The target report is the report that is executed from the original report (the source). The target report is filtered by the prompts on category, region, and customer that you created in the previous procedure.

To create a prompted report

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then select **Report**. The Report Editor opens.

If the New Grid dialog box is displayed, click the **Empty Report** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Add **Region**, **Customer**, and the following metrics to the report:
 - Revenue
 - Cost
 - Profit
 - Units Sold
 - Revenue Per Employee
- 3 Add the **Category**, **Region**, and **Customer** prompts that you created in the previous procedure to the report filter.
- 4 Save and close the report, naming it **Revenue Data**.

Create the source report (Category Revenue by Region and Customer)

The source report is the report that contains the links to the target. The links in this source report pass the category of the selected conditional metric to answer the prompt in the target. One link passes the selected region and customer as well, while the other ignores the region and customer prompts in the target.

To create a report that links to another report

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then select **Report**. The Report Editor opens.

If the New Grid dialog box is displayed, click the **Empty Report** icon. If you do not want this dialog box to be shown in the future, select **Don't show this dialog in the future**. Click **OK**. For a full description of object templates, including a list of the objects that can use object templates, see [Object templates, page 357](#).

- 2 Add **Region**, **Customer**, the **Books Revenue** metric, and the **Movies Revenue** metric to the report. (These are the metrics that you created in the previous procedures.)

Create the link that passes category only from the Books Revenue metric

- 3 Right-click **Books Revenue** and select **Edit Links**. The Link Editor opens.
- 4 Click ... (**Browse**) next to **Run this report or document**. The Select Target dialog box opens.
- 5 Navigate to the **Revenue Data** report, select it, and click **Open**. This report is the target of the link.
- 6 Select **Category** in the **Target prompts** list. This prompt on the **Revenue Data** report filters the target report by category, and should be answered with the category in the conditional metric. Use a static element list to pass that year to the target.
- 7 Choose **Static element list** from the drop-down list under **Select Prompt Answer Method**. This prompt answer method answers the prompt using a list that you will create in the next step. The Element List area becomes available.

- 8 Click **Add**. The Select Elements dialog box opens.
- 9 Select **Books** (because this is the link from the **Books Revenue** metric) from the list of **Available objects** and click > to move it to the list of **Selected objects**.
- 10 Click **OK** to return to the Link Editor.
- 11 Select **Region** in the **Target prompts** list. This prompt on the **Revenue Data** report filters the target report by region. Since all regions should be included in the target, this prompt should be ignored.
- 12 Choose **Empty answer** from the drop-down list under **Select Prompt Answer Method**. This prompt answer method ignores the prompt, which means that the Region prompt is not answered. No prompt answer is provided from the source and the user is not prompted to provide answers.
- 13 Select **Customer** in the **Target prompts** list. This prompt on the **Revenue Data** report filters the target report by customer. Since all customers should be included in the target, this prompt should be ignored.
- 14 Choose **Empty answer** from the drop-down list under **Select Prompt Answer Method**.
- 15 By default, the link is named Link1. To help analysts identify the link, rename it. Click **Rename**, and type **Books Revenue Data** over **Link1**. Including the name of the report (in this case, a part of the full name) and a description of how its prompts are answered (in this case, with Books only) helps identify what the target will be.

Create the link that passes category, region, and customer from the Books Revenue metric
- 16 Click **New** to start a new link.
- 17 By default, the link is named Link2, but you can rename it now. Type over the name with **Books Region and Customer Revenue Data**, and press ENTER. Including the name of the report (in this case, a part of the full name) and a description of how its prompts are answered (in this case, with Books, and the selected region and customer) helps identify what the target will be.
- 18 Click ... (**Browse**) next to **Run this report or document**. The Select Target dialog box opens.

- 19 Navigate to the **Revenue Data** report, select it, and click **Open**. This report is the target of the link.
- 20 Select **Category** in the **Target prompts** list. This prompt should be answered with the year in the conditional metric. Use a static element list to pass that category to the target.
- 21 Choose **Static element list** from the drop-down list under **Select Prompt Answer Method**. The Element List area becomes available.
- 22 Click **Add**. The Select Elements dialog box opens.
- 23 Select **Books** from the list of **Available objects** and click **>** to move it to the list of **Selected objects**.
- 24 Click **OK** to return to the Link Editor.
- 25 Select **Region** in the **Target prompts** list. This prompt on the **Revenue Data** report filters the target report by region. Only the region selected on the source should be included in the target.
- 26 Choose **Dynamically** from the drop-down list under **Select Prompt Answer Method**. This prompt answer method answers the prompt using the object selected in the source (in this case, the region).
- 27 Select **Customer** in the **Target prompts** list. This prompt on the **Revenue Data** report filters the target report by category. Only the customer selected on the source should be included in the target.
- 28 Choose **Dynamically** from the drop-down list under **Select Prompt Answer Method**.

Copy the link from Books Revenue to Movies Revenue

The links from the Movies Revenue metric are the same as those on the Books Revenue metric, except for the category. Rather than re-creating the links, copy them.

- 29 Click **Copy Links**. The Copy Links dialog box opens.
- 30 In the list under **Select the links to be copied**, select **Books Revenue Data** and **Books Revenue and Customer Revenue Data**.
- 31 In the list under **Select the units to which the selected links will be copied**, select **Movies Revenue**.

32 Click **Copy**. You are returned to the Link Editor.

33 Click **OK** to return to the report.

Modify the copied links

The links from the Books Revenue metric are now on the Movies Revenue metric as well. The category must be changed from Books to Movies in the link names and element lists.

34 Right-click **Movies Revenue** and select **Edit Links**. The Link Editor opens.

35 Select **Books Revenue Data** in the list of links.

36 Click **Rename**.

37 Replace **Books** with **Movies** in the name area, since this link is on the Movies Revenue metric. Press ENTER.

38 Select **Category** in the **Target prompts** list.

39 In the Element List area, select **Modify**. The Select Elements dialog box opens.

40 Select **Books** in the **Selected objects** list and click < to move it to the **Available objects** list.

41 Select **Movies** in the **Available objects** list and click > to move it to the **Selected objects** list. Now Movies rather than Books will be passed to the target.

42 Click **OK** to return to the Link Editor.

43 Select **Books Region and Category Revenue Data** in the list of links.

44 Repeat steps 36 through 42 above, to replace Books with Movies in the link name and the element list.

45 Click **OK** to return to the report.

Save the report

46 Save the report, naming it **Category Revenue by Region and Customer**.

To use the links

- 1 To use the links, do one of the following:
 - In MicroStrategy Developer, execute the **Category Revenue by Region and Customer** report. To use the default link (*Category Revenue Data*) which passes only the category, not region or customer, double-click a revenue amount. To use the link that passes category, region, and customer, right-click a revenue amount, point to **Link**, and then select **Category Region and Customer Revenue Data**.
 - Log in to MicroStrategy Web and execute the **Category Revenue by Region and Customer** report. To use the default link (*Category Revenue Data*) which passes only the category, not region or customer, click a revenue amount. To use the link that passes category, region, and customer, right-click a revenue amount, point to **Links**, and then select **Category Region and Customer Revenue Data**.
-  In the names of the links, *Category* is replaced by the category of the metric.

CUSTOM SQL QUERIES: FREEFORM SQL AND QUERY BUILDER

Introduction

Freeform SQL and Query Builder provide two alternative methods to access your data and begin analysis with MicroStrategy. Freeform SQL allows you to write your own SQL statements to run directly against a data warehouse or operational data store, giving you full control over accessing your data. Query Builder provides you with a graphical user interface that helps guide you on building SQL queries that can adapt to different data models.

These two features allow you to run queries against ODBC data sources that are not easily modeled to an attribute and fact schema. This includes databases that are a collection of flat tables rather than being defined into fact and lookup tables. You can also save a great amount of time setting up a project to use with Freeform SQL or Query Builder because you are not required to create an attribute and fact schema.

Additionally you can connect to third-party web services to retrieve data and report on that data within MicroStrategy. This is accomplished by using the MicroStrategy Freeform SQL Editor to write XQuery statements to access and retrieve information from a web service.

This chapter discusses different aspects of the Freeform SQL and Query Builder features, including the following:

- *Customize your SQL statements: Freeform SQL, page 604*
- *Reporting on third-party web services with XQuery, page 638*
- *Retrieving web folder resources with XQuery, page 664*
- *Updating data with Transaction Services, page 675*
- *Customize your SQL queries: Query Builder, page 694*
- *Connect to databases, Excel files, and text files, page 731*
- *Map data to non-project objects: Managed objects, page 735*
- *Creating Intelligent Cubes, page 740*

Customize your SQL statements: Freeform SQL

The Freeform SQL functionality adds great flexibility to MicroStrategy's query and reporting capabilities. Traditionally, you use the MicroStrategy Engine to generate SQL to run against one specific relational database to get a desired report. Starting from MicroStrategy version 8.0, in addition to generating reports in the traditional way, you can also use your own customized SQL statements to generate reports from operational systems included in a MicroStrategy project. This capability can save you time since you do not need to place the data into a data mart or data warehouse first.

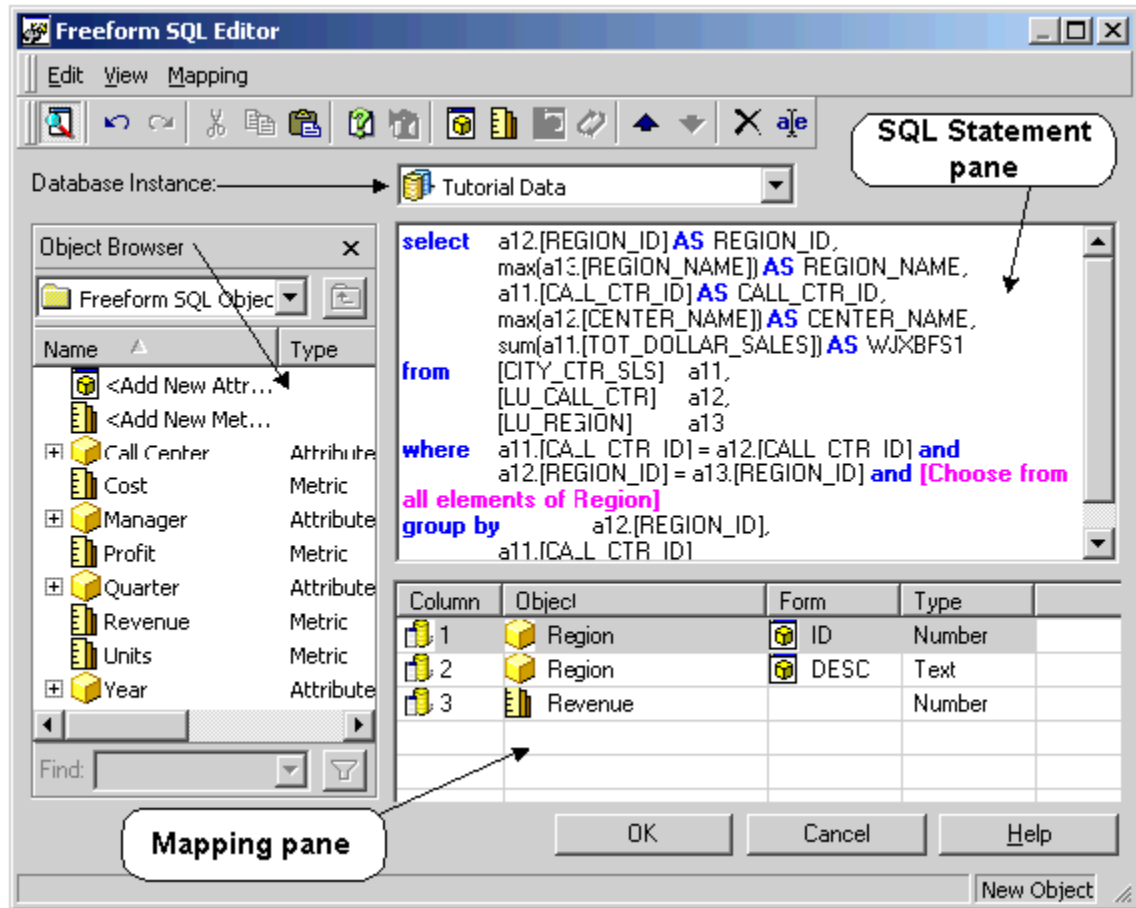


Reports that are built using the Freeform SQL feature are referred to as Freeform SQL reports in this chapter.

The Freeform SQL feature allows you to use your own SQL statements to access data from various ODBC data sources, including relational databases, Excel files, and flat files, as long as they are included in the MicroStrategy environment. Since you create your own SQL statements to create reports with Freeform SQL, a strong knowledge of how to create and use SQL statements is essential.

Details on how to create Freeform SQL reports from these ODBC data sources are discussed in this chapter.

The following image is an illustration of the Freeform SQL Editor, where you define the SQL statement for the report. Notice the different panes for different purposes.



Usage scenarios and SQL standards

How to use the Freeform SQL feature effectively depends on your work environment. As with every other MicroStrategy functionality, before you start using this feature, you need to assess your particular work situation and find a way to strike a good balance between project maintainability and fast-paced development. For example, building three or four Freeform SQL reports could be very valuable, but building 100 such reports could make maintenance and testing very difficult.

Whether you should use the Freeform SQL feature to build reports at all is another question that you should ask yourself. You may want to consider using this feature if you are in one of the situations discussed below. For

information on using the Freeform SQL Editor to report on third-party web services, see [Reporting on third-party web services with XQuery, page 638](#).

When should I use the Freeform SQL feature?

If your company is accustomed to creating static reports using customized SQL to retrieve data from a certain ODBC data source, and especially if your SQL queries have worked well in the past, then you may want to simply use MicroStrategy to deploy those reports to your users. There is no need to recreate the SQL with the MicroStrategy Engine, as is done if the data is moved to a data warehouse for report generation. Freeform SQL allows you to use your own SQL statements rather than generating new and possibly different SQL statements.

If you have existing stored procedures that have proven to be successful, then you can continue to use them to generate MicroStrategy reports. One important thing to note is that you must know what data the stored procedure is supposed to retrieve because this information is essential in building a Freeform SQL report. Specifically, you need to know the number of columns, column names, and their data types, all of which are necessary for mapping the columns to MicroStrategy objects.

Another situation for which you might want to use Freeform SQL reporting is when you need to run queries against a set of OLTP tables that are not set up for OLAP analysis. Data warehouses that do not lend themselves well to MicroStrategy's attribute and fact schema can be accessed with Freeform SQL and mapped to new MicroStrategy objects automatically created by the Freeform SQL feature. As for all the Freeform SQL reports, connecting to the desired ODBC data source is a prerequisite.

Freeform SQL can also be a helpful tool to perform database maintenance tasks such as updates and inserts. For information on this scenario see [Using Freeform SQL for updates, inserts, and other database maintenance tasks, page 611](#).

SQL query syntax

Well-written SQL queries are the key to building successful Freeform SQL reports. To take full advantage of the Freeform SQL feature, MicroStrategy recommends that you ensure the correctness and validity of your SQL statements before creating any reports using Freeform SQL. MicroStrategy does not offer consultation or technical support for the syntax or composition of your SQL queries.

Depending on your needs, you can compose SQL statements in several ways:

- Create new SQL statements from scratch.
- Use existing SQL statements that you previously defined, which have proven to be successful in terms of retrieving data from the data source.



This includes using stored procedures.

- Tune the MicroStrategy Engine-generated SQL by modifying it to suit your needs.

This means you can re-use the Engine-generated SQL by changing some of its clauses or syntax to get a different result set.

You cannot tune the Engine-generated SQL that involves the use of the Analytical Engine. Typically, the Analytical Engine comes into play for metric qualifications using analytical functions (such as OLAP functions), custom group banding, use of the plug-ins, and so on. If the Analytical Engine is used for any part of the SQL during the report generation, that part is labeled as “[An Analytical SQL]” in the report SQL.



All MicroStrategy functions, including the ones that use the Analytical Engine, are described in detail in the *Analytical Functions Reference*.

SQL support

With the Freeform SQL feature, you can use both single-pass and multi-pass SQL queries to generate reports. However, you must convert your multi-pass SQL statements into derived table or common table expression syntax that can be completed in a single pass.

Derived table and common table expressions can have only one SELECT clause in the SQL query. This means that a report SQL statement with Pass 1, Pass 2, Pass 3, and so on, as can be found in many MicroStrategy Tutorial reports, cannot yield the desired report, unless you modify the query using derived tables or common table expressions. Check the database that you use to ensure that derived tables, common table expressions, or both are supported.

For example, if you have the following multi-pass SQL statement, it needs to be converted to derived table or common table expression syntax shown below.

Multi-pass SQL

```
create table MQ00 (
    A1 INTEGER,
    B1 INTEGER)
insert into MQ00
select a11.[A1] AS A1,
    a11.[B1] AS B1
from [T1] a11
    where (a11.[M1] > 0);

select a11.[A1] AS A1,
    a11.[B1] AS B1,
    a12.[B2] AS B2,
    a11.[M1] AS M1
from [T1] a11,
    [MQ00] pa1,
    [T2] a12
where a11.[A1] = pa1.[A1] and
    a11.[B1] = pa1.[B1] and
    a11.[B1] = a12.[B1]

drop table MQ00
```

Equivalent derived table

```
select a11.[A1] AS A1,
    a11.[B1] AS B1,
    a12.[B2] AS B2,
    a11.[M1] AS M1
from [T1] a11,
    (select a11.[A1] AS A1,
        a11.[B1] AS B1
    from [T1] a11
    where a11.[M1] > 0) pa1,
    [T2] a12
where a11.[A1] = pa1.[A1] and
    a11.[B1] = pa1.[B1] and
    a11.[B1] = a12.[B1]
```

Equivalent common table expression

```
with pa1 as
(select a11.[A1] AS A1,
    a11.[B1] AS B1
from [T1] a11
```

```
        where a11.[M1] > 0)
select a11.[A1] AS A1,
       a11.[B1] AS B1,
       a12.[B2] AS B2,
       a11.[M1] AS M1
from [T1] a11,
     pa1,
     [T2] a12
where a11.[A1] = pa1.[A1] and
      a11.[B1] = pa1.[B1] and
      a11.[B1] = a12.[B1]
```

Freeform SQL reports vs. standard reports

A Freeform SQL report is simply a standard report, which has been built using the Freeform SQL feature. Due to some variations in how standard reports and Freeform SQL reports are built, there are also some varying functionalities between the two.

You can create Freeform SQL reports using your own SQL queries against a data warehouse, or from an Excel file, or a flat text file. Information on how to create these reports is provided later in this chapter. Although Freeform SQL reports can only be created on MicroStrategy Developer, once created, they can be executed from both MicroStrategy Developer and MicroStrategy Web like any other MicroStrategy standard reports. Functions that you can perform on MicroStrategy standard reports can also be performed on Freeform SQL reports, including the following:

- Sorting
- Auto-styles
- Caching
- Exporting
- Thresholds
- Graphing
- Filtering
- Value prompts
- Shortcut-to-a-report qualification. For more information on using a Freeform SQL report as a shortcut-to-a-report qualification, see [Using Freeform SQL reports to filter other reports, page 628](#).

- Narrowcast Server subscriptions and report execution
- Object security
- OLAP services
- Prioritization
- Report Services documents
- Scheduling
- Subtotals

The following features are available for use with Freeform SQL reports if an attribute and fact schema is used to model the data returned from your queries:

- Element list prompts
- Security filters

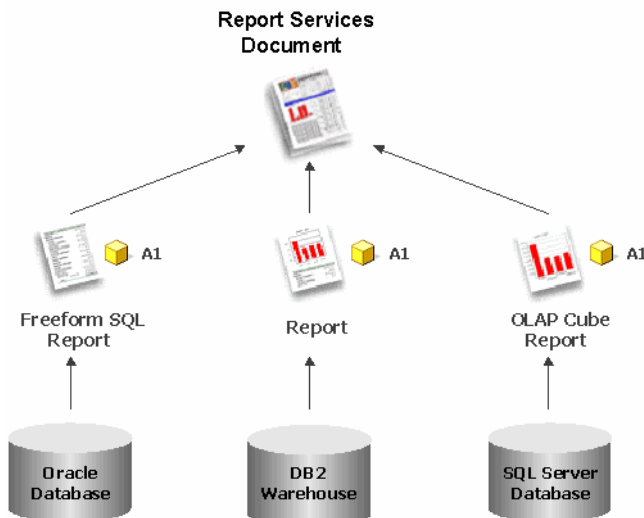
The following features do not apply to Freeform SQL reports:

- Custom groups
- Consolidations
- Transformations
- Existing filters
- Save as template/filter
- Data marting

Freeform SQL reports in Report Services documents

Once created, Freeform SQL reports can be included in Report Services documents in the same way as standard reports. The same document can also contain reports from other data sources, such as MDX cube reports. For information regarding MDX cube reports, refer to the *MDX Cube Reporting Guide*.

For data to be joined across different data sources in a document, a common attribute is needed across the datasets. In the following diagram, the common attribute which appears on each of the three reports is named A1.



You can establish a common attribute by mapping objects, such as attributes and prompts, retrieved from different data sources to existing objects in the MicroStrategy environment. Information on mapping columns for Freeform SQL reports is provided in [Mapping columns to metadata objects, page 618](#).

For example, in a Report Services document, you have three datasets from three reports: one standard MicroStrategy report, one Freeform SQL report, and one MDX cube report using SAP BI as the MDX cube source. All three reports use the same attribute, Product. This means that Product is used in the standard report as a project attribute, the Freeform SQL report has one object mapped to Product, and the MDX cube report uses Product to map one of its characteristics from the imported SAP BI query cube. Because data is joined by the common attribute Product, the document is generated successfully.



If each of the three reports originally has a prompt on Product, then the prompt is only displayed one time when a user executes the document; this means a user only needs to answer the prompt one time, instead of three times.

Using Freeform SQL for updates, inserts, and other database maintenance tasks

While Freeform SQL is designed to return data from a data source to be displayed on a report, it can also be used for database maintenance tasks such as updates and inserts. This can be achieved because Freeform SQL

allows you the flexibility to create the SQL statements that are run against your database.

Database maintenance actions such as update, insert, and create table actions do not return any results. By default, Freeform SQL reports return a warning when no result is returned. If you routinely use Freeform SQL for these purposes, you can hide these warnings since an empty result set is expected. The procedure below shows you how to disable these warnings and use Freeform SQL for database maintenance tasks.

To use Freeform SQL for updates, inserts, and other database maintenance tasks

- 1 In MicroStrategy Developer, on the **File** menu, point to **New**, and choose **Report**. The New Grid dialog box opens.
- 2 On the **Freeform Sources** tab, select **Create Freeform SQL report**.
- 3 In the **Source** area, select a database instance for the database to access using Freeform SQL. For information on connecting to databases, see [Connect to databases, page 732](#).
- 4 Click **OK**. The Freeform SQL Editor opens.
- 5 Create the SQL statement for the database maintenance task.
- 6 Click **OK** to complete the Freeform SQL statement definition. The Report Editor opens.
- 7 From the **Data** menu, select **VLDB Properties**. The VLDB Properties Editor opens.
- 8 In the **VLDB Settings** list, expand the **Freeform SQL** folder, and select **Ignore Empty Result for Freeform SQL**.
- 9 Clear the **Use default inherited value** check box.
- 10 Select **Turn off warnings for Freeform SQL statements with empty results, such as updates** to hide all warnings when a Freeform SQL statement causes a Freeform SQL report to return an empty result. This prevents users from seeing a warning every time a SQL statement to perform a database maintenance task is executed using Freeform SQL.

- 11 Click **Save and Close** to save your changes and return to the Freeform SQL report.
- 12 From the toolbar, click the **Run Report** icon to execute the Freeform SQL report, which executes the database maintenance action you defined.

Access and analyze multiple data sources with Freeform SQL

With Freeform SQL, you can access multiple data sources in the same project and use MicroStrategy reporting features to analyze your data. Connection to databases, Excel files, and text files is described in [Connect to databases, Excel files, and text files, page 731](#).

For information on using the Freeform SQL Editor to report on third-party web services as a data source, see [Reporting on third-party web services with XQuery, page 638](#).

Once you connect to a valid data source, you can create a Freeform SQL report to retrieve and analyze the data from the data source. This section describes the process of using Freeform SQL to:

- [Create a Freeform SQL report to access databases, Excel files, or text files, page 613](#)
- [Create a Freeform SQL report using a stored procedure, page 616](#)
- [Mapping columns to metadata objects, page 618](#)

Create a Freeform SQL report to access databases, Excel files, or text files

After you create a connection to your data source, you can create a Freeform SQL that queries that data source.

The steps to create a Freeform SQL report from a database, Excel file, and text file are the same, except for which data source the report connects to. For steps to create a Freeform SQL report from a stored procedure, see [Create a Freeform SQL report using a stored procedure, page 616](#).

Prerequisites

- You must connect Query Builder to a database, Excel file, or text file, which is described in the sections listed below:
 - [Connect to databases, page 732](#)
 - [Connect to Excel files, page 733](#)
 - [Connect to text files, page 734](#)
- Freeform SQL reports can be created on MicroStrategy Developer only. However, these reports can be manipulated and executed from both MicroStrategy Developer and Web. Access to the Freeform SQL Editor is available only to Developers with the Define Freeform SQL Report privilege and those with the Create schema objects privilege.

To create a Freeform SQL report

- 1 In MicroStrategy Developer, from the **File** menu, select **New**, and then choose **Report**. The New Grid dialog box opens.
- 2 On the **Freeform Sources** tab, in the **Source** area, select a database instance for the data source to access using Freeform SQL. The sections listed below provide the configuration steps for the different data sources:
 - [Connect to databases, page 732](#)
 - [Connect to Excel files, page 733](#)
 - [Connect to text files, page 734](#) If a database instance is not defined in the metadata, the Freeform SQL Editor cannot be loaded and a message is displayed.
- 3 Select **Create Freeform SQL report** and click **OK**. The Freeform SQL Editor opens.
- 4 In the SQL Statement pane (the top pane on the right), type in your SQL query. Be aware of the following when creating a SQL statement:
 - Column names in the SQL statement have to match the column names in the database, column headers in the Excel file, or the field names in the text file.
 - The column names you use in your SQL query are NOT case sensitive.

- When connecting to Excel and text files, you do not use the file name as the table name for the “From” clause. You must use the table names you defined when creating and connecting to the files. Remember that the Excel file or text file is the data source that contains the tables, which are named individually.
- 5 In the Mapping pane (the bottom pane on the right), map the columns in your SQL statement to attributes and metrics to be used in the report. Be aware of the following when mapping columns in your SQL statement to attributes and metrics:
 - When mapping the columns, it is important that you follow the same sequence of the columns as they appear in the SQL statement. Doing otherwise causes the report to fail.
 - Make sure that the number of mappings is the same as the number of columns in the SQL statement. For example, if your SQL statement lists 10 columns from which to retrieve data, you should map these columns to exactly 10 attributes or metrics.
 - You must map a column to the ID form for each attribute you use in your Freeform SQL report.
 - 6 Insert prompts into the SQL statement, if needed. For information on including prompts in Freeform SQL reports, see [Prompts, page 624](#).
 - 7 Insert security filter qualifications, if needed. For more information on inserting security filter qualifications in Freeform SQL reports, see [Security filters, page 633](#).
 - 8 Click **OK** to close the Freeform SQL Editor. The Report Editor opens.
 - 9 Format and define the Freeform SQL report in the same way as you would a standard report, using features such as sorting, view filters, thresholds, exporting, and so on.
 - 10 From the **File** menu, click **Save As**. The Save Report As dialog box opens.

 You must save the report before you can run it, otherwise a message is displayed.
 - 11 Enter a name for the report and click **Save**.
 - 12 Run the Freeform SQL report.

Create a Freeform SQL report using a stored procedure

Since stored procedures are run against databases, creating a Freeform SQL report using a successful stored procedure requires you to first connect to a database (see [Connect to databases, page 732](#)). Mapping columns to attributes and metrics requires prior knowledge of your data warehouse structure. Although the stored procedure itself does not display any column names, you need to know in advance what columns will be retrieved once the procedure is executed. Without this knowledge, it may be difficult for you to map the columns.

For example, if you use the following stored procedure:

```
Execute sp_customer_profit
```

you may need to map the columns to the following MicroStrategy objects:

- Customer ID
- Customer DESC
- Customer City ID
- Customer City DESC
- Profit

Below are the general steps you need to take when you use a stored procedure to create a Freeform SQL report.

Prerequisites

- Freeform SQL reports can be created on MicroStrategy Developer only. However, these reports can be manipulated and executed from both MicroStrategy Developer and Web. Access to the Freeform SQL Editor is available only to Developers with the Define Freeform SQL Report privilege and those with the Create schema objects privilege.
- You must connect Freeform SQL to a database, which is described in [Connect to databases, page 732](#).

To use a stored procedure to create a Freeform SQL report

- 1 In MicroStrategy Developer, on the **File** menu, point to **New**, and then select **Report**. The New Grid dialog box opens.

- 2 On the **Freeform Sources** tab, select **Create Freeform SQL report**.
- 3 In the **Source** area, select the database instance you created for Freeform SQL reports with stored procedures.
- 4 Click **OK**. The Freeform SQL Editor opens.
- 5 In the SQL Statement pane (the top pane on the right), type in your stored procedure.



Different databases use different syntax to execute stored procedures. Information on stored procedure execution for some major databases is given in the table below.

Database	Stored procedure	Notes
DB2	call <i>stored_procedure_name</i> with DB2 ODBC driver	The stored procedure must have been created indicating that it has a result set. The results will be sent back to the client.
Oracle	call <i>stored_procedure_name</i> () with MicroStrategy ODBC driver {call <i>stored_procedure_name</i> } with Oracle ODBC driver	The stored procedure must return the results into a table that can subsequently be selected.
SQL Server	exec <i>stored_procedure_name</i> with SQL Server ODBC driver	The stored procedure returns the data to the client. No particular precaution is needed.

- 6 In the Mapping pane (the bottom pane on the right), map the columns that are retrieved by the stored procedure to attributes and metrics that are used in the MicroStrategy report.
- 7 Click **OK** to close the Freeform SQL Editor. The Report Editor opens.
- 8 From the **File** menu, click **Save As**. The Save Report As dialog box opens.



You must save the report before you can run it, otherwise a message is displayed.

- 9 Enter a name for the report and click **Save**.
- 10 Run the Freeform SQL report.

Mapping columns to metadata objects

You use the Mapping pane in the Freeform SQL Editor to map the columns in your SQL query to attributes and metrics included in a MicroStrategy report.

You can map columns to:

- **Managed object attributes and metrics:** Managed objects are created on the fly and only exist in the context of Freeform SQL and Query Builder reports. These objects are stored in the Freeform Objects folder, and are known as managed objects. Managed objects enable you to create a Freeform SQL or Query Builder report quickly, without having to take the time to manually create or map data to schema objects.

For more information on managed objects and how they are used in MicroStrategy, see [Map data to non-project objects: Managed objects, page 735](#).



Managed objects are also created when integrating MDX cube sources into MicroStrategy. MDX cube sources include SAP BW, Microsoft Analysis Services, Hyperion Essbase, and IBM Cognos TM1. For information on integrating MDX cube sources into MicroStrategy, see the *MDX Cube Reporting Guide*.

- **Project attributes:** These attributes exist in the project as part of the relational schema, and therefore can be used to generate standard reports. Mapping columns in Freeform SQL reports to project attributes enables you to present a consistent representation of data to report analysts. This also allows you to use Freeform SQL reports as filters on standard reports, as described in [Using Freeform SQL reports to filter other reports, page 628](#).

Project attributes also facilitate joins with standard MicroStrategy reports in a Report Services document. This capability enables you to create many types of documents. For example, you can create a Report Services document with two datasets that share a common project attribute. One retrieves historical data from a data warehouse, and the other contains current data using Freeform SQL from an operational system.

Unlike attributes, which can be mapped to project attributes, all metrics are created or mapped to managed object metrics. The only existing metrics you can use are managed object metrics that are defined for Freeform SQL and Query Builder reports in the current project.

For information on mapping attributes and metrics to columns in your Freeform SQL statement, see the sections below:

- [Mapping a new attribute form to a column, page 619](#)
- [Mapping a new metric to a column, page 620](#)
- [Mapping an existing object to a column, page 621](#)
- [Renaming an object mapped to a column, page 622](#)
- [Replacing an object mapped to a column, page 622](#)

Mapping a new attribute form to a column

You can map a column in a SQL statement to a new attribute form in the MicroStrategy environment. All new attribute forms are mapped to managed objects, which are discussed in [Map data to non-project objects: Managed objects, page 735](#).

By default, every newly added attribute uses the ID form. You can change the column mapped to the ID form by using the Forms drop-down list in the Mapping pane of the Freeform SQL Editor.

Prerequisites

- The procedure below assumes that you have already created a Freeform SQL report and that it is open in the Freeform SQL Editor.
- You must have the Define Freeform SQL Report and Create Schema Objects privileges to access and use the Freeform SQL Editor.

To map a new attribute form to a column

- 1 From the **Mapping** menu, select **Add New Attribute Form**. An attribute form is inserted into the Mapping pane.



An attribute can have multiple attribute forms, and every attribute must map a column to the ID attribute form.

- 2 Type in the object name of your choice and make the selection for Form and Type.

You can show and hide column data types along with other pertinent information. From the **View** menu, select **Show Column Data Type**. This

option displays the full data type information for the column, including Data Type, Precision/Length, and Scale.

- 3 Ensure that the column number for the object matches the order of the column in the SQL statement. For example, an object is to be mapped to the second column listed in the `SELECT` clause. In the Mapping pane, the object must be included as column 2. You can use the Move Up and Move Down options in the toolbar to change the column position of an object.



Note the following:

- Use the DB Query Tool to test the validity of the stored procedure before creating any Freeform SQL report.
- If you use a stored procedure, it is important that you know its content in advance, including exactly which columns the stored procedure retrieves. Otherwise, it may be difficult for you to correctly map the columns retrieved.

Mapping a new metric to a column

You can map a column in the SQL statement to a new metric that can be recognized in the MicroStrategy environment. All new metrics are mapped to managed objects, which are discussed in [Map data to non-project objects: Managed objects, page 735](#).

Prerequisites

- The procedure below assumes that you have already created a Freeform SQL report and that it is open in the Freeform SQL Editor.
- You must have the Define Freeform SQL Report and Create Schema Objects privileges to access and use the Freeform SQL Editor.

To map a new metric to a column

- 1 From the **Mapping** menu, select **Add New Metric**. A new metric is inserted into the Mapping pane.
- 2 Type in the metric name of your choice and make the selection for Form and Type.

You can show and hide column data types along with other pertinent information. From the **View** menu, select **Show Column Data Type**. This

option displays the full data type information for the column, including Data Type, Precision/Length, and Scale.

- 3 Ensure that the column number for the object matches the order of the column in the SQL statement. For example, an object is to be mapped to the second column listed in the `SELECT` clause. In the Mapping pane, the object must be included as column 2. You can use the Move Up and Move Down options in the toolbar to change the column position of an object.

Mapping an existing object to a column

By mapping existing objects to columns in Freeform SQL reports, you can avoid creating duplicate copies of the same object. Mapping with a project object also serves the purpose of joining data when the object is used in a Report Services document that contains multiple datasets. Using project objects also enables you to use security filters with your Freeform SQL reports.

An existing object can be:

- An attribute defined in an existing MicroStrategy project that is part of a relational schema.
- A managed object attribute that has been created for previous Freeform SQL reports or Query Builder reports in the current project.
- A managed object metric that has been created for previous Freeform SQL reports or Query Builder reports.



A metric defined in an existing MicroStrategy project that is not a managed object cannot be used for mapping in Freeform SQL reports or Query Builder reports.

Prerequisites

- The procedure below assumes that you have already created a Freeform SQL report and that it is open in the Freeform SQL Editor.
- You must have the Define Freeform SQL Report and Create Schema Objects privileges to access and use the Freeform SQL Editor.

To map an existing object to a column

- 1 From the **Mapping** menu, select **Insert**. The Select an Object dialog box opens.

- 2 Browse to and select an attribute or metric, and click **OK**. You are returned to the Freeform SQL Editor and the object is mapped to a new column.
- 3 Ensure that the column number for the object matches the order of the column in the SQL statement. For example, an object is to be mapped to the second column listed in the `SELECT` clause. In the Mapping pane, the object must be included as column 2. You can use the Move Up and Move Down options in the toolbar to change the column position of an object.

Renaming an object mapped to a column

You can use the Rename option to change the name of an attribute or a metric in the Object field in the Mapping pane. Only new attributes and metrics can be renamed. You cannot rename those that are mapped to existing attributes and metrics that have already been saved to the metadata, that is those that already exist in the Freeform Objects folder or in any folder displayed in the Object Browser window. This reduces the number of duplicate metadata objects.

Prerequisites

- The procedure below assumes that you have already created a Freeform SQL report and that it is open in the Freeform SQL Editor.
- You must have the Define Freeform SQL Report and Create Schema Objects privileges to access and use the Freeform SQL Editor.

To rename an object mapped to a column

- 1 In the Mapping pane, right-click the row for the object you want renamed, and select **Rename**.
- 2 Type the new name for the object.

Replacing an object mapped to a column

You can replace an object mapped to a column with a different existing object. This can be helpful if you previously used managed object attributes to map your Freeform SQL data, but now you want to use project attributes that are part of your relational schema.

Prerequisites

- The procedure below assumes that you have already created a Freeform SQL report and that it is open in the Freeform SQL Editor.
- You must have the Define Freeform SQL Report and Create Schema Objects privileges to access and use the Freeform SQL Editor.

To replace an object mapped to a column

- 1 In MicroStrategy Developer, browse to a Freeform SQL report, right-click the report, and select **Edit**. The Report Editor opens.
- 2 If the report is not in Design view, switch the report to Design view.
- 3 From the **Data** menu, select **Freeform SQL Definition**. The Freeform SQL Editor opens.
- 4 In the Mapping pane, right-click the row for the object you want to replace, and select **Replace**. The Select an Object dialog box opens.
- 5 Select an object that you want to use and click **OK**. All references to the original object are replaced with the new object.

Reporting analysis features

This section discusses the following MicroStrategy reporting analysis features in relation to Freeform SQL reports:

- [Filters](#)
- [Prompts](#)
- [Drilling](#)
- [Using Freeform SQL reports to filter other reports](#)

Filters

A filter specifies the conditions that the data must meet to be included in the report results. For information on Filters in general, refer to [Chapter 3, Advanced Filters](#), in this guide.

You cannot use existing filters in a Freeform SQL report; however, you can still filter data by including `WHERE` and `HAVING` clauses in the SQL statement. You can even embed prompt objects in the `WHERE` or `HAVING` clause, if needed. For example,

```
WHERE Year_ID = Year_prompt
```

 Only two types of prompts can be used: value prompts and element list prompts. For more information, refer to [Prompts, page 624](#).

In addition, you can use the view filter functionality for Freeform SQL reports in the same way as for regular reports.

Prompts

A prompt is a MicroStrategy object that allows user interaction at report run time. For general information on prompts, refer to [Chapter 6, Advanced Prompts](#), in this guide.

For Freeform SQL reports, only two types of prompts are supported—value prompts and element list prompts. To add prompts, you can select from the two options on the **Edit** menu in the Freeform SQL Editor:

- **Add New Prompt:** launches the Prompt Generation Wizard, which allows you to create a new value prompt or an element list prompt.

 Note the following:

- Only project attributes can be used with element list prompts in Freeform SQL reports, as opposed to managed object attributes. For more information on managed objects, see [Map data to non-project objects: Managed objects, page 735](#).
 - Any prompt created this way is saved as a normal object in the metadata.
- **Insert Prompt:** displays the Select a Prompt dialog box where you can select an existing prompt that you have previously created in the project, either a value prompt or an element list prompt.

 You cannot type the name of an existing prompt directly into the SQL statement.

Once you exit the Prompt Generation Wizard or the Select a Prompt dialog box, the prompt is inserted into the SQL statement at the current cursor position. If an area in the SQL statement is highlighted, it is replaced by the

prompt name. Prompt objects appear in the SQL statement in pink and are enclosed in brackets ([]) if the name of the prompt contains any spaces, for example:

```
WHERE Year_ID = Year_prompt
```

```
WHERE Year_ID = [Year prompt]
```

Element list prompts

If the prompt is an element list prompt and you use the key word **In**, you need to manually add parentheses around the prompt name in the SQL statement. You can select either a single answer or multiple answers to the prompt, yielding results such as (4) or (1,2,3,4). See the example below.

```
Select a11.[YEAR_ID] AS YEAR_ID  
From [LU_YEAR] a11  
Where a11.[YEAR_ID] in (Year_prompt)
```

If you use other operators such as =, >, <, or =/, you do not need to add any parentheses around the prompt name. However, you can only select a single answer to the prompt. Therefore, make sure that the maximum number of answers allowed for the prompt is set to 1. See the example below.

```
Select a11.[YEAR_ID] AS YEAR_ID  
From [LU_YEAR] a11  
Where a11.[YEAR_ID] = Year_prompt
```

Value prompts

Date and number value prompts are properly formatted to the standards of the database platform that the Freeform SQL report is executed against. For example, a date value prompt yields TO-DATE('08-MAY-06') for Oracle and "2006-05-08" for DB2.

However, for text value prompts, you need to manually add single quotes (' ') around the prompt name if you want the text prompt answer to be applied as a text constant. See the example below.

```
Select a11.[YEAR_ID] AS YEAR_ID  
From [LU_YEAR] a11  
Where a11.[YEAR_ID] in ('Text_prompt')
```

You do not need to add the quotes around the prompt name if the answer is part of the SQL command. See the example below.

```
Select Product_ID, Product_DESC, Budget  
From Text_prompt
```



If you are prompting on which table to retrieve data from, you should enter the most commonly used or most appropriate table as the default value of the prompt.

Optional prompts

When you create a new prompt to add to the SQL statement, you can make the answer optional by clearing the Prompt answer required option in the Prompt Generation Wizard. Alternatively, if you use an existing prompt, you need to know if this option was selected during the prompt creation. You can open the prompt with the Prompt Editor to check whether the option has been selected.

For both new and existing prompts, if the prompt answer is optional make sure that the syntax related to the prompt is also made part of the prompt string. Taking this step ensures that if the optional prompt is not answered, then neither the syntax nor the prompt will be processed when the report SQL is generated. If you do not take this step, the report will fail due to an invalid SQL statement.

To accomplish this, refer to the procedure below.

Prerequisites

- The procedure below assumes you have already created a Freeform SQL report and it is open in the Freeform SQL Editor.
- You must have the Define Freeform SQL report and Create schema objects privileges to access and use the Freeform SQL Editor.

To define Freeform SQL statements to support optional prompts

- 1 In the SQL Statement pane (the top pane on the right), highlight the related syntax before and/or after the prompt together with the prompt itself.
- 2 Right-click the highlighted part and select **Prompt-dependent SQL**. The highlighted syntax turns pink just as the prompt.

You should highlight the entire syntax that could be affected by the optional prompt. For example, the first example shown below is correct, whereas the second example could return an error if no answer is supplied for the optional prompt.

```
Select a11.[YEAR_ID] AS YEAR_ID  
FROM [LU_YEAR] a11  
WHERE a11.[YEAR_ID] in (Year_prompt)
```

```
Select a11.[YEAR_ID] AS YEAR_ID  
FROM [LU_YEAR] a11  
WHERE a11.[YEAR_ID] in (Year_prompt)
```

Drilling

Drilling allows you to look at specific data at levels other than what is originally displayed on the grid or graph. For standard reports, you can drill in different directions, such as down, up, or across attributes, based on the drill map. For Freeform SQL reports, support for drilling is limited to attributes within what is known as a personal Intelligent Cube. For a description of what a personal Intelligent Cube is, see the section *Intelligent Cubes*, page 288.

Drilling in Freeform SQL reports works in the following ways:

- This functionality is controlled by the privilege named Drill within Intelligent Cube, and requires an OLAP Services license.
- You can only drill to attributes that are included in the Report Objects pane but are not on the report grid.
- You can only use drill maps with project attributes, since managed object attributes do not have associated drill maps.
- You can drill from managed objects to any other project objects that are included in the Report Objects pane but are not on the report grid. You cannot drill from a managed object to another managed object.



For more information on managed objects, see *Map data to non-project objects: Managed objects*, page 735.

- You can drill down and drill up on project attributes. For example, a Freeform SQL report has the Year and Quarter attributes on the report grid. When you move Quarter off the report to the Report Objects pane, you can drill down from Year to Quarter on the report. Additionally, if Quarter remains on the report and you move Year off the report, you can drill up from Quarter to Year.

- You can drill to attributes that are not in the same hierarchy as the attribute you are drilling from. For example, if Quarter from the Time hierarchy is on the report and Category from the Product hierarchy is only in the Report Objects pane, you can drill from Quarter to Category.



Managed objects do not have an associated hierarchy, so they are able to drill to any other project attribute that resides in the Report Objects pane but is not on the report grid.

Using Freeform SQL reports to filter other reports

Freeform SQL reports can be used to filter other standard reports in MicroStrategy. This can be supported by using a Freeform SQL report as a shortcut-to-a-report qualification on a standard report. For information on shortcut-to-a-report qualifications, see [About the report-as-filter, page 283](#).

However, there are additional configurations and requirements to allow Freeform SQL reports to be used as a shortcut-to-a-report qualification, as described below:

- The Freeform SQL report must map data to project attributes rather than managed object attributes. This is required in order for the standard report to recognize the data that is available on the Freeform SQL report. For information on mapping data in Freeform SQL reports to project attributes, see [Mapping columns to metadata objects, page 618](#).
- If the Freeform SQL report returns data from a data source that is not the primary data warehouse, you must have the Execute Multiple Source Report privilege. This privilege is available as part of MicroStrategy MultiSource. For information on MultiSource, see the *Project Design Guide*.

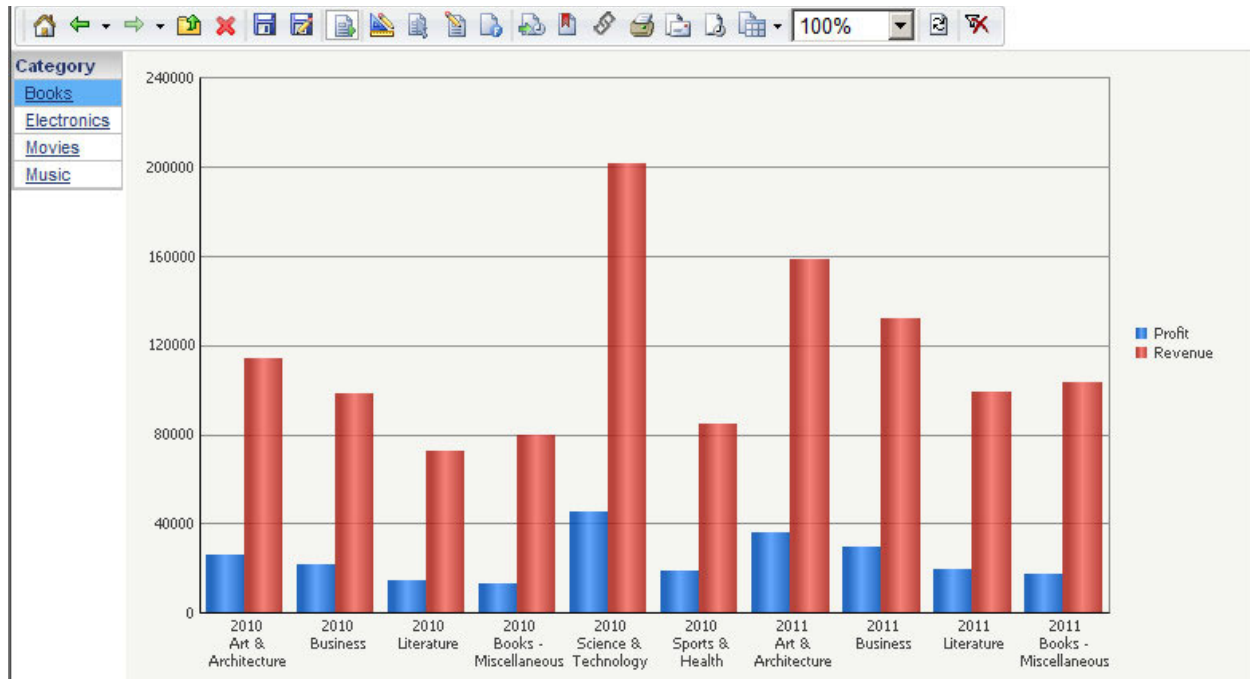
To add a Freeform SQL report as a shortcut-to-a-report qualification in a standard report, open the standard report in MicroStrategy Developer. In the **Report Filter** area, double-click the arrow to add a new qualification. In the **Filtering Options**, select **Add a Shortcut to a Report**. Click **OK**. Click the ... (browse button) to select a Freeform SQL report, and click **OK**. Click **OK** again to create the filter qualification.

Supporting Grid/Graph filtering of document datasets based on Freeform SQL reports

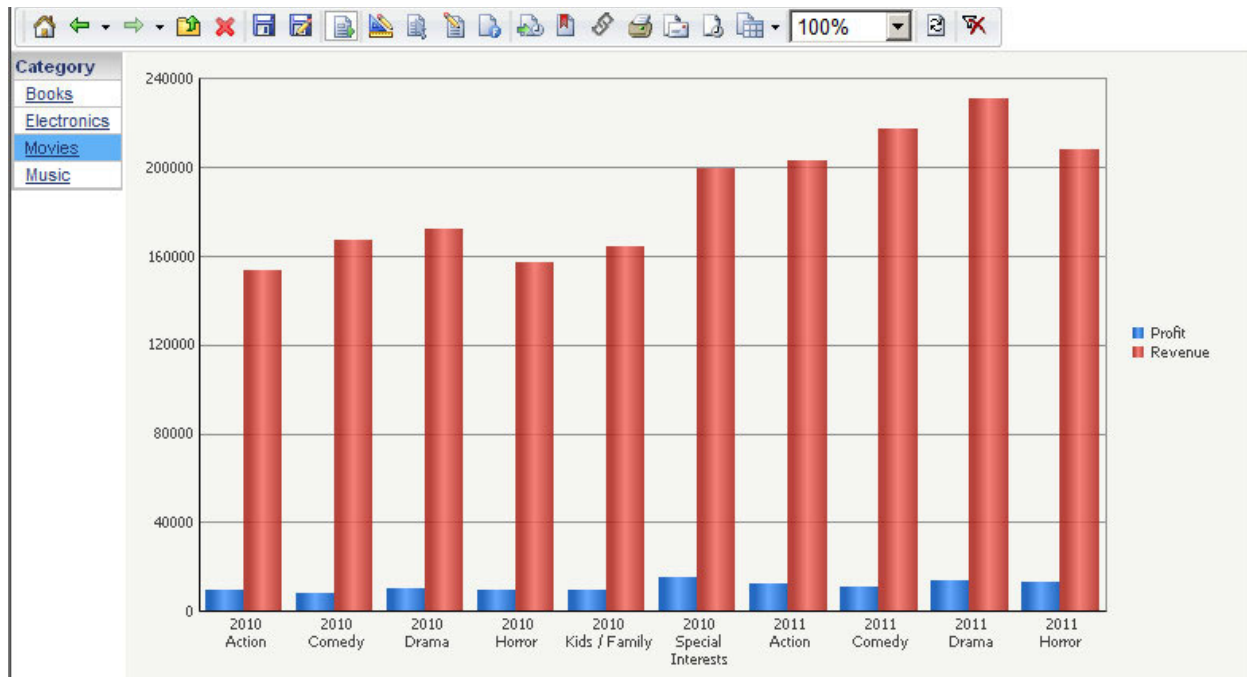
Report Services document designers can enable Grid/Graphs as report condition selectors to filter the data in a dataset, allowing users to view

subsets of large amounts of data, rather than loading and displaying all the data at once. For example, the document below includes a graph that displays the revenue and profit for subcategories of products during various years. This document also includes a Grid/Graph report condition selector that includes the Category attribute. You can click the categories of this Grid/Graph to display results for different categories in the graph.

For example, you can view results for the Books category, as shown below:



If you click the Movies category, the graph displays the results for movies, as shown below:



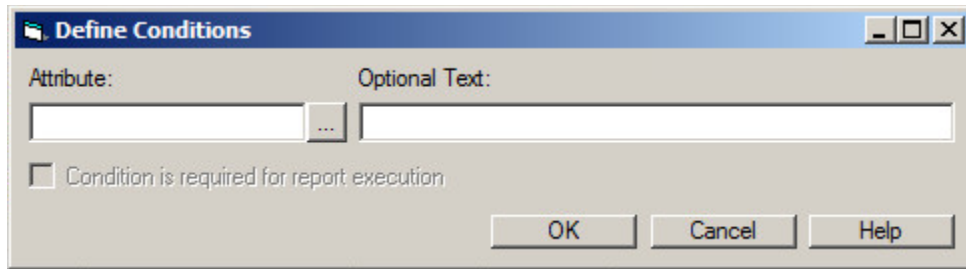
For standard reports in MicroStrategy, this dataset filtering is automatically supported by re-executing the underlying report definition for a dataset. For the example shown above, the graph's SQL statement that is automatically generated by MicroStrategy when clicking a category is as follows:

```
select a13.[YEAR_ID] AS YEAR_ID,
       a12.[SUBCAT_ID] AS SUBCAT_ID,
       max(a14.[SUBCAT_DESC]) AS SUBCAT_DESC0,
       sum((a11.[TOT_DOLLAR_SALES] - a11.[TOT_COST])) AS
WJXBFS1,
       sum(a11.[TOT_DOLLAR_SALES]) AS Revenue
from [ITEM_MNTH_SLS] a11,
     [LU_ITEM] a12,
     [LU_MONTH] a13,
     [LU_SUBCATEG] a14
where a11.[ITEM_ID] = a12.[ITEM_ID] and
      a11.[MONTH_ID] = a13.[MONTH_ID] and
      a12.[SUBCAT_ID] = a14.[SUBCAT_ID]
and a14.[CATEGORY_ID] in (3)
group by a13.[YEAR_ID],
         a12.[SUBCAT_ID]
```

In the SQL statement listed above, the highlighted syntax **and a14.[CATEGORY_ID] in (3)** is dynamically generated when clicking the category selector.

If a dataset uses a Freeform SQL report to retrieve its data, a static Freeform SQL statement cannot automatically be updated to reflect this type of filtering. You must define the Freeform SQL report to apply this filtering to the Freeform SQL definition. You can provide this dynamic filtering support by inserting a condition placeholder into the Freeform SQL statement.

Condition placeholders are created in a Freeform SQL statement by right-clicking within the Freeform SQL statement, and selecting **Insert Conditions**. The Define Conditions dialog box opens, which is shown below.



You can then define the condition placeholder as follows:

- **Attribute:** Click ... (Browse button) to select the attribute used in the report condition selector to filter the display of the dataset in the document. In the example described above, the Category attribute is used to filter the dataset.
- **Optional Text:** Type the SQL syntax that is used for the condition in the Freeform SQL report. The SQL syntax should filter data based on attribute elements, which ensures that the report condition selector can accurately filter the data.

This text is used as part of the SQL statement, so it must support valid SQL syntax along with the rest of the SQL statement. Since the condition placeholder filters the results of the statement, this syntax should be included as part of the `WHERE` clause of the SQL statement.

In the example described above, the optional text can be provided as:

```
and a14.[CATEGORY_ID] in ([Category])
```

This statement filters the results based on the elements of the Category attribute. By including this at the end of the `WHERE` clause, this syntax is added to the SQL statement when a user clicks the report condition selector. Prior to clicking the report condition selector, the optional text is not included in the SQL statement.

- **Condition is required for report execution:** Select this check box if the report for the dataset should only be executed when a selection is made using the separate report condition selector. If you select this check box,

before the user makes a selection, no data is shown for the report. Additionally, this means that the SQL syntax that you include as the Optional Text is only included in the report when a user clicks on the report condition selector.

If you clear this check box, the report is executed before a user makes a selection using the report condition selector. This can provide users a view of the data prior to filtering the data. In this case, the SQL statement must be valid even if the optional text is removed completely from the SQL statement. In the example described for the optional text, the syntax and `a14.[CATEGORY_ID] in ([Category])` is used because this can be removed completely from the SQL statement, and the `WHERE` clause still uses valid SQL.

Once you complete the definition, the condition placeholder is included in the Freeform SQL statement. This condition placeholder is dynamically generated whenever a user makes a selection using a separate report condition selector. For steps to design a document that enables Grid/Graphs as selectors to update dataset results, see the *Dashboards and Widgets Creation Guide*.

Security for data access

MicroStrategy has a robust security model that enables you to create users and groups, determine how they are authenticated, control what data they can see, and what functional privileges they can have. For detailed information on these features, refer to the *System Administration Guide*. This section discusses the access control list (ACL) and security filters that relate to Freeform SQL reports only.

Access control list

An access control list (ACL) is a list of users and groups and the access permission that each one has to objects in a MicroStrategy project. Different users may have different permissions on the same object.

When you use existing objects (including project objects) in Freeform SQL column mapping, the ACL's of these objects are used. However, new attributes and metrics created in Freeform SQL reports inherit the default ACL defined in the Project Configuration Editor. You can modify the default ACL in MicroStrategy Developer by right-clicking a project and selecting Project Configuration. In the Project Configuration window point to Project definition, then Security, and then for Set Freeform SQL and MDX objects

default security select Modify. The Properties[XDA Objects] dialog box is displayed. The Permissions list has the following settings:

User	Children
Administrator	Full Control
Everyone	View
Public/Guest	View



The user who creates the new attributes and metrics with Freeform SQL is automatically given the Full Control permission of the new objects.

In the Properties [XDA Objects] dialog box, you can change the settings of the default ACL for different groups of users.



The changed settings will only affect the new attributes and metrics created subsequently in Freeform SQL reports, but not those objects created prior to the change.

Security filters

In MicroStrategy, a security filter is a filter object that is used to narrow down the result set that users or groups can view when they execute reports or browse elements. Usually assigned by administrators, security filters control what warehouse data users can see within MicroStrategy.

Once a security filter is created for users or groups, it is automatically applied when those users or groups view report data or browse attribute elements. However, you must perform some additional configuration to apply security filters to a Freeform SQL report. You configure your Freeform SQL report to apply security filter qualifications by inserting and configuring a placeholder for a user or group's security filter. Placeholders of this type created in Freeform SQL reports are referred to as "security filter placeholders" in this section.

Freeform SQL works with security filters in the following ways:



If you use a version of MicroStrategy earlier than 8.0, make sure that you run the project update for the metadata; otherwise, security filter functionality with Freeform SQL is not supported.

- By default, Freeform SQL reports ignore security filters. The Report Designer has to insert a security filter placeholder as a qualification in the

WHERE clause of a Freeform SQL report and configure it; otherwise, any user can run the Freeform SQL report without being limited in the data they see. For more information on creating security filter placeholders, see [Create security filter placeholders, page 635](#).

- Security filter qualifications are performed on Freeform SQL reports only if the attribute in the qualification is mapped to a column in the Freeform SQL report using a security filter placeholder. For more information on mapping attributes in security filter placeholders, see [Attribute Mappings, page 636](#).
- Security filter qualifications are ignored in Freeform SQL reports if the attributes in the qualifications are explicitly ignored in a security filter placeholder. For more information on ignoring security filter attributes in security filter placeholders, see [Ignored Attributes, page 636](#).
- Security filters which qualify on multiple attributes can be used with Freeform SQL reports only if every attribute in the security filter is either mapped to a column or ignored in a security filter placeholder. Freeform SQL reports fail for users with security filters containing multiple attribute qualifications that are not mapped or set to be ignored by a security filter placeholder.
- A security filter can restrict the attributes a user can view in relation to the level at which attributes are found within a MicroStrategy hierarchy. For more information on allowing top range and bottom range attributes in security filter placeholders see [Allow security filters with Top and Bottom levels to be evaluated based on the select level of this report, page 637](#). The two attribute range options are as follows:
 - **Top range attribute:** specifies the highest level of detail that the security filter allows the user to view.
 - **Bottom range attribute:** specifies the lowest level of detail that the security filter allows the user to view.
- Because the SQL statement is static, a security filter placeholder string (“WHERE Security Filter” or “and Security Filter”) needs to be manually embedded into the statement, such as the following:

```
Select Year_ID, Store_ID, M1
From Store_Fact
WHERE Security Filter
```

The string WHERE Security Filter would be replaced by WHERE Store_ID = 1 when the report is executed for a user who has a security filter (Store@ID = 1) similar to the following syntax:


```
Select Year_ID, Store_ID, M1  
From Store_Fact  
WHERE Store_ID = 1
```

For more information on embedding security filter placeholder strings, see [Replacement string, page 635](#).

For more information on security filters in general, refer to the *Setting Up User Security* chapter in the *System Administration Guide*.

Create security filter placeholders

Security filter placeholders are created in the Freeform SQL Security Filter Dialog, which can be accessed by selecting the **Insert Security Filter** option from the **Edit** menu in the Freeform SQL Editor.

When you close the editor, the security filter placeholder string is automatically inserted into the SQL statement at the current cursor location. The string is displayed in an uneditable mode, just like a prompt, and is bold and underlined in green, for example:



You can edit the security filter placeholder after it is inserted into the SQL statement by double-clicking the statement.

Within the Freeform SQL Security Filter Dialog that opens, you can create a security filter placeholder using the following options:

- **Replacement string**

The Replacement String field is located at the bottom of the Freeform SQL Security Filter Dialog. The default value for the replacement string is “Security Filter”, which is replaced by the security filter qualification when the report is generated.

To complete the string, add “WHERE” or “and” in front of “Security Filter”. If there is no valid security filter, then the whole string (“WHERE Security Filter” or “and Security Filter”) does not appear in the generated report SQL. For example, when using the replacement string “WHERE Security Filter” as mentioned above, if a user without a security filter runs the same report, the SQL looks like the following:

```
Select Year_ID, Store_ID, M1  
From Store_Fact
```

The whole replacement string is dropped from the generated SQL statement and no security filter is used.

If you write “WHERE” or “and” directly into the SQL statement in the Freeform SQL Editor, instead of in the Replacement String field in the Freeform SQL Security Filter Dialog, the following will happen:

- **For a user with a project security filter:** The report will be generated without any problem.
- **For a user without a project security filter:** The report will fail due to invalid SQL statement.

- **Attribute Mappings**

The Attribute Mapping pane is located on the upper-right side of the Freeform SQL Security Filter Dialog. This is where you map attributes to columns in the SQL statement. For every attribute qualification to be applied in a security filter, you need to provide the form and string for the column mapped to that attribute in the SQL statement. The column mapped in the SQL statement must also be a column (attribute form) that is used in security filters. For example,

Attribute	Form	String
Customer	ID	LU_CUSTOMER.CUSTOMER_ID

- **Ignored Attributes**

The Ignored Attributes pane is located below the Attribute Mapping pane in the Freeform SQL Security Filter Dialog. This is where you specify attributes that may be ignored by the Engine when the report is being generated, even if they appear in a security filter qualification. This option can be useful if your Freeform SQL report does not include attributes that are included in security filters. By ignoring attributes that are not included in the Freeform SQL report, it allows security filters to apply qualifications for mapped attributes without creating any security holes. For example, if you define the following:

- **Attribute Mappings:** Year is mapped to the LU_YEAR.YEAR_ID column
- **Ignored Attributes:** Customer is chosen to be ignored because it is not part of the Freeform SQL report
- **Security filter definition:** Year = 2006 and Customer = Bob

Users with the security filter defined above are able to run the report with the Year = 2006 qualification applied to the Freeform SQL report. There is no security hole because the Customer attribute does not appear on the report.

You can ignore attributes that are included in the Freeform SQL report, but this can cause security holes because all qualifications on these attributes within security filters are ignored.

- **Allow security filters with Top and Bottom levels to be evaluated based on the select level of this report**

This check box option is located in the lower part of the Freeform SQL Security Filter Dialog. This option explicitly defines the **Select Level** of the report (displayed in the line just above this option) as the true level of the data that is retrieved for the report when Top and Bottom criteria are evaluated.



Exercise caution with this option:

- Not selecting this option when the user has Top and Bottom levels defined causes the report to fail.
- Select this option only when you are sure that your query does not retrieve data outside the restriction defined by the security filters. This means that you may need to check the security filters for individual users one by one and see how each one may interact with the query.

A security filter can be applied to a Freeform SQL report only if every attribute in the security filter is either mapped to a column or ignored. The report fails if only a subset of the attributes used in a security filter are either mapped to columns or ignored within the security filter placeholder. For example, a security filter that includes qualifications on Year and Category is applied for a group of users. In the Freeform SQL report, Year is the only attribute that has been mapped to a column in the security filter placeholder. Since Category is not included either as an attribute mapping or an ignored attribute, the report fails for all users with this security filter.

Reporting on third-party web services with XQuery

You can connect to third-party web services to retrieve data and report on that data within MicroStrategy. This allows further integration of information from these data sources into your MicroStrategy projects.

XQuery is a language for processing XML data. Any web service that uses the REST architecture or SOAP protocol can be accessed by using the MicroStrategy Freeform SQL Editor to write XQuery statements. For example, the MicroStrategy report shown below returns city, area code, and zip code information for cities in the state of Virginia.

City	Area Code	Zip Code
Abingdon	540	24210
		24211
		24212
Accomac	757	23301
Achilles	804	23001
Afton	540	22920
Alberta	804	23821
Aldie	703	20105
		22001
		22301
		22302
		22303
		22304
		22305
		22306
		22307
		22308
		22309
		22310
		22311
Alexandria	703	22311

The information displayed in the report shown above is retrieved from the third-party web service <http://www.webserviceX.NET/uzip.asmx>. The high-level steps required to create this report are listed below:

- 1 Identify a third-party web service to retrieve data from, and verify that it uses the REST architecture or SOAP protocol. The third-party web service's documentation should include information on how to use one of these methods to access the web service.

- 2 Create a database instance that lets you connect to and report on web services in a MicroStrategy project: See [Allowing connections to web services in a project, page 639](#) for steps to create a database instance for a web service.
- 3 Develop an XQuery statement to retrieve data from a web service: See [Using XQuery to retrieve data from a web service, page 641](#) for steps to develop an XQuery statement.
- 4 Use the Freeform SQL Editor to integrate the web services data into MicroStrategy for reporting and analysis purposes: See [Creating a report to analyze web service data, page 659](#) for steps to use the Freeform SQL Editor to create a report to analyze web services data.
- 5 If you are creating a large XQuery report, and using the report as a dataset in a document, you can support Grid/Graph filtering on the dataset, which allows users to view subsets of the data, rather than loading and displaying all the data at once. For steps to support Grid/Graph filtering, see [Supporting Grid/Graph filtering for document datasets based on large XQuery reports, page 661](#).

XQuery can also be used to integrate web folder resources into MicroStrategy using the Multimedia widget, as described in [Retrieving web folder resources with XQuery, page 664](#).

Allowing connections to web services in a project

To connect to and report on data from a web service, you can create an XQuery database instance for the web service. The steps below show you how to create a database instance, which allows reports in a MicroStrategy project to connect to web services.

To configure a database instance

- 1 In MicroStrategy Developer, log in to a project source, using a user account with administrative privileges.
- 2 In the **Folder List**, expand the project source, expand **Administration**, expand **Configuration Managers**, and then select **Database Instances**.
- 3 From the **File** menu, point to **New**, and then select **Database Instance**. The Database Instances Editor opens.

- 4 On the **General** tab, in the **Database instance name** field, type a descriptive name for the database instance.
- 5 From the **Database connection type** drop-down list, select **XQuery**.
- 6 In the **Database connection** area, click **New**. The Database Connections dialog box opens.
- 7 In the **Database connection name** field, type a descriptive name for the database connection.
- 8 In the **Default database login name** area, click **New**. The Database Logins dialog box opens.
- 9 In the **Database login** field, type a descriptive name for the database login.
- 10 The login information that you provide depends on the web service that you plan to connect to:
 - If the web service that you plan to connect to requires a user name and password, you must type a valid user name and password in the **Login ID** and **Password** fields, respectively. For information on how you can include this security in your report so that you can access the web services data, see [Implementing security in an XQuery statement, page 644](#).

Alternatively, you can include authentication information as part of the database connection. This allows you to then apply authentication for a web service when a user logs in to a project, rather than including authentication in each XQuery report. For steps on how to include web services authentication information in a database connection for project-wide authentication, see [Using an XQuery source to authenticate users for a MicroStrategy project, page 649](#).
 - If the web service that you plan to connect to does not require a user name and password, you can type any user name in the **Login ID** field, and leave the **Password** field blank. The example web service <http://www.webserviceX.NET/uszip.aspx> does not have any specific user name and password requirements.
- 11 Click **OK**. You are returned to the Database Connections dialog box.
- 12 In the **Default database login name** area, select the new database login that you created, and then click **OK**. You are returned to the Database Instances Editor.

- 13 In the **Database connection** area, select the new database connection that you created, and then click **OK**. The database instance is created.
- 14 To make the database instance available for Freeform SQL reports, right-click a project and select **Project Configuration**. The Project Configuration Editor opens.
- 15 Expand **Database instances**, and select **SQL Data warehouses**.
- 16 From the list of available database instances, select the check box for the database instance that you created in the previous steps. If you are prompted to configure data mart optimization, click **No**.
- 17 Click **OK** to save your changes and close the Project Configuration Editor.

Using XQuery to retrieve data from a web service

The Freeform SQL Editor can be used to connect to third-party web services to retrieve and report on data. However, you cannot use SQL to report on web services. Instead, you must use XQuery statements.

XQuery is a language that is used to retrieve data from documents or data sources that can be viewed as XML. Since you create your own XQuery statements to create reports with the Freeform SQL Editor, an understanding of how to create and use XQuery statements is essential. Some basic techniques of how to create XQuery statements are provided below. You can also use the XQuery Editor and Generator (XEG) to generate an XQuery statement to query a given data source. For steps to use XEG, see [Generating an XQuery statement with XQuery Editor and Generator \(XEG\)](#), page 654.

An XQuery statement, that is used to create a report in MicroStrategy to report on data retrieved from a web service, can be separated into two logical parts. These two logical parts separate the tasks of connecting to the data and returning the data so it can be integrated into MicroStrategy. This separation

of tasks in the structure of the XQuery statement is described below and shown in the following image:

```

declare namespace mstr-xquery = "http://www.microstrategy.com/xquery/rest-functions";
declare function mstr-xquery:post($uri) external;

let $uri := ('http://www.webserviceX.NET/uszip.aspx/GetInfoByState?USState=VA')

let $result := mstr-xquery:post($uri)

return <Table>
<ColumnHeaders>
  <ColumnHeader name="CITY" type="xsd:string"/>
  <ColumnHeader name="STATE" type="xsd:string"/>
  <ColumnHeader name="ZIP" type="xsd:integer"/>
  <ColumnHeader name="AREA_CODE" type="xsd:integer"/>
  <ColumnHeader name="TIME_ZONE" type="xsd:string"/>
</ColumnHeaders>
<Data>
{
for $data in $result//Table
return <Row>{$data/"}</Row>
}
</Data>
</Table>

```

- 1 Connecting to a web service and requesting data: The first part of an XQuery statement includes all the necessary definitions and information to connect to a web service. This includes defining any functions required for the XQuery statement, determining whether to use REST architecture or SOAP protocol, creating a request to retrieve data from the web service, and so on. These definitions are highlighted with a red box with a solid border in the example XQuery statement above.

The definitions required in this part of the XQuery depend on the web service that you access and the data that you plan to retrieve. However, the following parts are required for any XQuery statement:

- Defining the REST architecture or SOAP protocol: In the example XQuery statement shown above, the line `declare namespace mstr-xquery = "http://www.microstrategy.com/xquery/rest-functions";` defines whether the XQuery statement uses the REST architecture or SOAP protocol. MicroStrategy provides the following sets of functions to support querying third-party web services:
 - **MicroStrategy REST functions:** Use the URL `http://www.microstrategy.com/xquery/rest-functions`.
 - **MicroStrategy SOAP functions:** Use the URL `http://www.microstrategy.com/xquery/soap-functions`.

You can also use REST or SOAP functions provided by other third-party vendors such as Zorba. However, MicroStrategy only actively tests and supports the MicroStrategy functions listed above.

- **Defining the web service:** In the example XQuery statement shown above, the line `let $uri := ('http://www.webserviceX.NET/uszip.asmx/GetInfoByState?USState=VA')` defines the URL required to access the web service. Refer to your third-party web service's documentation to determine an accurate URL.
- **Requesting data from the web service:** In the example XQuery statement shown above, the line `let $uri := ('http://www.webserviceX.NET/uszip.asmx/GetInfoByState?USState=VA')` also defines the request for data from the web service. The request for data is included as the `GetInfoByState?USState=VA` part of the URL. For this web service, this statement requests data on the state of Virginia. However, not all web service requests can include the request for data as part of the URL. Refer to your third-party web service's documentation to determine the required syntax for the request of data.
- **Implementing security:** In the example XQuery statement shown above, no login information is required to access the web service. However, a web service can require various types of login information and security protocols to access the data. For information on implementing security in an XQuery statement and how this can change the syntax for the XQuery request, see [Implementing security in an XQuery statement, page 644](#).

- 2 Returning data from a web service so that it can be integrated into MicroStrategy:** The second part of an XQuery statement determines how to return the data retrieved from a web service so that it can be integrated into MicroStrategy. These definitions are highlighted with a blue box with a dashed border in the example XQuery statement above.

You must return the data in a table format, which is why the XQuery statement shown above starts with `return <Table>`. This ensures that the data is in a format that can be understood in MicroStrategy.

The `<ColumnHeader>` section of the statement defines the columns of data that are included in the table. The XQuery statement shown above creates columns for city, state, zip code, area code, and time zone data. These columns become the attributes and metrics of the MicroStrategy report.

The data for these columns is then retrieved in the `<Data>` section of the statement. How you create an XQuery statement to return data depends on the third-party web service's structure, as well as whether you plan to manipulate the data using XQuery prior to integrating it into MicroStrategy.

Implementing security in an XQuery statement

Web services can have various security requirements to access the data that the web service provides. MicroStrategy's support of data retrieval from web services using XQuery statements supports most of the web services security mechanisms available. A web service can require HTTP authentication, Web Services Security, or standard user name and password security requirements, which are all described below:

- HTTP authentication provides authentication using one of several authentication modes. MicroStrategy supports the use of basic and digest HTTP authentication for REST functions, and only basic HTTP authentication for SOAP functions. The user name and password that you provide as part of a database login is automatically used by MicroStrategy to complete any HTTP authentication required by the web service. Creating a database login is part of the steps required to create a database instance to access a web service, which is described in [Allowing connections to web services in a project, page 639](#). A default database login is associated with the web service that you are connecting to. You can also use connection mapping to determine the database login used for each MicroStrategy user. For information on configuring connection mapping, see the *System Administration Guide*. If the web service uses HTTP authentication, you cannot use Web Services Security authentication, which is described below.

The example provided in [Using XQuery to retrieve data from a web service, page 641](#) is an example of using REST functions with the parameter `URI`. In that example, the parameter `URI` provides the URL to access the web service as well as define the data to return. This parameter supports both basic and digest HTTP authentication.

An additional example of using REST functions that can support both basic and digest HTTP authentication is shown below:

```
declare copy-namespaces no-preserve, no-inherit;
declare namespace mstr-xquery = "http://www.microstrategy.com/xquery/rest-functions";

declare function mstr-xquery:post($uri, $payload, $content-type) external;

let $uri := ('http://www.webserviceX.NET/uszip.asmx/GetInfoByState')
let $content-type := ('application/x-www-form-urlencoded')
let $payload := ('USState=VA')

let $result := mstr-xquery:post($uri, $payload, $content-type)
```

This type of REST function requires the following parameters:

- *URI* is the Internet address required to access the web service.
- *Content-Type* is the Internet media type to use for the web form accessed by the request for data.
- *Payload* is the request for data from the web service. For the example above, this requests data on the state of Virginia.
- Web Services Security (WS-Security) provides web services authentication, as defined in the WS-Security standard. MicroStrategy utilizes the `mstr-soap:post()` external function to make SOAP web services calls. This function uses Axis2/C to invoke web services and uses Rampart/C to support WS-Security authentication. Using these XML-based assertions, you can enforce authentication and ensure the confidentiality and integrity of the message. If the web service uses WS-Security authentication, HTTP authentication must be defined as anonymous for the web service.

With WS-Security you can support security features such as:

- Authentication with Username tokens.
- Authentication with X.509 certificates.
- Inclusion of Timestamp tokens.
- SOAP message encryption.
- SOAP message signature.
- Message replay detection.

The MicroStrategy function required to utilize WS-Security has the following syntax and requirements:

```
mstr-soap:post($URI, $SoapInfo, $WSSPolicy,
$SoapCustomHeader, $SoapPayload)
```

Where:

- *URI* is the Internet address required to access the web service.
- *SoapInfo* is the version and action information of the SOAP request required to access the web service. For example, the following is a definition of *SoapInfo* for a web service SOAP request:

```
let $soap-version := element SOAPInfo { element
SOAPVersion{ 'SOAP11' }, element SOAPAction{ ' ' } }
```

The *SoapInfo* element shown above is defined as two elements:

- SOAPVersion, which specifies the SOAP Version of the request. The two possible values for this element are 'SOAP11' or 'SOAP12'.
- SOAPAction, which specifies the required action for the web service.
- *WSSPolicy* is the location of the web service's security policy file, which conforms to the WS-Security and WS-Security policy assertion language along with Rampart/C extensions.

A policy file can be extracted from the Web Services Description Language (WSDL) supplied by a SOAP file. Most web services provide a way to view the WSDL in a web browser. The policy file can also be created based on the security requirements of a SOAP web service. Regardless of how you create the policy file, the file may require manual changes to fully configure the required Rampart/C behavior. This MicroStrategy documentation assumes you have a working knowledge of WS-Security, Axis2/C, and Rampart/C technologies. While a comprehensive explanation of these technologies is not provided, the following links are provided for further details and background information on these subjects:

- The specification for WS-Security can be found at <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>.
- The specification for WS-Security policy language can be found at <http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf>.

- Information on how Rampart/C can be configured using WS-Security policy language can be found at <http://ws.apache.org/rampart/c/docs/configurations.html>.
- *SoapCustomHeader* is the SOAP header required to process the information from the web service.
- *SoapPayload* is the statement that determines what information is retrieved from the web service.
- A web service can require additional authentication requirements to access the data. If a web service requires additional authentication through the use of a user name and password within the body of the request, you can provide a valid user name and password in one of the following ways:
 - Include the user name and password directly in the XQuery statement. While this provides access to the web service, providing a password directly in an XQuery statement can cause a security risk. Additionally, the same security would be used regardless of the user running the report.
 - Use a MicroStrategy database login to provide the user name and password. Creating a database login is part of the steps required to create a database instance to access a web service, which is described in [Allowing connections to web services in a project, page 639](#). The benefits of this approach include:
 - Placeholders are used to supply the database login user name and password, which keeps the user name and password information secure. The required syntax to use the user name and password from a database login in an XQuery statement is
#?DBLOGIN_UID?# for the user name and #?DBLOGIN_PWD?# for the password.
 - You can use different database logins for each MicroStrategy user. This allows the security access to reflect the permissions of the user running the report. A default database login is associated with the web service that you are connecting to. You can also use connection mapping to determine the database login used for each MicroStrategy user. For information on configuring connection mapping, see the *System Administration Guide*.

Using security filters to restrict results from an XQuery source

By integrating web services data into MicroStrategy using XQuery reports, you can also take advantage of standard MicroStrategy security features such

as security filters. You can include a security filter as part of an XQuery report to restrict the web services data that a MicroStrategy user can access.

To use security filters to restrict access to web services data returned on XQuery reports, you must do the following:

- The XQuery statement must use the security filter as part of the definition in the where clause. As an example, part of an XQuery statement is provided below, including the security filter as part of the XQuery statement.

```
let $example := XML definition of the third-party web
service
return
<Table>

  <ColumnHeaders>

    <ColumnHeader name='FIRST_NAME' type='xs:string' /
    >
    <ColumnHeader name='LAST_NAME' type='xs:string' />
    <ColumnHeader name='ADDRESS' type='xs:string' />

  </ColumnHeaders>

  <Data>{for $example1 in $example/Employees/Employee
  where $example1/Security Filter
  return

  <Row>

    <FIRST_NAME>{fn:data($example1/FIRST_NAME)}</
    FIRST_NAME>
    <LAST_NAME>{fn:data($example1/LAST_NAME)}</
    LAST_NAME>
    <ADDRESS>{fn:data($example1/ADDRESS)}</ADDRESS>

  </Row>

  </Data></Table>
```

In the example XQuery statement above, the text **Security Filter** acts as a placeholder for the security filter, which is applied to the Employee data.

- You must create the security filter placeholder within the XQuery report. These must be created using the same techniques used to create a security filter placeholder in Freeform SQL reports, which is described in [Security filters, page 633](#) and [Create security filter placeholders, page 635](#).

Additional details related to creating these security filters in XQuery reports are provided below:

- The attribute mappings for the security filter placeholder should match the attribute form that is mapped to the web services data, and thus can be used to restrict data in the XQuery statement. For the example above, you can include the `FIRST_NAME` attribute form of the Employee attribute as the attribute mapping for the security filter.
- The replacement string for the security filter placeholder must match the syntax for your XQuery statement. This replacement string defines the text in your XQuery statement that is replaced when resolving the security filter. In the example XQuery statement provided above, only the text **Security Filter** is replaced. To support the possibility that a user without a security filter runs the XQuery report, you can define the replacement string as:

```
where $example1/Security Filter
```

By using this replacement string, when a user without a security filter runs the XQuery report, the entire where clause of the XQuery statement is removed, allowing the report to use correct XQuery syntax.

Using an XQuery source to authenticate users for a MicroStrategy project

The authentication options described in [Implementing security in an XQuery statement](#) above can be used directly in an XQuery report.

Additionally, by using database authentication, you can authenticate against an XQuery source when users log in to a project, and save any tokens returned by the authentication process. These tokens can then be used in XQuery reports for the duration of a given user's MicroStrategy session. This authentication technique has the added benefit of keeping your MicroStrategy project secure by authenticating upon login, while allowing the use of tokens to avoid the need to include authentication details in every XQuery report.

The steps below show you how to configure your environment to support authenticating all XQuery reports for a given project.

Prerequisites

- An XQuery statement that returns various tokens from a web service, which can then be used in XQuery reports as authentication tokens, in place of authenticating with every request. A common scenario is to

return two tokens, including a session URL (a URL for subsequent requests) and a session Token (which identifies the caller as an already authenticated user). You can return up to four distinct pieces of information from a web service.

To use an XQuery source to authenticate users for a MicroStrategy project

- 1 In MicroStrategy Developer, log in to a project source, using a user account with administrative privileges.
- 2 In the **Folder List**, expand the project source, expand **Administration**, expand **Configuration Managers**, and then select **Database Instances**.
- 3 From the **File** menu, point to **New**, and then select **Database Instance**. The Database Instances Editor opens.
- 4 On the **General** tab, in the **Database instance name** field, type a descriptive name for the database instance.
- 5 From the **Database connection type** drop-down list, select **XQuery**.
- 6 In the **Database connection** area, click **New**. The Database Connections dialog box opens.
- 7 In the **Database connection name** field, type a descriptive name for the database connection.
- 8 In the **Default database login name** area, click **New**. The Database Logins dialog box opens.
- 9 In the **Database login** field, type a descriptive name for the database login.
- 10 The login information that you provide depends on the web service that you plan to connect to:
 - If the web service that you plan to connect to requires a user name and password, you must type a valid user name and password in the **Login ID** and **Password** fields, respectively. For information on how you can include this security in your report so that you can access the web services data, see *Implementing security in an XQuery statement*, page 644.

- If the web service that you plan to connect to does not require a user name and password, you can type any user name in the **Login ID** field, and leave the **Password** field blank.

11 Click **OK**. You are returned to the Database Connections dialog box.

12 In the **Default database login name** area, select the new database login that you created.

13 On the **Advanced** tab, in the **XQuery authentication parameters** field, type the XQuery statement to return the required authentication credentials from the web service. A complete XQuery statement can be included in this field, but the return statement must have the following structure:

```
<Table>
  <ColumnHeaders>
    <ColumnHeader name="Token1" type="xsd:string" />
    <ColumnHeader name="Token2" type="xsd:string" />
    <ColumnHeader name="Token3" type="xsd:string" />
    <ColumnHeader name="Token4" type="xsd:string" />
  </ColumnHeaders>
  <Data>{
    <Row>
      <Token1>Example1</Token1>
      <Token2>Example2</Token2>
      <Token3>Example3</Token3>
      <Token4>Example4</Token4>
    </Row>
  }</Data>
</Table>
```

The syntax above returns four separate credentials, which can be supplied in the *Example1*, *Example2*, *Example3*, and *Example4* locations respectively. If fewer than four credentials are required, you can delete the associated `<ColumnHeader>` tags and `<TokenN>` tags. For example, the following XQuery statement makes a call to a SOAP web service and returns a session ID and a URI to be used for subsequent connections to a web service:

```
declare namespace mstr-soap =
http://www.microstrategy.com/xquery/soap-functions;
declare function mstr-soap:post($args) external;

let $username := '#?DBLOGIN_UID?#'
let $password := '#?DBLOGIN_PWD?#'

let $uri := 'http://ExampleURI'
```

```

let $soap-version := element SOAPInfo {element
SOAPVersion {'SOAP11'}, element SOAPAction{''}}
let $ws-sec-policy := ''
let $soap-custom-header := ''
let $soap-payload := concat('<login
xmlns="ExampleNamespace">
<username>', $username, '</username>
<password>', $password, '</password></login>')

let $args := ($uri, $soap-version, $ws-sec-policy,
$soap-custom-header, $soap-payload)
let $response := mstr-soap:post($args)//loginResponse

let $sessionToken := $response//sessionId
let $sessionURI := $response//serverUrl
return
<Table>
  <ColumnHeaders>
    <ColumnHeader name="Token1" type="xsd:string" />
    <ColumnHeader name="Token2" type="xsd:string" />
  </ColumnHeaders>
  <Data>{
    <Row>
      <Token1>{$sessionToken}</Token1>
      <Token2>{$sessionURI}</Token2>
    </Row>
  }</Data>
</Table>

```

- 14 Once you have typed a valid XQuery statement, click **OK**. You are returned to the Database Instances Editor.
- 15 In the **Database connection** area, select the new database connection that you created, and then click **OK**. The database instance is created.
- 16 To make the database instance available for Freeform SQL reports, right-click a project and select **Project Configuration**. The Project Configuration Editor opens.
- 17 Expand **Database instances**, and select **SQL Data warehouses**.
- 18 From the list of available database instances, select the check box for the database instance that you created in the previous steps. If you are prompted to configure data mart optimization, click **No**.

- 19 Within the **Database Instances** category, expand **Authentication**, and select **Metadata**.
- 20 From the drop-down list, select the database instance you created earlier in this procedure.
- 21 Click **OK** to save your changes and close the Project Configuration Editor.
- 22 Since you must use database authentication, you must perform some additional configurations:
 - Modify the MicroStrategy user group `Public/Guest` so that it has the necessary privileges to view the projects that use database authentication.
 - Modify any required MicroStrategy user accounts to link them to the web service account. In the User Editor, in the **Authentication: Metadata** category, type the login ID in the **Database Login** field.
 - Create a new project source or modify an existing project source to use database authentication.

For additional details on supporting database authentication in MicroStrategy, see the *System Administration Guide*.

- 23 You must also ensure that each XQuery report references the tokens that store the authentication credentials. The tokens can be accessed by using system prompts in MicroStrategy XQuery reports. These system prompts are stored in the project folder `Public Objects/Prompts/System prompts` and are named `Token 1`, `Token 2`, `Token 3`, and `Token 4`. Each token corresponds to the respective column returned by the XQuery authentication parameters.

Within an XQuery report, you can define variables in an XQuery statement to retrieve the authentication credentials stored in the tokens. For example, the `sessionId` and `uri` variables shown below retrieve the authentication credentials from the `Token 1` and `Token 2` system prompts.

```
let $sessionId := [Token 1]
let $uri := [Token 2]
```

To insert a prompt in an XQuery statement, within the Freeform SQL Editor, from the **Edit** menu, select **Insert Prompt**. You can then select the relevant Token prompt from the list of system prompts. You must include these tokens in any XQuery report that requires this authentication to connect to a web service.

This completes the steps to configure authentication for all XQuery reports for a given project. For additional details on how to create XQuery reports to return data from a web service, see [Creating a report to analyze web service data, page 659](#).

Generating an XQuery statement with XQuery Editor and Generator (XEG)

MicroStrategy Developer includes the XQuery Editor and Generator (XEG) to automatically generate XQuery statements. You can provide XEG with appropriate input parameters, such as the input structure of the web service and any search parameters and data transformations, and XEG generates XQuery statements to query the web service.

Prerequisites

- To use XEG, you must be on a machine that has MicroStrategy Developer installed.
- On Windows 2008 or Windows 7, you must be logged in as an administrator for the machine you are using.

To start XQuery Editor and Generator

On a machine that has MicroStrategy Developer installed, from the **Start** menu, point to **Programs**, then **MicroStrategy Tools**, and select **XQuery Editor and Generator**. XEG opens.



On Windows 2008, Windows 7, or Windows 8, you must start XEG using the **Run As Administrator** option. In the **Start** menu, right-click **XQuery Editor and Generator**, and select **Run as administrator**. XEG opens.

To load an existing XEG scenario

In XEG, from the **File** menu, select **Load**. Browse to the saved scenario file, select it, and click **OK**. The scenario is loaded and each source file is opened in a Tree View window.

Creating a new XQuery scenario with XEG

XEG can generate XQuery statements to query REST and SOAP web services, as well as XML data sources. Refer to the following sections for steps to create a new XQuery statement for each type of data source:

- [To create a new XQuery XML scenario, page 655](#)
- [To create a new XQuery REST scenario, page 655](#)
- [To create a new XQuery SOAP scenario, page 656](#)

To create a new XQuery XML scenario

- 1 In XEG, from the **File** menu, select **New**. The New Simple XQuery dialog box opens.
- 2 From the **Source Type** drop-down list select **XML**. The XML Definition page opens.
- 3 Click ... (the Browse button) next to the **File Name / UNC** field. Browse to the XML file that you want to use as a data source and click **OK**. The XML file is listed in the **File Name / UNC** field.
- 4 To use an XML schema file, click ... (the Browse button) next to the **XML Schema** field. Browse to the XSD file that you want to use as an XML schema and click **OK**. The XSD file is listed in the **XML Schema** field.
- 5 Click **Finish**. The New Simple XQuery dialog box closes. An empty grid is displayed in the main pane, and a Tree View window opens containing the specified XML or XSD file.
- 6 For steps to use XEG to generate an XQuery statement using this data source, see [Generating an XQuery statement using XEG, page 658](#).

To create a new XQuery REST scenario

- 1 In XEG, from the **File** menu, select **New**. The New Simple XQuery dialog box opens.
- 2 From the **Source Type** drop-down list select **REST**. The HTTP URL And Method page opens.

- 3 In the **URL** field, type or paste the URL of the REST web service that you are using as a data source.
- 4 To add additional query parameters to the URL:
 - a From the list, specify whether to use a **GET** or **POST** request. (**PUT** and **DELETE** requests are not supported.)
 - b In the field, type the parameters.
- 5 Click **Next**. The HTTP Authentication page opens.
- 6 From the **Method** drop-down list, select the type of authentication you are using to access the REST web service.
- 7 If you selected an authentication method other than **Anonymous**, in the **Domain**, **User Name**, and **Password** fields, type the login information you are using to access the REST web service.
- 8 Click **Next**. The XML Definition Files page opens.
- 9 Select one:
 - To use an XML file as the definition file, click ... (the Browse button) next to the **Sample XML** field. Browse to the XML file that you want to use as a sample and click **OK**. The XML file is listed in the **Sample XML** field.
 - OR, to use an XML schema file as the definition file, click ... (the Browse button) next to the **XML Schema** field. Browse to the XSD file that you want to use as an XML schema and click **OK**. The XSD file is listed in the **XML Schema** field.
- 10 Click **Finish**. The New Simple XQuery dialog box closes. An empty grid is displayed in the main pane, and a Tree View window opens containing the specified XML or XSD file.
- 11 For steps to use XEG to generate an XQuery statement using this data source, see [Generating an XQuery statement using XEG, page 658](#).

To create a new XQuery SOAP scenario

- 1 In XEG, from the **File** menu, select **New**. The New Simple XQuery dialog box opens.

- 2 From the **Source Type** drop-down list select **SOAP**. The WSDL And Service Options page opens.
- 3 Click ... (the Browse button) next to the **WSDL File** field. Browse to the WSDL file that you want to use as a sample data source and click **OK**. The WSDL file is listed in the **WSDL File** field.
- 4 From the **WSDL Operations** list, select the WSDL operation to perform on the data source.
- 5 Click **Next**. The Request Definition page opens.
- 6 If the **Sample Request** field does not contain a sample SOAP request based on the WSDL operation you selected, click **Generate Request** to generate a sample request. Alternately, click ... (the Browse button) to load a saved sample request from a file.
- 7 Review the sample request and edit it as necessary.
- 8 The fields in the Service Detail area are pre populated based on the specified WSDL file and operation. Review these fields and edit them if necessary.
- 9 Click **Next**. The Request Security page opens.
- 10 From the **Method** drop-down list, select the type of authentication you are using to access the SOAP web service.
- 11 If you selected an authentication method other than **Anonymous**, in the **Domain**, **User Name**, **Password**, and **WS Security** fields, type the login information you are using to access the SOAP web service.
- 12 If you are using data from a Salesforce or Macro SOAP web service, select the appropriate check box.
- 13 Click **Next**. The Response Definition page opens.
- 14 From the **SOAP Version** list, specify which version of the SOAP protocol the specified SOAP web service uses.
- 15 If the **Sample Response Text** field does not contain a sample SOAP response based on the sample request you specified, click **Generate Response** to generate a sample request. Alternately, click ... (the Browse button) to load a saved sample request from a file, or click **Capture Response** to capture a response from the specified SOAP web service.

- 16 To save the text in the sample response field to a file, click **Save**. A Save dialog box opens.
- 17 Browse to the location where you want to save the file, type the file name, and click **OK**. The file is saved.
- 18 Click **Next**. The Other Information page opens.
- 19 To use explicit naming in the SOAP XQuery statement, select the **Use Explicit Naming** check box.
 MicroStrategy does not recommend using explicit naming in XQuery statements that are designed for use in MicroStrategy's Freeform SQL Editor.
- 20 Click **Finish**. The New Simple XQuery dialog box closes. An empty grid is displayed in the main pane. A Tree View window opens for each of the source files.
- 21 For steps to use XEG to generate an XQuery statement using this data source, see [Generating an XQuery statement using XEG, page 658](#).

Generating an XQuery statement using XEG

Once you have loaded the source files into XEG, you can begin creating the structure to generate an XQuery statement.

The Tree View windows contain a tree view of the source files for the XQuery statement. To expand the elements of a source file out to a given level, in the Tree View window, from the **Expand Tree Levels** drop-down list (next to the file name), select that level. For example, to expand all elements of the source file out to the second level, select **2**. To completely collapse the source file, select **0**. To expand or collapse individual elements, click the triangle next to each element's name.

To add an element to the XQuery statement, click and drag the element from the Tree View window to an empty row in the grid. When you add an element to the grid, all its child elements are automatically added as well.

To edit an element, double-click the value you want to edit. Depending on the value, either type the new value in the field, or select the new value from the drop-down list that opens.

To delete an element, select the element and press **Delete**. The row containing the element is deleted.

You can save the structure you have created as an XEG scenario. To save an XEG scenario, from the **File** menu, select **Save**. Specify the file name, browse to the location you want to save the scenario, and click **OK**. The scenario is saved.

To change any of the options you specified in the New Simple XQuery dialog box, from the **File** menu, select **Edit**. The Edit Simple XQuery dialog box opens. This dialog box has the same pages as the New Simple XQuery dialog box. Step through the Edit Simple XQuery dialog box and make any changes necessary. Click **Finish** on the last page of the dialog box. Your changes are applied to the XQuery scenario in progress.

Once you have added and edited all the elements you need to the XQuery grid, you can generate the XQuery statement. From the **File** menu, select **Generate**. The XQuery Generation window opens, with the generated XQuery statement.

To test the generated XQuery statement, from the **File** menu, select **Execute**. The XQuery statement is executed and the results are displayed in the XQuery Execution window.

To use the generated XQuery statement in a Freeform SQL report, copy it to the clipboard and paste it into the Freeform SQL editor. For steps to create a Freeform SQL report, see [Creating a report to analyze web service data](#), page 659.

Creating a report to analyze web service data

Using the Freeform SQL Editor you can create a report that integrates web service data into MicroStrategy for reporting and analysis purposes.

Prerequisites

- Freeform SQL reports can be created in MicroStrategy Developer only. However, these reports can be manipulated and executed from both MicroStrategy Developer and Web. Access to the Freeform SQL Editor is available only to Developers with the Define Freeform SQL Report privilege and those with the Create Schema Objects privilege.

To create a report to analyze web service data

- 1 In MicroStrategy Developer, from the **File** menu, select **New**, and then choose **Report**. The New Grid dialog box opens.
- 2 On the **Freeform Sources** tab, select **Create Freeform XQuery report**.
- 3 In the **Source** area, select the database instance created to access web services, as described in [Allowing connections to web services in a project, page 639](#).
 If no database instance is defined in the metadata, the Freeform SQL Editor cannot be loaded and a message is displayed.
- 4 Click **OK**. The Freeform SQL Editor opens.
- 5 In the SQL Statement pane (the top pane on the right), type or paste your XQuery statement. For information on how to develop an XQuery statement, along with a sample XQuery statement, see [Using XQuery to retrieve data from a web service, page 641](#).
- 6 In the Mapping pane (the bottom pane on the right), map the columns in your XQuery statement to attributes and metrics to be used in the report. Follow the requirements listed below when mapping columns in your XQuery statement to attributes and metrics:
 - When mapping the columns, it is important that you follow the same sequence of the columns as they appear in the `<ColumnHeaders>` section of the XQuery statement. This ensures that the data is mapped to the correct attributes and metrics.
 - Make sure that the number of mappings is the same as the number of columns in the `<ColumnHeaders>` section of the XQuery statement. For example, if your XQuery statement lists 10 columns from which to retrieve data, you should map these columns to exactly 10 attributes and metrics.
 - You must map a column to the ID form for each attribute that you use in your Freeform SQL report. You can also map columns to additional attribute forms for an attribute, but all attribute forms other than the ID form are optional.
 - Freeform SQL reports can map data to metadata objects that are part of the relational project, or metadata objects that are known as managed objects. For information on mapping data in Freeform SQL reports to metadata objects, see [Mapping columns to metadata](#)

[objects, page 618](#). The steps in the procedures linked below describe how to map attributes and metrics for Freeform SQL reports that use SQL statements. However, the same general steps apply for mapping attributes and metrics for Freeform SQL reports that use XQuery statements:

- [Mapping a new attribute form to a column, page 619](#).
- [Mapping a new metric to a column, page 620](#).
- [Mapping an existing object to a column, page 621](#).

For the example report displayed in [Reporting on third-party web services with XQuery, page 638](#), the four attributes created are City, State, Zip Code, and Area Code.

- 7 Click **OK** to close the Freeform SQL Editor. The Report Editor opens.
- 8 Format and define the Freeform SQL report in the same way as you would a standard report, using features such as sorting, view filters, thresholds, exporting, and so on.
- 9 From the **File** menu, click **Save As**. The Save Report As dialog box opens.

You must save the report before you can run it, otherwise a message is displayed.
- 10 Enter a name for the report and click **Save**.
- 11 Run the Freeform SQL report.

Supporting Grid/Graph filtering for document datasets based on large XQuery reports

Report Services document designers can enable Grid/Graphs as report condition selectors to filter the data in a dataset, allowing users to view subsets of large amounts of data, rather than loading and displaying all the data at once. For standard reports in MicroStrategy, this dataset filtering is automatically supported by re-executing the underlying report definition for a dataset.

An example of using dataset filtering to view subsets of data is described in [Supporting Grid/Graph filtering of document datasets based on Freeform SQL reports, page 628](#).

If your dataset is based on a large XQuery report, a static XQuery statement cannot automatically be updated to reflect this type of filtering. You must define the XQuery report to apply this filtering in the XQuery definition, by inserting a condition placeholder into the XQuery statement. The steps to add a condition placeholder are described below.

To add a condition placeholder to an XQuery statement

- 1 In MicroStrategy Developer, navigate to your XQuery report.
- 2 Right-click the report, and select **Edit**. The report opens in the Report Editor.
- 3 From the **Data** menu, select **Freeform SQL Definition**. The Freeform SQL Editor opens, with the XQuery statement for the report.

Typically, the placeholder is added before the `return` statement in XQuery.

For example, an XQuery report retrieves user data using a `for` loop, shown in the code fragment below:

```
<Data>

{

    for $ancestor0 in $ancestor/users/user return

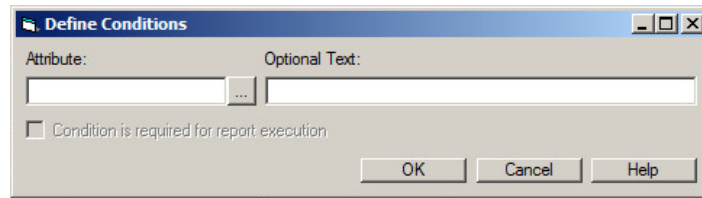
    <Row> <first-name>{fn:data($ancestor0/
    first-name)}</first-name></Row>

}

</Data>
```

In this case, the condition should be inserted before `return`, at the end of the `for` statement.

- 4 In the XQuery statement, right-click the location where you want to insert the condition placeholder, and click **Insert Condition**. The Define Conditions dialog box opens, shown below.



- 5 Next to the Attribute field, click ... (Browse button). The Select Attribute dialog box opens.
- 6 Select the attribute to use in the report's filter, and click **OK**. For example, if you want analysts to be able to filter results by a customer's first name, select First Name.
- 7 In the **Optional Text** field, type the XQuery syntax to filter the report. For example, the syntax to filter results by the First Name attribute is as follows:

```
where fn:data($ancestor0/first-name) = "[First Name]"
```
- 8 If the XQuery report for the dataset should only be executed when a selection is made using the separate report condition selector, enable the **Condition is required for execution** check box.
- 9 Click **OK**. The Define Conditions dialog box closes, and the condition is displayed in the XQuery statement.
- 10 In the Freeform SQL Editor, click **OK**. The XQuery statement is saved, and the Freeform SQL Editor is closed.
- 11 Save and close the XQuery report.

This condition placeholder is dynamically generated whenever a user makes a selection using a separate report condition selector. For steps to design a document that enables Grid/Graphs as selectors to update dataset results, see the *Dashboards and Widgets Creation Guide*.

Retrieving web folder resources with XQuery

MicroStrategy provides web folder content integration, which means that content that is shared using the WebDAV protocol can be integrated into the MicroStrategy environment. This integration allows the content within web folders, such as documents or videos, to be accessed in MicroStrategy Mobile.

The content in your web folders can be integrated into MicroStrategy using XQuery. First you must define an XQuery database instance to access and authenticate the web folder content, as described in the steps [To configure a database instance to retrieve web folder resources](#) below.

Once the database instance is configured, you can do either of the following to retrieve web folder resources and make them available to MicroStrategy Mobile users:

- [Creating an external mobile folder to allow web folder browsing, page 667](#)
- [Creating XQuery reports to retrieve web folder resources, page 668](#)

To configure a database instance to retrieve web folder resources

- 1 In MicroStrategy Developer, log in to a project source, using a user account with administrative privileges.
- 2 In the Folder List, expand the project source, expand **Administration**, expand **Configuration Managers**, and then select **Database Instances**.
- 3 From the **File** menu, point to **New**, and then select **Database Instance**. The Database Instances Editor opens.
- 4 On the **General** tab, in the **Database instance name** field, type a descriptive name for the database instance.
- 5 From the **Database connection type** drop-down list, select **XQuery**.
- 6 In the **Database connection** area, click **New**. The Database Connections dialog box opens.
- 7 In the **Database connection name** field, type a descriptive name for the database connection.

- 8 In the **Default database login name** area, click **New**. The Database Logins dialog box opens.
- 9 In the **Database login** field, type a descriptive name for the database login.
- 10 Type a valid user name and password in the **Login ID** and **Password** fields, respectively. The login information that you provide is used to authenticate a connection to the web folder and its contents if you choose to disable warehouse pass-through authentication for the database instance, or if you are using an authentication scheme other than LDAP.

For steps to enable warehouse passthrough authentication for a project, see [To configure the project to use the WebDAV database instance](#) below.

- 11 Click **OK**. You are returned to the Database Connections dialog box.
- 12 In the **Default database login name** area, select the new database login that you created, and then click **OK**. You are returned to the Database Instances Editor.
- 13 In the **Database connection** area, select the new database connection that you created, and then click **OK**. The database instance is created.
- 14 Right-click the new database instance and select **Properties**. The Properties dialog box opens. In the General category, copy down or otherwise make note of the ID value of the database instance. This ID value is required later if you use an XQuery report to retrieve web folder resources (see [Creating XQuery reports to retrieve web folder resources, page 668](#)).

To configure the project to use the WebDAV database instance

- 15 To make the database instance available for Freeform SQL reports, right-click a project and select **Project Configuration**. The Project Configuration Editor opens.
- 16 Expand **Database instances**, and select **SQL Data warehouses**.
- 17 From the list of available database instances, select the check box for the database instance that you created in the previous steps. If you are prompted to configure data mart optimization, click **No**.
- 18 Under **Database instances**, expand **Authentication**, and select **Warehouse**.

19 Depending on which credentials you want to use to access the WebDAV folder, do one of the following:

- To use the credentials that you defined in the database instance above, ensure the following, as applicable:
 - The **Use warehouse pass-through credentials from User Editor -> Authentication -> 'Warehouse' for warehouse execution** check box is cleared.
 - If the **Use warehouse pass-through credentials from User Editor -> Authentication -> 'Warehouse' for warehouse execution** check box is selected, ensure that the **For selected database instances** option is selected, and the check box for the WebDAV database instance you created above is cleared.
- To use the logged-in user's credentials:
 - Select the **Use warehouse pass-through credentials from User Editor -> Authentication -> 'Warehouse' for warehouse execution** check box, and select **For selected database instances**. Select the check box for the WebDAV database instance you created.
 - Use an LDAP authentication scheme. If you are using any other authentication scheme, the credentials you defined in the database instance are used. For steps to enable an LDAP authentication scheme, see the *System Administration Guide*.
 - Configure Intelligence Server to update pass-through credentials, which is described in [To update pass-through credentials in Intelligence Server](#) below.

For information on choosing whether to use passthrough authentication for database instances, see the *Controlling access to data* chapter in the *System Administration Guide*.

20 Click **OK** to save your changes and close the Project Configuration Editor.

To update pass-through credentials in Intelligence Server

If you selected to authenticate access to web folders and their contents with a user's LDAP credentials, it is recommended that you configure your Intelligence Server's security settings to update the credentials every time a user logs in successfully. If instead the database login is to be used to authenticate users, you can skip these steps and begin to create XQuery reports, as described in [To create XQuery reports to retrieve web folder resources, page 669](#).

- 21 In MicroStrategy Developer, from the **Administration** menu, point to **Server**, and then select **Configure MicroStrategy Intelligence Server**. The MicroStrategy Intelligence Server Configuration dialog box opens.
- 22 Expand the **Server Definition** category, and then select **Security**.
- 23 Within the **Authentication Policy** area, ensure that the **Update pass-through credentials when a successful login occurs** check box is selected, and select the **LDAP Authentication** option. This ensures that users' warehouse pass-through credentials are updated every time they log in.
- 24 Click **OK** to save your changes and close the MicroStrategy Intelligence Server Configuration dialog box. The next step to integrate web folder content into MicroStrategy is to either:
 - Create an external folder to allow for web folder browsing, as described in [Creating an external mobile folder to allow web folder browsing, page 667](#).
 - Create XQuery reports to retrieve the web folder resources, as described in [Creating XQuery reports to retrieve web folder resources, page 668](#).

Creating an external mobile folder to allow web folder browsing

You can create an external mobile folder to allow MicroStrategy Mobile users directly browse folders that are made available over the web. With this web folder integration, MicroStrategy Mobile users can browse the web folder from their MicroStrategy Mobile Home screen, their Shared Reports, or through links on documents.

To create an external mobile folder to allow web folder browsing

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then select **External Mobile Folder**. The External Mobile Folder dialog box opens.
- 2 In the **External mobile folder name** field, type a name to use for the folder within the MicroStrategy project.
- 3 From the **Database Instance** drop-down list, select the database instance created to access web folder resources, as described in [To](#)

configure a database instance to retrieve web folder resources, page 664.

- 4 In the **URL** field, type the URL for the web folder. This URL should provide access to the top-level folder that can be accessed and browsed by MicroStrategy Mobile users.
- 5 Click **OK** to save the new external mobile folder.

Once the external mobile folder is created in MicroStrategy, MicroStrategy Mobile users can browse the web folder from within their Shared Reports. You can also make the web folder accessible to MicroStrategy Mobile users in the following ways:

- **Custom Home screen:** You can provide access to the web folder from a MicroStrategy Mobile user's Home screen. For steps to configure a custom Home screen, refer to the *Mobile Design and Administration Guide*.
- **Links on documents:** You can provide a link to the web folder from within a document. For steps to create links on documents for MicroStrategy Mobile, refer to the *Mobile Design and Administration Guide*.

Creating XQuery reports to retrieve web folder resources

You can create XQuery reports to retrieve each web folder resource. This allows you to make the web folder resources available directly in the Multimedia widget to MicroStrategy Mobile users.

If required, you can define custom icons to display for different file types in your web folder. For example, you can define your own icon to display for PDF files in the folder.

Prerequisites

- Freeform SQL reports can be created in MicroStrategy Developer only. However, these reports can be manipulated and executed from both MicroStrategy Developer and Web. Access to the Freeform SQL Editor is available only to Developers with the Define Freeform SQL Report privilege and those with the Create Schema Objects privilege.
- You need the ID value of the database instance used to access the web folder. This value is retrieved as part of the steps *To configure a database instance to retrieve web folder resources, page 664.*

- To use custom icons for the file types in your WebDAV folder, you must create the images to use for the icons, and upload them to a web server. Use the following guidelines when creating the images:

- The recommended image formats are PNG, GIF, or JPEG.
- The recommended resolution for the images is 50 x 50 pixels.
- To take advantage of high-resolution screens on newer mobile devices, you may also upload higher-resolution versions of each image, with the following prefixes after the file name:

- @1.5x: Recommended for Android devices with higher resolutions. The recommended resolution is 75 x 75 pixels.

For example, if the image you want to use for PDF files is `icon_PDF.png`, and is 50 x 50 pixels, the high-resolution version is `icon_PDF@1.5x.png`, and is 75 x 75 pixels.

- @2x: Recommended for devices for very high resolutions, such as iOS devices with retina displays. The recommended resolution is 100 x 100 pixels.

For example, if the image you want to use for PDF files is `icon_PDF.png`, and is 50 x 50 pixels, the high-resolution version is `icon_PDF@2x.png`, and is 100 x 100 pixels.

- You can include a generic icon for file types that do not have a custom icon defined. For example, if you have defined icons for PDF and XLS, you can include a generic icon for other file types, such as DOC or PPT. If you do not include a generic icon, the default icons for the Multimedia widget are used.

To create XQuery reports to retrieve web folder resources

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then select **Report**. The New Grid dialog box opens.
- 2 On the **Freeform Sources** tab, select **Create Freeform XQuery report**.
- 3 In the **Source** area, select the database instance created to access web folder resources, as described in [To configure a database instance to retrieve web folder resources, page 664](#).



If no database instance is defined in the metadata, the Freeform SQL Editor cannot be loaded and a message is displayed.

- 4 Click **OK**. The Freeform SQL Editor opens.
- 5 In the SQL Statement pane (the top pane on the right), type or paste your XQuery statement. You can use the statement below to build your XQuery statement. If you use this statement, make sure you make the following substitutions in this XQuery statement:

- *WebDAV_Folder*: The URI for the web folder that contains the resources to make available using the Multimedia widget. You can access web folders shared with the WebDAV protocol that are hosted with IIS, Sharepoint, and Apache.

When providing the URI for a web folder, make sure that you end the URI with a / character. For example, a URI such as `http://example.com/MyWebFolder/` uses the proper syntax, but supplying it as `http://example.com/MyWebFolder` without a / character at the end can cause connection problems.

- *Database_Instance_ID*: The ID for the database instance that connects to the WebDAV resource. You must provide this in the XQuery statement as the value for the MW_DBIGUID attribute.
- *http://yourwebserver/icon_[extension]*: If applicable, the URL for custom icons that you can use for the file types in your WebDAV folder. For example, `http://example.com/images/icon_PDF.png`.

```
declare copy-namespaces no-preserve, no-inherit;
declare namespace D= 'DAV:';
declare namespace
Office="urn:schemas-microsoft-com:office:office";
declare namespace mstr-rest =
"http://www.microstrategy.com/xquery/rest-functions";
declare function mstr-rest:webdav($uri, $method, $payload,
$headers) external;
declare function mstr-rest:uri-unescape($uri) external;
declare function local:bool-as-num($value) as xs:string
{
    if (not($value))
    then xs:string(0)
    else xs:string($value)
};
declare function local:gethidden($value) as xs:string
{
```

```
        if (not($value))
        then xs:string(0)
        else xs:string($value)
    };

declare function local:getcollection($value, $resourcetype)
as xs:string
{
    if ($value)
    then xs:string($value)
    else if ($resourcetype/D:collection)
        then xs:string(1)
        else xs:string(0)
};

declare function local:getdisplayname($href) as xs:string
{
    let $name2 := fn:replace($href, ".*\/([^\.]*)\/.*", "$1")
    return mstr-rest:uri-unescape($name2)
};

(: Get the current file extension :)

(: Use this function if you are defining custom icons for
different file types :)

declare function local:getextension($displayname) as
xs:string
{
    let $rev-display := fn:codepoints-to-string(fn:reverse
(fn:string-to-codepoints($displayname)))
    let $rev-ext:= fn:substring-before($rev-display, ".")
    let $ext := fn:codepoints-to-string(fn:reverse
(fn:string-to-codepoints($rev-ext)))
    return $ext
};

(: Use custom icons for different file types :)

(: Use this function if you are defining custom icons for
different file types :)

declare function local:getthumbnailurl($ext) as xs:string
{
    let $extU := fn:upper-case($ext)
    let $file :=
        (: xls files :)
```

```

        if($extU = 'XLS' )
        then "http://yourwebserver/icon_XLS"
        (:pdf files :)
        else if ($extU = 'PDF')
        then "http://yourwebserver/icon_PDF"
        (: default icon :)
        (: To use the default icon configured for the Multimedia
        widget, use a blank string "" :)
        else "http://yourwebserver/icon_default"
        return $file
    };

    (: Set $uri to your WebDAV folder :)
    let $uri := ('WebDAV_Folder')
    (: Encode the uri :)
    let $uri := fn:iri-to-uri($uri)
    (: Set up other parameters for the WebDAV external function
    call :)
    let $method := ('PROPFIND')
    let $payload := ('')
    let $header1 := ('Depth: 0')
    let $header2 := ('Depth: 1')
    (: Make the call to the WebDAV function :)
    let $result1 := mstr-rest:webdav($uri, $method, $payload,
    $header1)
    (: The following call ensures that either you return
    information for one resource or all the resources in all
    folders :)
    let $iscollectiontest := local:getcollection
    ($result1/D:multistatus/D:response/D:propstat/D:prop
    /D:iscollection, $result1/D:multistatus/D:response
    /D:propstat/D:prop/D:resourcetype)
    let $result :=
    if ($iscollectiontest=xs:string(1)) then
    mstr-rest:webdav($uri, $method, $payload, $header2)
    else
        $result1
    (: The uri may need character replacement :)
    let $baseuri := fn:replace($uri, "([h|H].+:/+[^/]*/*).*",
    "$1")
    return
    <Table>

```

```

<ColumnHeaders>
  (:Create attribute form MW_URL with Type=Text:)
  <ColumnHeader name='MW_URL' type='xsd:string' />
  (:Create attribute form MW_Name with Type=Text:)
  <ColumnHeader name='MW_Name' type='xsd:string' />
  (:Create attribute form MW_ModifiedOn with
  Type=DateTime :)
  <ColumnHeader name='MW_ModifiedOn' type='xsd:datetime'
  />
  (:Create attribute form MW_ModifiedOnText with
  Type=Text :)
  <ColumnHeader name='MW_ModifiedOnText'
  type='xsd:string' />
  (:Create attribute form MW_DBIGUID with Type=Text :)
  <ColumnHeader name='MW_DBIGUID' type='xsd:string' />
  (: The MW_ThumbnailURL column is optional, remove the
  comments if you return a custom icon for each resource
  and use Type=Text :)
  (: <ColumnHeader name='MW_ThumbnailURL' type=
  'xsd:string' /> :)
  (: The MW_Description column is optional, remove the
  comments if you return a description for each resource
  and use Type=Text :)
  (:<ColumnHeader name='MW_Description'
  type='xsd:string' /> :)
</ColumnHeaders>
<Data>
{
  let $first_node :=
  $result/D:multistatus/D:response[1]
  for $response in $result/D:multistatus/D:response
  for $prop in $response/D:propstat/D:prop
    let $displayname := if (fn:data($prop/D:displayname) )
    then fn:data($prop/D:displayname)
    else local:getdisplayname(fn:data ($response/
    D:href))
  (: final processing of baseuri to url :)
  let $url := fn:replace(fn:data($response/D:href), "^(/)",
  $baseuri)
  let $compare_name := if (is-same-node($first_node,
  $response)) then "" else
    if($displayname) then fn:upper-case($displayname)
    else fn:upper-case($response/D:href)

```

```

        let $iscollection := local:getcollection
            (fn:data($prop/D:iscollection),$prop/D:resourcetype)
    order by $prop/D:isFolder, $compare_name
    where $iscollection = '0'
    return
<Row>
    <MW_URL>{$url}</MW_URL>
    <MW_Name>{$displayname}</MW_Name>
    <MW_ModifiedOn>{fn:data($prop/D:getlastmodified)}
    </MW_ModifiedOn>
    <MW_ModifiedOnText>{fn:data($prop/D:getlastmodified)}
    </MW_ModifiedOnText>
    (: Provide the ID of the database instance here :)
    <MW_DBIGUID>Database_Instance_ID</MW_DBIGUID>
    (: If you are using custom icons for file types,
    uncomment the following line :)
    (: <MW_ThumbnailURL>
    {local:getthumbnailurl(local:getextension($displayname))}
    </MW_ThumbnailURL> :)
    (: If your resource has description information,
    uncomment the following line and provide the syntax to
    return the description information :)
    (: <MW_Description></MW_Description> :)
</Row>
}
</Data>
</Table>

```

6 In the Mapping pane (the bottom pane on the right), map the columns in your XQuery statement to attributes. Follow the requirements listed below when mapping columns in your XQuery statement to attributes:

- You must map a column to the ID form for each of the following attributes:
 - **MW_URL:** Map the ID form of this attribute to the Text data type.
 - **MW_Name:** Map the ID form of this attribute to the Text data type.
 - **MW_ModifiedOn:** Map the ID form of this attribute to the DateTime data type.

- **MW_ModifiedOnText:** Map the ID form of this attribute to the Text data type.
 - **MW_DBIGUID:** Map the ID form of this attribute to the Text data type.
 - **MW_ThumbnailURL:** If you are not using custom icons for file types, do not create this attribute form. If you are using custom icons, map the ID form of this attribute to the Text data type.
 - **MW_Description:** If you do not return a description of the web folder resource, you should not create this attribute form. If you do create this optional attribute, map the ID form of this attribute to the Text data type.
- When creating the attributes listed above, make sure you list them in the same order they are returned in the `<Row></Row>` section of the XQuery statement.
- 7 Click **OK** to close the Freeform SQL Editor. The Report Editor opens.
 - 8 Run the report to ensure that information is being returned from the web folder resource.
 - 9 From the **File** menu, select **Save As**. The Save Report As dialog box opens.

You must save the report before you can run it, otherwise a message is displayed.
 - 10 Enter a name for the report and click **Save**.

Once the XQuery report to retrieve the web folder resources is created, you must connect this report to a Multimedia widget. For information on configuring a Multimedia widget, see the *Mobile Design and Administration Guide*.

Updating data with Transaction Services

MicroStrategy Transaction Services lets you embed write-back functionality into documents and dashboards for the purposes of decision-making or initiating a transaction. These transactions can include one-click approvals and denials, notes for tracking and directing business activity, and write-back to data sources in real time.

To embed this write-back functionality, you must create a Transaction Services report that defines what data can be updated. Transaction Services reports can be created using the Freeform SQL Editor.

Creating a Transaction Services report

When using the Freeform SQL Editor to create a Transaction Services report, you can use SQL or XQuery to define the data that needs to be updated:

- SQL is used to update the data stored in databases. Creating SQL as part of a Freeform SQL report is described in [Customize your SQL statements: Freeform SQL, page 604](#).
- XQuery is used to report on third-party web services. Creating XQuery as part of a Freeform SQL report is described in [Reporting on third-party web services with XQuery, page 638](#).

In addition to writing the SQL or XQuery statement, you must also define which parts of the statement are used to update values in the data source. The steps below show you how to create a Transaction Services report using the Freeform SQL Editor.

The MicroStrategy Tutorial project includes examples of Transaction Services. These examples include all the reports, documents, and other objects and configurations that are required to use Transaction Services. You can review these Transaction Services example in the following MicroStrategy Tutorial project location: MicroStrategy Tutorial\Public Objects\Reports\MicroStrategy Platform Capabilities\MicroStrategy Transaction Services. High-level steps are provided to explain the creation of the example Transaction Services document Item Order form, see [Transaction Services example: Item Order Form, page 684](#).

Prerequisites

- A license for Transaction Services is required to create a Transaction Services report.
- Access to the Freeform SQL Editor to create a Transaction Services report is available only to Developers with the Define Freeform SQL Report, Create Transaction Report, and Create Schema Objects privileges.

To create a Transaction Services report

- 1 In MicroStrategy Developer, from the **File** menu, select **New**, and then choose **Report**. The New Grid dialog box opens.
- 2 On the **Freeform Sources** tab, select **Create Transaction report**.
- 3 In the **Source** area, select the database instance for the data source to be updated using Transaction Services.
- 4 Click **OK**. The Freeform SQL Editor opens.
- 5 If you are using XQuery to connect to and update data, in the SQL Statement pane (the top pane on the right), type the XQuery statement required to connect to the third-party web service or data source. Creating this part of an XQuery is described as part of the *Connecting to a web service and requesting data* step within [Using XQuery to retrieve data from a web service, page 641](#).

If you are using SQL to connect to and update data, this step is not required.

- 6 Right-click within the SQL Statement pane (the top pane on the right), and select **Insert Transaction**. If you included XQuery syntax as part of the previous steps, you should place the cursor at the end of the XQuery syntax when performing this action.

The Transaction Options dialog box opens.

- 7 To determine whether or not all records of the table that is updated by the transaction are inserted, select or clear the **Insert only distinct records** check box.

By default, this check box is cleared and all records are inserted when a transaction triggers an update. This behavior is intended for using Transaction Services to update fact tables that store fact data such as revenue and profit.

If you select this check box, only distinct records are inserted. This behavior is intended for using Transaction Services to update lookup tables that store attribute information such as customer information.

- 8 Click **OK**. You are returned to the Freeform SQL Editor and `Begin Transaction` `End Transaction` is displayed in the SQL Statement pane.

- 9 Between the `Begin Transaction` and `End Transaction` placeholders, type the statement to update values for your data source.

An input object must be defined for each column of data that is to be updated or is used to distinguish each record. Defining input objects is included as a task later in this procedure. While typing the statement, you can leave the input objects for each column of data blank in the statement.

For example, the SQL statement below updates the columns `TOT_DOLLAR_SALES`, `TOT_UNIT_SALES`, and `TOT_COST` of the `CITY_CTR_SLS` table. Blanks are left after each equals (=) sign.

```
Begin Transaction
Update CITY_CTR_SLS
set TOT_DOLLAR_SALES=, TOT_UNIT_SALES=, TOT_COST=
where CUST_CITY_ID= and CALL_CTR_ID=
End Transaction
```

The `CUST_CITY_ID` and `CALL_CTR_ID` columns are included in the query so that each record in the table can be distinguished when updating the values.

The statement below shows an example of an XQuery statement that updates the records `Amount`, `Date`, and `Comments` of the `Table1` table. Blanks are left for each of the XML tags.

```
Begin Transaction
<Update>
  <table name="Table1">
    <ID> </ID>
    <Amount> </Amount>
    <Date> </Date>
    <Comments> </Comments>
  </table>
</Update>
End Transaction
```

The `ID` record is included in the query so that each record in the table can be distinguished when updating the values.

To define the input objects for a transaction

- 10 You must define the input objects, which provide a link between the data that is updated by a transaction and the object that users interact with to update the data. To define the input objects:
- a Within the statement, place the cursor where the input object needs to be included so that it can define the data for a column. If you are using a SQL statement, it is common to include the input object after an equals sign (=) for a column. If you are using an XQuery statement, it

is common to include the input object within an XML tag for a given record.

- b Once the cursor is placed appropriately, right-click, point to **Define New Input**, and then select one of the following:
 - **Attribute form:** Select this option to map the statement text and its associated data to an attribute. For example, the `CUST_CITY_ID` and `CALL_CTR_ID` columns of the example SQL statement must be mapped to attribute forms. A new attribute form is created in the Input Objects tab.
 - **Metric:** Select this option to map the statement text and its associated data to a metric. For example, the `TOT_DOLLAR_SALES`, `TOT_UNIT_SALES`, and `TOT_COST` columns of the example SQL statement must be mapped to metrics. A new metric is created in the Input Objects tab.
- c You must map the new input object to an attribute form or metric:
 - To create a new attribute or metric for the input object, in the **Object** column, type a name.
 - To select an existing attribute form or metric to map to the input object, right-click the row for the input object and select **Replace**. The Select an Object dialog box opens. Select the object that you want to use and click **OK**.

If the input object is an attribute form, you can map it to the attribute form of a project attribute. This provides a logical connection between the input object and the attribute in your project.

If the input object is a metric, you must map it to a new or existing managed object. Attribute form input objects can also be mapped to managed objects.

Once a name is given to the input object, this name is reflected in the statement. Each input object is enclosed with square brackets (`[]`) within the statement. For example, the SQL statement shown below has input objects defined for each column of data that is to be updated or is used to distinguish each record.

```
Begin Transaction
Update CITY_CTR_SLS
set TOT_DOLLAR_SALES=[Revenue] , TOT_UNIT_SALES=[Units Sold], TOT_COST=[Cost]
where CUST_CITY_ID=[Customer City@ID] and CALL_CTR_ID=[Call Center@ID]
End Transaction
```

- d If the input object is an attribute form, in the **Form** column, select the attribute form for the input object. For each attribute that you define

as an input object, you must map the ID form to an input object. Mapping additional attribute forms to input objects is optional.

- e In the **Required** column, select one of the following:
 - **No:** Defines the input object as optional for a transaction. The input object is only included in a transaction if a user selects to update its value. If you define an input object as optional, make sure that the syntax of your statement would be accurate if the input object is not included in a transaction.

With the new input object selected in the Input Objects tab, select the text of the statement that is dependent on the input object. With the text selected, right-click and select **Input-dependent Text**. For example, using the SQL statement example mentioned previously, select `TOT_DOLLAR_SALES=[Revenue]`, to define the first input. Examples of how to create valid statement syntax with optional input objects are provided in [Defining input-dependent SQL or XQuery syntax, page 681](#).

- **Yes:** Defines the input object as required for a transaction. You must define any input object that is used to distinguish the record updated using a transaction as required. For the example SQL statement provided, the input objects for `CUST_CITY_ID` and `CALL_CTR_ID` must be defined as required for the transaction.

Repeat the steps [To define the input objects for a transaction, page 678](#) to create input objects for each column of data that is to be updated or is used to distinguish each record in your transaction statement.

To create an output object

- 11 On the **Output Objects** tab, right-click and select **Add New Metric**. A new output object is created. This output object does not display any values, but it is required to create a Transaction Services report.

- 12 Type a name for the metric.

To complete and save the Transaction Services report

- 13 Click **OK**. If any dialog boxes are displayed which require you to determine which object to use for an attribute or metric, select the appropriate object. Remember that attributes can be mapped to either project or managed objects, but metrics can only be mapped to managed objects.

The Freeform SQL Editor closes and you are returned to the Report Editor.

- 14 From the toolbar, click **Save and Close**. The Save Report As dialog box opens.
- 15 Type a name for the report and click **Save**. The Transaction Services report is saved.

Once you have created a Transaction Services report, use MicroStrategy Web to create Report Services documents, which can use the Transaction Services report to allow users to update values. This can include interactive documents and dashboards, which can be viewed using MicroStrategy Web as well as MicroStrategy Mobile. For details on how to create a document that can use a Transaction Services report, see the *MicroStrategy Web Help*. An example of the steps required to create the final Report Services document is also provided in [Transaction Services example: Item Order Form, page 684](#).

Defining input-dependent SQL or XQuery syntax

While creating a SQL or XQuery statement for a Transaction Services report, you must define which parts of the statement are dependent on an input. When a transaction is requested, the inputs are included in the SQL or XQuery statement to update the values in the data source.

When creating a Transaction Services report, you can make some of the inputs optional. This supports the scenario that a user only wants to update values for certain metrics or attributes, while leaving the other objects with their current data.

You can define parts of the SQL or XQuery syntax as input dependent as described below:

-  The information below only describes the logic of how to determine which parts of the SQL or XQuery syntax to define as input dependent. For steps to define syntax as input dependent, see [Creating a Transaction Services report, page 676](#).
- If the input is optional, you must define all parts of the syntax that should be removed if a new value is not provided for the input:
 - When using SQL, this can mean including commas and other SQL syntax as dependent on an input. For example, the following SQL statement can be used to update three columns of data:

```
Update CITY_CTR_SLS
set TOT_DOLLAR_SALES=[Revenue],
TOT_UNIT_SALES=[Units Sold], TOT_COST=[Cost]
```

```
where CUST_CITY_ID=[Customer City@ID] and
CALL_CTR_ID=[Call Center@ID]
```

The columns `TOT_DOLLAR_SALES`, `TOT_UNIT_SALES`, and `TOT_COST` can all be defined as optional. However, to ensure that the SQL statement can be valid, at least one of the columns must be required. For this example, the `TOT_COST` column is mapped to a required input. This means the following sections of syntax need to be mapped to inputs:

```
- TOT_DOLLAR_SALES=[Revenue],
```

The entire syntax listed above, including the comma, is mapped to a Revenue metric as an input, which is defined as an optional input.

```
- TOT_UNIT_SALES=[Units Sold],
```

The entire syntax listed above, including the comma, is mapped to a Units Sold metric as an input, which is defined as an optional input.

```
- TOT_COST=[Cost]
```

The entire syntax listed above is mapped to a Cost metric as an input, which is defined as a required input.

Mapping the syntax in this way ensures that the syntax is removed from the statement if the input objects associated with `TOT_DOLLAR_SALES` and `TOT_UNIT_SALES` are not included in the transaction.

- When using XQuery, this can mean including the entire line that requests a certain record as dependent on an input. For example, the following XQuery statement can be used to update three columns of data:

```
<Update>
  <table name="Table1">
    <ID>[ID@ID]</ID>
    <Amount>[Amount]</Amount>
    <Date>[Date@ID]</Date>
    <Comments>[Comments@DESC]</Comments>
  </table>
</Update>
```

The records `Amount`, `Date`, and `Comments` can all be defined as optional. This means the following sections of syntax need to be mapped to inputs:

```
- <Amount>[Amount]</Amount>
```


The entire syntax listed above, including the Amount tags, is mapped to an Amount metric as an input, which is defined as an optional input.

```
- <Date>[Date@ID]</Date>
```

The entire syntax listed above, including the Date tags, is mapped to a Date attribute as an input, which is defined as an optional input.

```
- <Comments>[Comments@DESC]</Comments>
```

The entire syntax listed above, including the Comments tags, is mapped to a Comments attribute as an input, which is defined as an optional input.

Mapping the syntax in this way ensures that the associated syntax is removed from the statement if the input objects associated with Amount, Date, and Comments are not included in the transaction.

- If the input is required, you can define just the input parameter of the syntax that is used to update the required input. This is done automatically when defining a new input object. For example, the following SQL statement can be used to update three columns of data:

```
Update CITY_CTR_SLS
set TOT_DOLLAR_SALES=[Revenue],
TOT_UNIT_SALES=[Units Sold], TOT_COST=[Cost]
where CUST_CITY_ID=[Customer City@ID] and
CALL_CTR_ID=[Call Center@ID]
```

The columns CUST_CITY_ID and CALL_CTR_ID are required to make sure the proper record is updated with the transaction. This means the following sections of syntax need to be mapped to inputs:

▫ [Customer City@ID]

The entire syntax listed above is mapped to a Customer City attribute as an input, which is defined as a required input.

▫ [Call Center@ID]

The entire syntax listed above is mapped to a Call Center attribute as an input, which is defined as a required input.

Since the inputs are required, a value is always provided for CUST_CITY_ID= and CALL_CTR_ID=, which ensures that the syntax is valid.

For an example of a required input for an XQuery statement, the following XQuery statement can be used to update three columns of data:

```
<Update>
  <table name="Table1">
    <ID>[ID@ID]</ID>
    <Amount>[Amount]</Amount>
    <Date>[Date@ID]</Date>
    <Comments>[Comments@DESC]</Comments>
  </table>
</Update>
```

The record ID is required to make sure the proper record is updated with the transaction. This means this record must be mapped to required input objects, using the following syntax:

```
[ID@ID]
```

The entire syntax listed above is mapped to an ID attribute as an input, which is defined as a required input.

Transaction Services example: Item Order Form

The MicroStrategy Tutorial project includes examples of Transaction Services in the location: MicroStrategy Tutorial\Public Objects\Reports\MicroStrategy Platform Capabilities\MicroStrategy Transaction Services.

Included in these examples is the Item Order Form document. This document is an example of using Transaction Services to review the current inventory of items and place an order for items to replenish that inventory as needed. As part of the transaction, you can request a new order, including the quantity of the item and whether the order is high priority. You can also use the same document to update your orders before they are fully processed,

for example, providing a new delivery date or changing a priority flag. The Item Order Form is shown below.

ITEM ORDER FORM Subcategory: **Video Equipment**

INSERT Fill new item order.
UPDATE Order is filled. Update values.
INCORRECT Incorrect values, please update.

For each Item, please insert delivery date, priority flag and units requested.

Order Status	Item	Revenue - Profit Dec 2011	Price	Inventory	Delivery Date	Priority Flag	Units Requested
INSERT	166 ErgoRev Elite Wireless Keyboard	\$29,200 14.33%	\$	31		☒	
UPDATE	167 Panasonic 4-Head Hi-Fi VCR	\$3,150 12.89%	\$ 140	24	05/16/2012	☒ YES	10
INSERT	168 Panasonic Home Theater Stereo VCR	\$5,634 12.42%	\$ 180	34		☒	
INSERT	169 Panasonic Multiformat DVD Player	\$11,100 12.05%	\$ 300	40		☒	
INSERT	170 Panasonic Hi-Fi Stereo VCR	\$6,920 16.53%	\$ 200	37		☒	
INSERT	171 Panasonic 27" TV/DVD Player Combo	\$16,740 14.89%	\$ 600	30	5/24/2012	☒	5
INSERT	172 Sony DVD/CD Player	\$16,300 18.50%	\$ 400	43		☒	

4/27/2012 12:40:17 PM

The steps below provide a high-level overview of how this example document was created. The steps focus on the configurations most closely associated with configuring Transaction Services. Detailed steps to create the supporting objects such as metrics, attributes, and so on are not provided, but you can review their definitions in the Tutorial project.

To create a Transaction Services item order form

- 1 In MicroStrategy Developer, log in to the MicroStrategy Tutorial project.

To create the Transaction Services reports

The Item Order Form document uses two different Transaction Services reports, one to insert new orders (Insert Item Order) and a second to update any existing order (Update Item Order).

- 2 From the **File** menu, point to **New**, and select **Report**. The New Grid dialog box opens.
- 3 From the **Freeform Sources** tab, select **Create Transaction report**. Then select the **Operational Datamart** as the source, as this example creates the data in this data source.

- 4 Click **OK**. The Freeform SQL Editor opens.
- 5 Right-click within the SQL Statement pane (the top pane on the right), and select **Insert Transaction**. The Transaction options dialog box opens.
- 6 Click **OK** to return to the Freeform SQL Editor. The `Begin Transaction End Transaction` text is displayed in the SQL Statement pane.
- 7 Between the `Begin Transaction` and `End Transaction` placeholders, type the following statement:

```
INSERT INTO F_INVENTORY_ORDERS  
VALUES  
  
( , , , , , , , , 1, )
```

This statement will insert a new record into the `F_INVENTORY_ORDERS` table, which has nine columns of data. The value `1` indicates that the data is a new order that has been inserted; the other eight columns must be connected to attributes and metrics to provide their values.

- 8 Define the input objects for the transaction report. You accomplish this by putting the cursor just before each comma in the statement provided above, then right-click and point to **Define New Input**. The first six you must define as new attribute forms, the last three are defined as metrics. For each attribute form or metric you can create a new attribute form or

metric as required. This creates the syntax for the transaction statement as well as creates the input object, as shown below:

Begin Transaction

```
INSERT INTO F_INVENTORY_ORDERS
VALUES
([Subcategory_I@ID], [Subcategory_I@DESC], [Month_I@ID], [Month_I@DESC],
[Item_I@ID], [Item_I@DESC], [Priority_Flag], [Delivery_Date], 1, [Units_Requested])
```

End Transaction

Output Objects | Input Objects

Column	Object	Form	Type	Required
0	Subcategory_I	ID	Number	Yes
1	Subcategory_I	DESC	Text	Yes
2	Month_I	ID	Number	Yes
3	Month_I	DESC	Text	Yes
4	Item_I	ID	Number	Yes
5	Item_I	DESC	Text	Yes
6	Priority_Flag		Text	Yes
7	Delivery_Date		Date	Yes
8	Units_Requested		Number	Yes

All of these input objects are defined as required.

- 9 From the **Output Objects** tab, map the Output attribute's ID form as the one output object.
- 10 Click **OK**. The Report Editor opens.
- 11 From the toolbar, click **Save and Close**. Name the Transaction Services report as Insert Item Order. This completes the creation of the Transaction Services report to insert new orders.
- 12 From the **File** menu, point to **New**, and select **Report**. The New Grid dialog box opens.
- 13 From the **Freeform Sources** tab, select **Create Transaction report**. Then select the **Operational Datamart** as the source, as this example creates the data in this data source.
- 14 Click **OK**. The Freeform SQL Editor opens.
- 15 Right-click within the SQL Statement pane (the top pane on the right), and select **Insert Transaction**. The Transaction Options dialog box opens.

16 Click **OK** to return to the Freeform SQL Editor. The `Begin Transaction End Transaction` text is displayed in the **SQL Statement** pane.

17 Between the `Begin Transaction` and `End Transaction` placeholders, type the following statement:

```
UPDATE F_INVENTORY_ORDERS

SET

DELIVERY_DATE = ,

PRIORITY_FLAG = ,

UNITS_REQUESTED =

WHERE

MONTH_KEY =

AND ITEM_KEY =
```

This statement will update records in the `F_INVENTORY_ORDERS` table.

18 Define the input objects for the transaction report. You accomplish this by putting the cursor just after each equal sign (=) in the statement provided above, then right-click and point to **Define New Input**. The `MONTH_KEY` and `ITEM_KEY` are defined as attribute forms, and the `DELIVERY_DATE`, `PRIORITY_FLAG`, and `UNITS_REQUESTED` are

defined as metrics. This creates the syntax for the transaction statement as well as creates the input objects, as shown below:

Begin Transaction

```

UPDATE F_INVENTORY_ORDERS
SET
DELIVERY_DATE = [Priority_Flag],
PRIORITY_FLAG =[Delivery_Date] ,
UNITS_REQUESTED = [Units_Requested]
WHERE
MONTH_KEY = [Month_I@ID]
AND ITEM_KEY = [Item_I@ID]

```

End Transaction

Output Objects Input Objects

Column	Object	Form	Type	Required
0	Month_I	ID	Number	Yes
1	Item_I	ID	Number	Yes
2	Priority_Flag		Text	Yes
3	Delivery_Date		Date	Yes
4	Units_Requested		Number	Yes

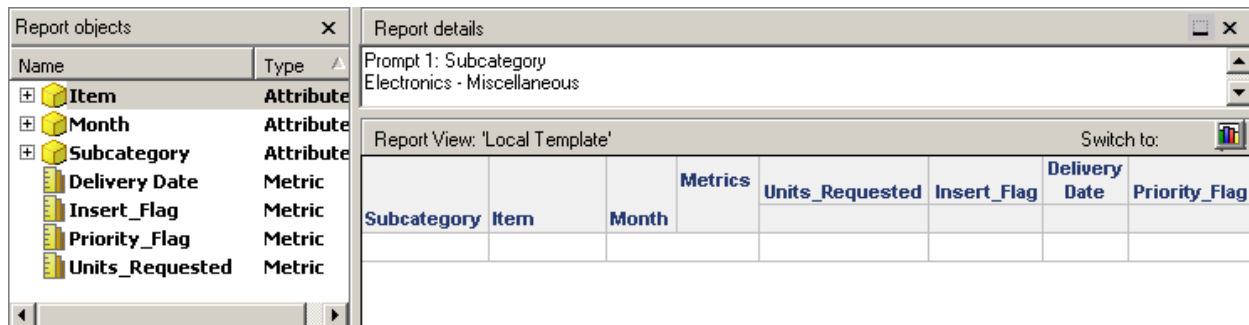
Notice that all of these input objects are defined as required.

- 19 From the **Output Objects** tab, map the Output attribute's ID form as the one output object.
- 20 Click **OK**. The Report Editor opens.
- 21 From the toolbar, click **Save and Close**. Name the Transaction Services report as Update Item Order. This completes the creation of the Transaction Services report to update orders.

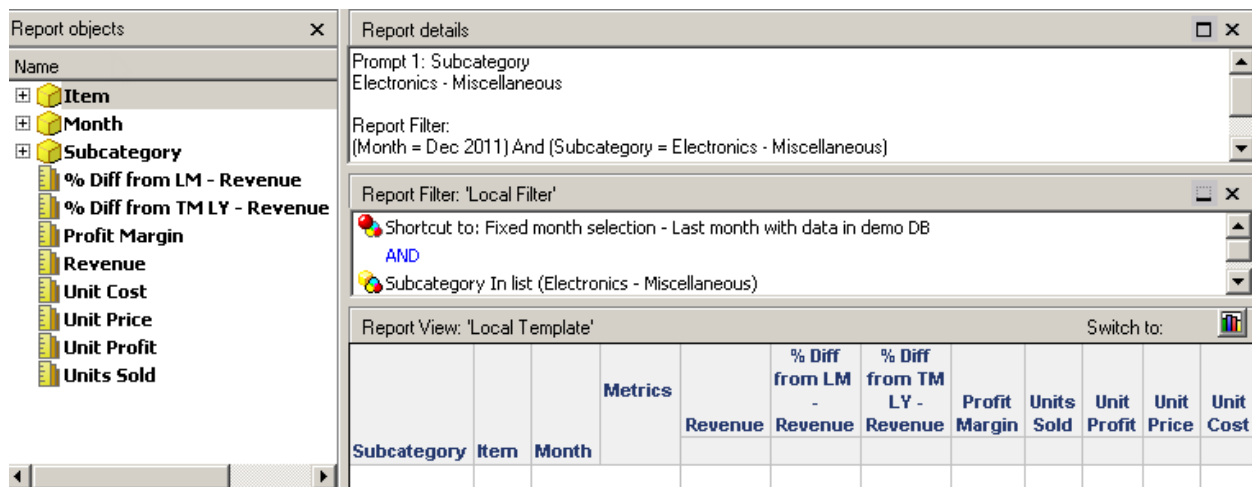
To create the reports to display data on your document

The Item Order Form document uses two reports as datasets to display the data. These reports are provided in the Tutorial project as Report Item Orders and Revenue and Units Sold by Subcategory.

- 22** Create the Report Item Orders report, which serves to include the objects from the Transaction Services reports on the document. The general definition of this report is shown below:



- 23** Create the Revenue and Units Sold by Subcategory report, which serves to include additional data on the document. The general definition of this report is shown below:



To create the Item Order Form document

- 24** In MicroStrategy Web, log in to the Tutorial project.
- 25** Click the **MicroStrategy** icon, and then select **Create Document**. A list of dashboard-style document templates and document templates is displayed.
- 26** From the list of document templates, click **Blank Document**. The Blank Document template is displayed.
- 27** Click within the Dataset Objects area to add datasets for the document. Add the two reports Report Item Orders and Revenue and Units Sold by Subcategory as datasets of the document.

- 28** Create and format the document. The general structure of the document is shown below.

You can also access the Item Order Form document in the Tutorial project to review each component of the document, and refer to the *Report Services Document Creation Guide* for steps to create a document. For the purposes of this example, only the parts related to support Transaction Services are explained below.

- 29** This document uses panel stacks to support each transaction. The Insert, Update, and Incorrect sections shown below are each a separate panel stack for the document.

You must create these separate panel stacks and then include the objects as shown below on each panel stack. For best practices on creating panel stacks, see the *Dashboards and Widgets Creation Guide*.

30 Once you have created each panel stack, you must define the transaction that is performed for each panel stack, as described below:

- **Insert panel stack:** This panel stack supports the ability to insert new item orders. Right-click within the panel stack and select **Configure Transaction**. The Configure Transaction dialog box opens. Since this panel stack is inserting new orders, in the **Transaction Report** field, browse to and select the Insert Item Order report you created earlier in these steps. This displays all the input objects available for the Insert Item Order report. You must then define which of these input objects performs the transaction. These definitions are shown below:

Transaction Input	Field	Editable	Control Style
Month_I@ID	{Month@ID}	☐	
Month_I@DESC	{Month@DESC}	☐	
Subcategory_I@ID	{Subcategory@ID}	☐	
Subcategory_I@DESC	{Subcategory@DESC}	☐	
Item_I@ID	{Item@ID}	☐	
Item_I@DESC	{Item@DESC}	☐	
Delivery_Date	{[Delivery Date]}	☒	Calendar
Priority_Flag	{Priority_Flag}	☒	Switch
Units_Requested	{Units_Requested}	☒	Slider

Advanced

☐ Flag fields with modified data

OK Cancel

- **Update panel stack:** This panel stack supports the ability to update item orders. Right-click within the panel stack and select **Configure Transaction**. The Configure Transaction dialog box opens. Since this panel stack is updating existing orders, in the **Transaction Report** field, browse to and select the Update Item Order report you created earlier in these steps. This displays all the input objects available for the Update Item Order report. You must then define which of these

input objects performs the transaction. These definitions are shown below:

Transaction Input	Field	Editable	Control Style
Month_I@ID	{Month@ID}	☐	
Item_I@ID	{Item@ID}	☐	
Delivery_Date	{[Delivery Date]}	☒	Calendar
Priority_Flag	{Priority_Flag}	☒	Switch
Units_Requested	{Units_Requested}	☒	Slider

Advanced
☐ Flag fields with modified data

- **Incorrect panel stack:** This panel stack supports the ability to update item orders that have been marked as incorrect. Right-click within the panel stack and select **Configure Transaction**. The Configure Transaction dialog box opens. Since this panel stack is updating existing orders, in the **Transaction Report** field, browse to and select the Update Item Order report you created earlier in these steps. This displays all the input objects available for the Update Item Order report. You must then define which of these input objects performs the transaction. You can use the same definitions that were used for the update panel stack described above.

31 With all the transactions defined, you must include a selector on the document that lets users submit their changes. In the example, this is handled with the Submit button highlighted below:

ITEM ORDER FORM Subcategory: {Subcategory}

For each Item, please insert delivery date, priority flag and units requested.

Submit

INSERT Fill new item order.
UPDATE Order is filled. Update values.
INCORRECT Incorrect values, please update.

Order Status	Item	Revenue - Profit {Month}	Price	Inventory	Delivery Date	Priority Flag	Units Requested

To create this selector, from the **Insert** menu, point to **Selector**, and then select **Action Selector Button**. Within the Subcategory Header section, click and drag the area for the new button. The new selector is created.

32 Right-click the new selector and select **Properties and Formatting**. The Properties and Formatting dialog box opens.

- 33** Underneath Properties, click the **Selector** category. This is where you define how the transaction is submitted, as described below:
- a In the **Action Type** drop-down list, select **Submit**.
 - b Select the **Display message after submitting** check box, and then type a message that is displayed as a confirmation to the person who submits a transaction.
 - c Select the **Invalidate mobile device cache** check box. This ensures that if the document is viewed using MicroStrategy Mobile that the updated data can be retrieved.
 - d In the **Targets** area, move the three panel stacks you created earlier to the Selected pane.

All other defaults can be kept.

- 34** Click **Apply** and then click **OK**. You can then save the document as Item Order Form. This completes the steps to create the document.

With this document created, analysts can now view this document in MicroStrategy Web using Express mode. While viewing the document, they can submit item orders.

Customize your SQL queries: Query Builder

Query Builder provides an easy way to quickly access your ODBC data sources without having to write any SQL. You can create queries to be run against imported database tables, which allows you to begin reporting and analyzing with MicroStrategy without performing the project creation step of modeling attribute and fact schemas. This step is necessary for the ROLAP Engine to define attribute and fact schemas. You can also import tables into a project's Warehouse Catalog using the Query Builder feature.

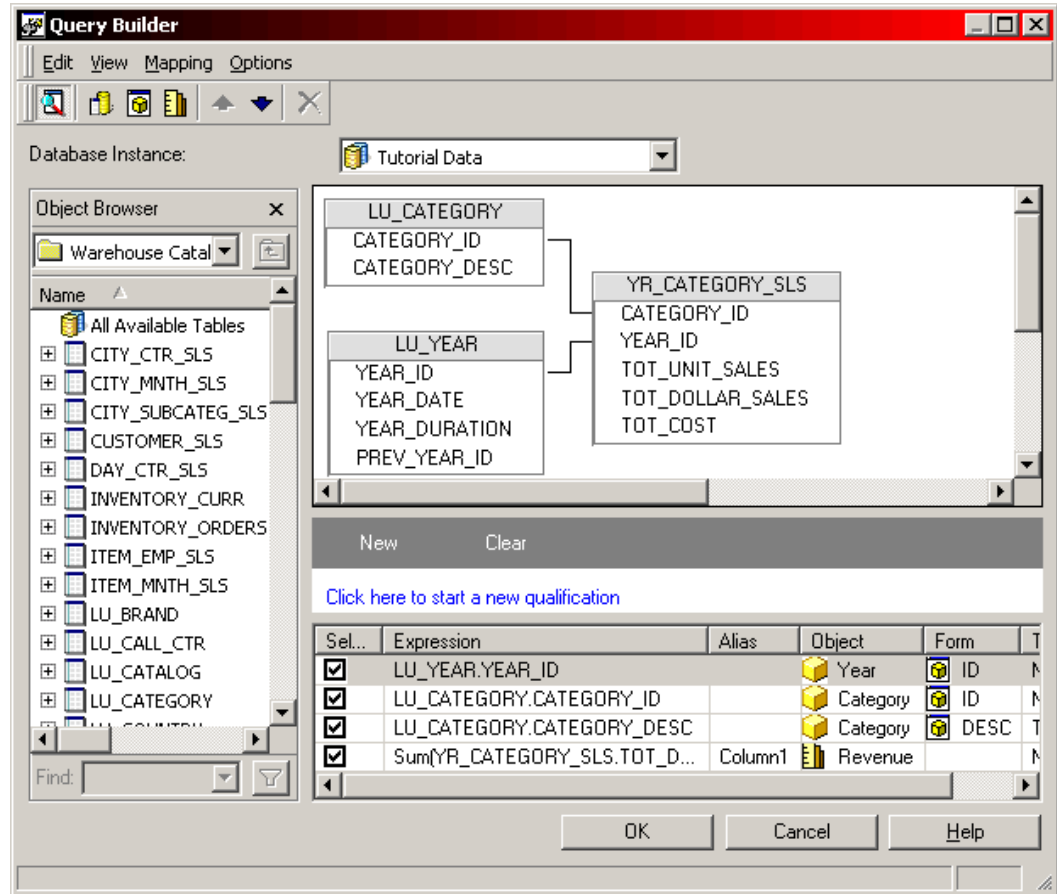
Query Builder allows you more control over the SQL generated against your database systems, without the need for extensive knowledge on how to create SQL statements. A basic knowledge of how SQL statements use tables, columns, and joins to build queries is essential. You should also be familiar with the names of the tables and columns in your data warehouse or ODBC data sources, as well as what information they store.

In addition to simplifying the process of data access, Query Builder provides the majority of MicroStrategy's wealth of reporting and analysis capabilities.

For more information on the MicroStrategy features that apply to reports created with Query Builder, see [Query Builder reports vs. standard reports](#), page 697.

 Reports that are built using the Query Builder feature are referred to as Query Builder reports in this chapter.

The following image shows the Query Builder Editor, where you choose the tables, columns, and joins to define the database queries for a report.



The individual panes of the Query Builder Editor shown above provide the following features:

- **Object Browser pane:** This pane is located on the left side of the Query Builder Editor. Here you can browse the Warehouse Catalog or your data warehouse directly for tables to include in your queries. You can also browse the project folders for objects such as attributes that can also be used to build your queries.
- **Tables and Joins pane:** This is the top pane on the right side of the Query Builder Editor. Tables and joins can be added to the definition of your queries here, which hold the columns that provide the data from

which your report results are drawn. For more information on how Query Builder uses tables and joins to build queries, see [Access and analyze multiple data sources with Query Builder, page 699](#).

- **Conditions pane:** This is the middle pane on the right side of the Query Builder Editor. Here you can define qualifications and insert security filters that will be included in the `WHERE` or `HAVING` clause of your SQL queries, which filter the data that appears on the resulting report. For more information on defining qualifications for Query Builder reports, see [Filters, page 717](#).
- **Selections pane:** This is the bottom pane on the right side of the Query Builder Editor. Columns added to this pane are included in the `SELECT` clause of your SQL queries and become the attributes and metrics on the resulting report. For more information on selecting columns in Query Builder, see [Select columns of data to return, page 712](#).

Usage scenarios

Query Builder allows more control over the queries against your database tables than normal MicroStrategy reports allow, without having to write any SQL statements by hand with the Freeform SQL feature. It allows you to create queries by dragging database tables from the warehouse catalog or directly from the data warehouse into the Query Builder Editor, defining the joins between tables, and selecting the columns of data to be returned. These queries create reports that can be displayed in MicroStrategy Developer or Web. This functionality can be useful in the following scenarios.

When should I use the Query Builder feature?

You can use Query Builder to run queries against a set of database tables that are not easily modeled to an attribute and fact schema. This includes databases that are a collection of flat tables rather than being defined into fact and lookup tables, or other operational data stores. It can also save time when setting up your project since Query Builder only requires that database tables are imported to a project's Warehouse Catalog. Query Builder does not require you to create an attribute and fact schema for your project. You can even import tables into a project's Warehouse Catalog within the Query Builder Editor. For more information on importing tables with the Query Builder Editor, see [Select and add tables, page 705](#).

Query Builder allows you to pick the tables, columns, and joins used in your queries. This can be helpful if you are more accustomed to creating queries at

this level. Query Builder allows you more control over the SQL generated against your database without the need for extensive knowledge on how to create SQL statements.

Query Builder also provides a robust system when moving from one data warehouse to another. Unlike Freeform SQL, which uses static SQL statements that you provide, Query Builder creates the SQL statements based on tables and columns you select and map to MicroStrategy objects. Because of this functionality, Query Builder SQL statements can automatically change to reflect the syntax of a data warehouse.

For example, if you move from a development Oracle data warehouse to a production SQL Server data warehouse, the SQL statements for your Query Builder reports automatically change to reflect SQL Server syntax. For Query Builder reports, this seamless move from a development to a production data warehouse requires that both data warehouses have the same exact table structures, relationships, and column naming conventions.

Query Builder reports vs. standard reports

A Query Builder report is simply a standard report, which has been built using the Query Builder feature. Due to some variations in how standard reports and Query Builder reports are built, there are also some varying functionalities between the two.

You can create Query Builder reports by building your own queries against imported database tables. Although Query Builder reports can only be created on MicroStrategy Developer, once created, they can be executed from both MicroStrategy Developer and Web like any other MicroStrategy standard reports. Functions that you can perform on MicroStrategy standard reports can also be performed on Query Builder reports, including the following:

- Formatting
- Exporting
- Thresholds
- Filtering
- Value prompts
- Graphing

- Shortcut-to-a-report qualification. For more information on using a Query Builder report as a shortcut-to-a-report qualification, see [Using Query Builder reports to filter other reports, page 726](#).
- Narrowcast Server subscriptions and report execution
- Object security
- OLAP services
- Prioritization
- Report Services documents
- Drilling within a personal Intelligent Cube
- Scheduling
- Subtotals

The following features are available for use with Query Builder reports if an attribute and fact schema is used to model the data returned from your queries:

- Element list prompts
- Security filters

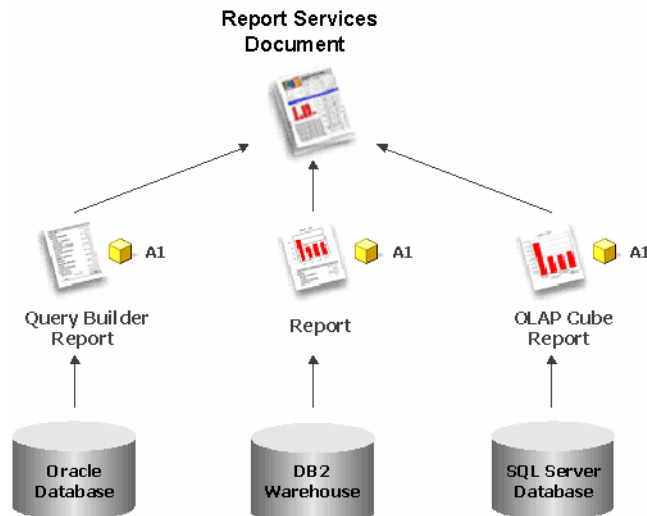
The following features are not available for use with Query Builder reports:

- Custom groups
- Consolidations
- Transformations
- Existing filters
- Save as template/filter
- Data marting

Query Builder reports in Report Services documents

Once created, Query Builder reports can be included in Report Services documents in the same way as standard reports. The same document can also contain reports from other data sources, such as MDX cube reports. For information regarding MDX cube reports, refer to the *MDX Cube Reporting Guide*.

For data to be joined across different data sources in a document, a common attribute is needed across the datasets. In the following diagram, the common attribute which appears on each of the three reports is named A1.



You can establish a common attribute by mapping objects, such as attributes and prompts, retrieved from different data sources to existing objects in the MicroStrategy environment. For more information on mapping columns for Query Builder reports, see [Select columns of data to return, page 712](#).

For example, in a Report Services document, you have three datasets from three reports: one standard MicroStrategy report, one Query Builder report, and one MDX cube report using SAP BI as the MDX cube source. All three reports use the same attribute, Product. This means that Product is used in the standard report as a project attribute, the Query Builder report has one object mapped to Product, and the MDX cube report uses Product to map one of its characteristics from the imported SAP BI query cube. Because data is joined by the common attribute Product, the document is generated successfully.



If each of the three reports originally has a prompt on Product, then the prompt will only be displayed one time when a user executes the document; this means a user only needs to answer the prompt one time, instead of three times.

Access and analyze multiple data sources with Query Builder

With Query Builder, you can access multiple data sources in the same project and use MicroStrategy reporting features to analyze your data. Connection to

databases, Excel file, and text files is described in [Connect to databases, Excel files, and text files, page 731](#).

Once you connect to a valid data source, you can create a Query Builder report to retrieve and analyze the data from the data source. The Query Builder Editor helps you create and modify SQL queries to run against your ODBC data sources, while requiring only a basic knowledge of SQL functionality. These queries are the basis of any Query Builder report, which enable you to analyze the data returned from the data source.

The sections below describe how to create Query Builder reports:

- [Create a Query Builder report to access relational databases, page 700](#)
- [Create a Query Builder report to access Excel files, page 703](#)
- [Create a Query Builder report to access text files, page 704](#)
- [Select and add tables, page 705](#)
- [Define joins, page 706](#)
- [Automatically join columns, page 709](#)
- [Select columns of data to return, page 712](#)

Create a Query Builder report to access relational databases

The following steps run through a scenario to create a Query Builder report that accesses the database provided with the MicroStrategy Tutorial project. The report created in these steps is used throughout the rest of this Query Builder section to demonstrate the different features available.



The MicroStrategy Tutorial project comes with a fully developed attribute and fact schema, which is not necessary to build Query Builder reports.

Prerequisites

- Query Builder reports can be created in MicroStrategy Developer only. However, these reports can be manipulated and executed from both MicroStrategy Developer and Web. Access to the Query Builder Editor is available only to Developers with the Define Query Builder Report privilege and those with the Create schema objects privilege.
- You must connect Query Builder to a database, which is described in [Connect to databases, page 732](#).

To create a Query Builder report

- 1 In MicroStrategy Developer, from the **File** menu select **New**, and then **Report**. The New Grid dialog box opens.
- 2 On the **Freeform Sources** tab, select **Create Query Builder report**.
- 3 In the **Source** area, select a database instance for the data source to access using Query Builder. For this example procedure, select the **Tutorial Data** database instance.
- 4 Click **OK**. The Query Builder Editor opens.

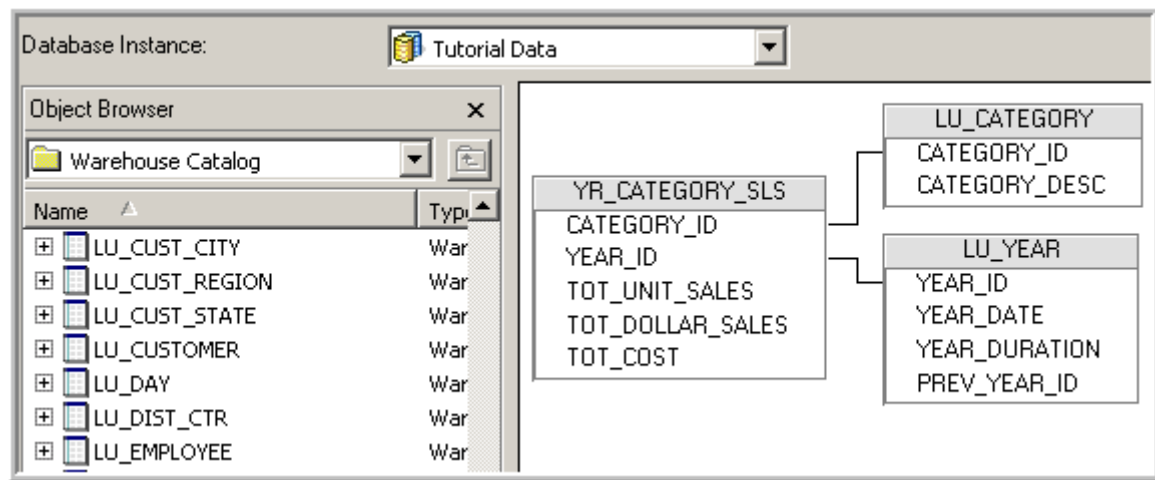
Add tables to your query definition

- 5 From the **Object Browser** drop-down list, select **Warehouse Catalog**. The list of imported tables along with the All Available Tables feature is displayed.
 - If the Object Browser is not displayed, from the **View** menu, select **Show Object Browser**. The Object Browser pane is displayed.
- 6 Select the **LU_YEAR** table and drag it into the Tables and Joins pane (the top pane on the right). A graphical representation of the LU_YEAR table is displayed within the pane. Drag the following additional tables into the Tables and Joins pane:
 - LU_CATEGORY
 - YR_CATEGORY_SLS

Create the joins

- 7 In the Tables and Joins pane, select the **CATEGORY_ID** column in the LU_CATEGORY table and drag it onto the CATEGORY_ID column in the YR_CATEGORY_SLS table. A line representing the join is drawn between the two columns.

- 8 Repeat the previous step to join YR_CATEGORY_SLS and LU_YEAR by the YEAR_ID column. The graphic below shows how your tables and joins should be defined:



Add columns to the select list

- 9 Double-click the **YEAR_ID** column in the LU_YEAR table. The column expression is added to the Selections pane (the bottom pane on the right). Add the following additional columns to the Selections pane:
- CATEGORY_ID from the LU_CATEGORY table
 - CATEGORY_DESC from the LU_CATEGORY table
 - TOT_DOLLAR_SALES from the YR_CATEGORY_SLS table

The image below shows all the columns after they have been added to the Selections pane:

Select	Expression	Alias	Object	Form	Type
☒	LU_YEAR.YEAR_ID		Year	ID	Number
☒	LU_CATEGORY.CATEGORY_ID		Category	ID	Number
☒	LU_CATEGORY.CATEGORY_DESC		Category	DESC	Text
☒	YR_CATEGORY_SLS.TOT_DOLLAR_S...		Revenue		Number

This report is created within the MicroStrategy Tutorial project, therefore the column expressions have been automatically mapped to the existing metadata objects. In a project without attribute or metric definitions, you must manually enter attribute and metric names, forms, and data types to be associated with each column expression. For more information on mapping columns to different metadata objects, see the section [Map columns to metadata objects, page 714](#).

10 Right-click the **YR_CATEGORY_SLS.TOT_DOLLAR_SALES** column expression, point to **Simple Aggregation**, and select **Sum**. This action modifies the column expression to include a sum aggregation.

11 Click **OK** to exit the Query Builder Editor. The Report Editor is displayed.

You can define the Query Builder report in the same way you define a standard report, adding features such as formatting, sorting, view filters, thresholds, exporting, and so on.

12 Save the Query Builder report as **Query Builder Demo**.



You must save the report first before you can run it.

13 Run the report. The image below shows the report in grid view:

Year	Category	Metrics	Revenue
2006	Books		650,192
	Electronics		6,027,843
	Movies		1,012,594
	Music		956,610
2007	Books		868,207
	Electronics		8,020,662
	Movies		1,345,502
	Music		1,283,235
2008	Books		1,121,696
	Electronics		10,342,798
	Movies		1,740,847
	Music		1,653,523

Notice that revenue data is displayed with a general format by default. You can format the Revenue metric's values to apply a currency format to the revenue data.

Create a Query Builder report to access Excel files

The Query Builder feature enables you to create reports that use data from Excel files. Follow the steps below to create a Query Builder report which retrieves data from an Excel file.

Prerequisites

- Query Builder reports can be created in MicroStrategy Developer only. However, these reports can be manipulated and executed from both MicroStrategy Developer and Web. Access to the Query Builder Editor is

available only to Developers with the Define Query Builder Report privilege and those with the Create schema objects privilege.

- You must connect Query Builder to an Excel file, which is described in [Connect to Excel files, page 733](#).

To create a Query Builder report from an Excel file

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and then select **Report**. The New Grid dialog box opens.
- 2 On the **Freeform Sources** tab, select **Create Query Builder report**.
- 3 In the **Source** area, select the database instance for your Excel file. For steps to connect to and create a database instance for an Excel file, see [Connect to Excel files, page 733](#).
- 4 Click **OK**. The Query Builder Editor opens.

You can now use the tables created by the Excel file to build your Query Builder report using the methods described in the sections listed below:

- [Select and add tables, page 705](#)
- [Define joins, page 706](#)
- [Automatically join columns, page 709](#)
- [Select columns of data to return, page 712](#)

Create a Query Builder report to access text files

The Query Builder feature enables you to create reports using data from text files. Follow the steps below to create a Query Builder report which retrieves data from a text file.

Prerequisites

- Query Builder reports can be created in MicroStrategy Developer only. However, these reports can be manipulated and executed from both MicroStrategy Developer and Web. Access to the Query Builder Editor is available only to Developers with the Define Query Builder Report privilege and those with the Create schema objects privilege.

- You must connect Query Builder to a text file, which is described in [Connect to text files, page 734](#).

To create a Query Builder report from a text file

- 1 In MicroStrategy Developer, on the **File** menu, point to **New**, and then select **Report**. The New Grid dialog box opens.
- 2 On the **Freeform Sources** tab, select **Create Query Builder report**.
- 3 In the **Source** area, select the database instance for your text file. For steps to connect to and create a database instance for a text file, see [Connect to text files, page 734](#).
- 4 Click **OK**. The Query Builder Editor opens.

You can now use the tables created by the text file to build your Query Builder report using the methods described in the sections listed below:

- [Select and add tables, page 705](#)
- [Define joins, page 706](#)
- [Automatically join columns, page 709](#)
- [Select columns of data to return, page 712](#)

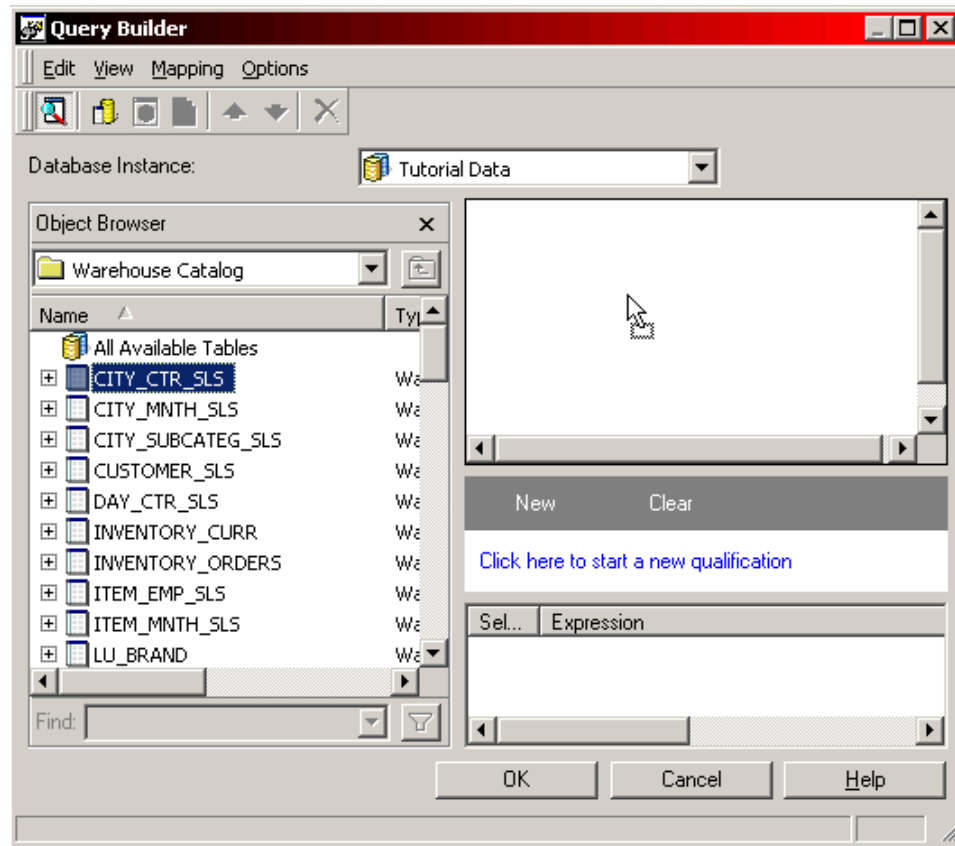
Select and add tables

Unlike a standard MicroStrategy report that queries against logical tables in the MicroStrategy metadata, Query Builder builds queries using a combination of imported database tables, columns, and joins.

Within the Query Builder Editor, you can browse both imported tables and all tables that exist for the database instance.

Tables that have been imported into the project's Warehouse Catalog are listed within the Warehouse Catalog folder in the Object Browser. These tables can be added directly to your Query Builder reports. To add tables to your Query Builder report, drag a table from the Object Browser on the left to

the Tables and Joins pane (the top pane on the right), as illustrated in the image below.



Access non-imported tables

The All Available Tables feature displays every table for the selected database instance. To run queries on non-imported tables, Query Builder can import tables from the data warehouse into the Warehouse Catalog. This allows you to import tables directly through Query Builder rather than using the Warehouse Catalog itself. You can import a table using the same method as adding a table to your Query Builder report.

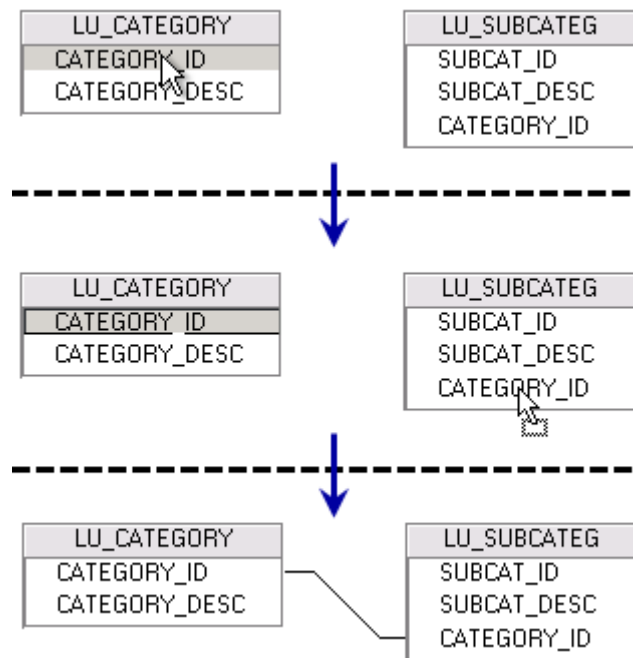
Accessing the All Available Tables feature can be costly in terms of execution time, therefore All Available Tables should be accessed only when it is necessary to import tables for the database instance.

Define joins

Once you have selected tables to include in your queries, you can define the joins between the columns of separate tables. Query Builder does not require

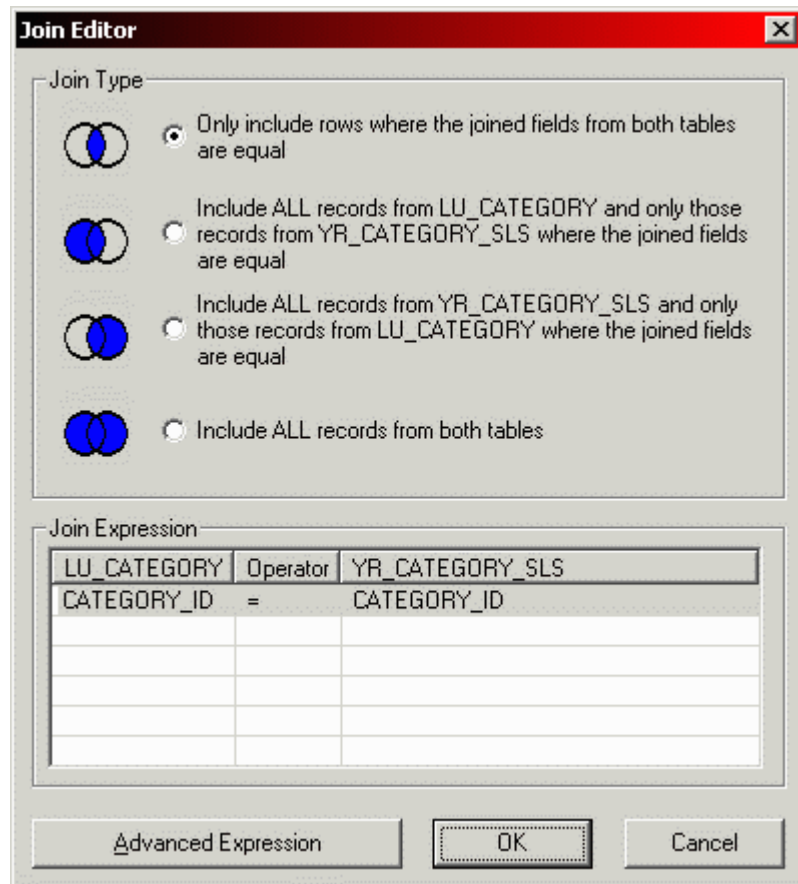
knowledge of the SQL syntax for the join operation, but a knowledge of how joins combine tables in SQL queries is essential. In a SQL query a join performs a relation on one or more columns stored in two separate tables and combines them into a single table. The table created by the join provides data that answers your query and can be displayed on your report.

You can create a join by selecting a column in a table and dragging it onto another column in a separate table as shown in the image below. The Query Builder Editor represents the join by drawing a line connecting the two columns.



 Query Builder can be configured to automatically join table columns by name and data type, or primary key and foreign key relationships. For more information on configuring Query Builder to automatically join the tables you include in your queries, see [Automatically join columns, page 709](#).

After creating a join between two tables you can double-click the line representing the join to open the Join Editor. The Join Editor, shown in the image below, allows you to modify the join operation.



The image shown above displays the settings for a join on two columns named `CATEGORY_ID`. The columns reside in the tables `LU_CATEGORY` and `YR_CATEGORY_SLS` that have been added to the Query Builder Editor.

Define the join type

The Join Editor allows you to select the join type for the join. The different join types that are supported are the following:

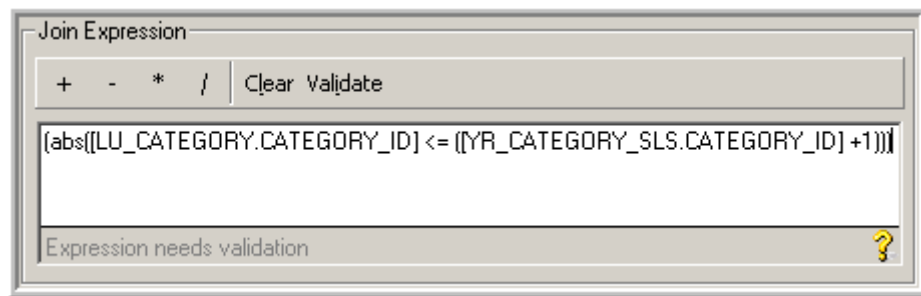
- Inner join: The first radio button in the image above performs an inner join on the columns.
- Left outer join: The second radio button in the image above performs a left outer join on the columns.
- Right outer join: The third radio button in the image above performs a right outer join on the columns.

- Full outer join: The fourth radio button in the image above performs a full outer join on the columns.

Modify the join expression

Query Builder allows you to modify the join expression at two different levels. The image above displays the simple join expression. With the simple join expression you can only modify the join operator. The Operator drop-down list allows you to apply different simple comparison operators (=, >, <, >=, <=) for the relation between the column expressions.

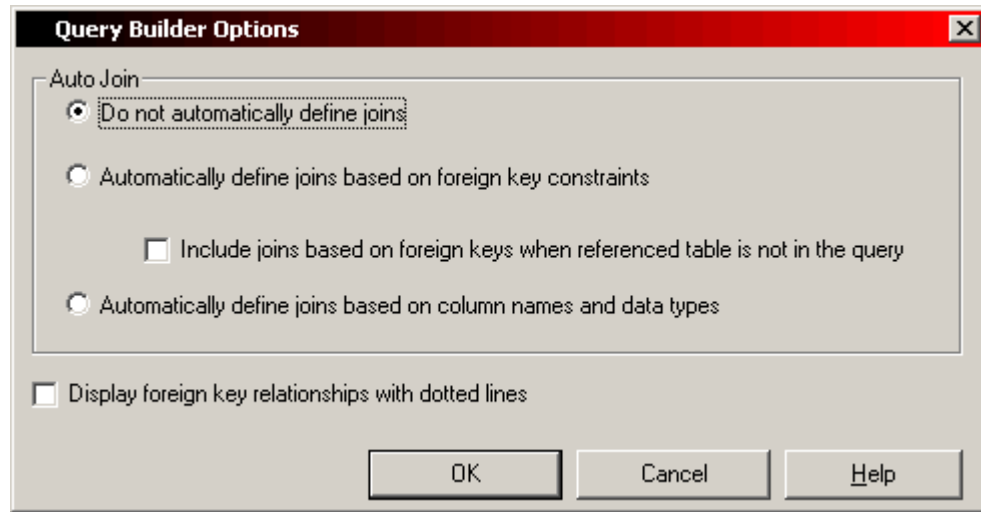
Within the Join Editor, you can select Advanced Expression to provide more options in which to modify the join expression. You can replace the join operator with any of the simple comparison operators described above. Arithmetic and functional expressions can be performed on the columns included in the join expression. The image below shows the Advanced Expression window containing a modified join expression.



Automatically join columns

In MicroStrategy version 8.0.3 and later, you can configure Query Builder to automatically join columns from the separate tables you add to your query. You can access the different automatic join options from the **Options** menu in the Query Builder Editor. When you click **Options** from the **Options**

menu, the Query Builder Options dialog box displays, as shown in the image below.



By default, Query Builder does not automatically join columns. If you keep this default, you must manually define all of the joins between tables added to your query. For more information on defining joins manually, see [Define joins, page 706](#).

Both automatic join options described below share the following behaviors:

- Select an automatic join option before adding tables to your query. Automatic joins are created when tables are added to your query. If you modify the automatic join option after you have added your tables to your query, no automatic joins are defined or deleted. For more information on adding tables to your query, see [Select and add tables, page 705](#).
- If you enable automatic joins, you can still manually define, modify, and delete joins. For more information on manually defining joins, see [Define joins, page 706](#).

Automatically define joins based on foreign key constraints

You can configure Query Builder to automatically join columns based on your database tables' primary key and foreign key constraints. When this option is selected, primary key columns are automatically joined with their associated foreign key columns.

For example, you have two lookup tables in your database. One table is named LU_CATEGORY for the Category attribute and one table is named LU_SUBCATEG for the Subcategory attribute. You define the CATEGORY_ID column as the primary key for the LU_CATEGORY table. The LU_SUBCATEG

table includes its own primary key `SUBCAT_ID` and the foreign key `CATEGORY_ID`. When you add the tables `LU_CATEGORY` and `LU_SUBCATEG`, a join is automatically created on `CATEGORY_ID` to join the two tables.

If you select to automatically define joins based on foreign key constraints, you also have the option to automatically create joins based on foreign keys when the table containing the primary key is not in the query. When this option is selected, the same foreign key columns in different tables are automatically joined even if the table that defines the column as a primary key is not in the query. However, the column must be defined as a primary key in at least one table in the database.

For example, recall the example scenario above. Along with the tables `LU_CATEGORY` and `LU_SUBCATEG` you have a lookup table named `LU_ITEM` for the attribute Item. The `LU_ITEM` table includes `CATEGORY_ID` as a foreign key. If you add `LU_SUBCATEG` and `LU_ITEM` to your query without adding `LU_CATEGORY`, a join is automatically created on `CATEGORY_ID` to join the two tables. Since no table is included in the query that defines `CATEGORY_ID` as a primary key, this join is created by searching the database tables for relevant primary keys. Once it is determined that `CATEGORY_ID` is a primary key, the join between `LU_SUBCATEG` and `LU_ITEM` on `CATEGORY_ID` as a foreign key is created. If `LU_CATEGORY` is included before or after the other two tables, the joins are created based on `CATEGORY_ID` as a primary key and any redundant joins are deleted.

You can update primary key and foreign key information for your database tables by updating the structure of your database tables in the Warehouse Catalog. For more information on updating table structure, see the *Project Design Guide*.

Automatically define joins based on column names and data types

You can configure Query Builder to automatically join columns based on column names and data types. When this option is selected, columns are automatically joined by matching columns by name and data type, as long as no primary key columns are included in the tables of your query.

For example, you have two lookup tables in your database. One table is named `LU_CATEGORY` for the Category attribute and one table is named `LU_SUBCATEG` for the Subcategory attribute. The `CATEGORY_ID` column is defined as an integer in both tables. Neither of the tables have any primary key or foreign key constraints. When you add the tables `LU_CATEGORY` and `LU_SUBCATEG`, a join is automatically created on `CATEGORY_ID` to join the two tables.

If the table LU_CATEGORY stores CATEGORY_ID as an integer and the table LU_SUBCATEG stores CATEGORY_ID as a character, a join is not automatically created. Similarly, if the table LU_CATEGORY stores the column as CATEGORY_ID and the LU_SUBCATEG table stores the column as CATEG_ID, a join is not automatically created. The latter scenario is an example of heterogeneous mapping. For more information on heterogeneous mapping, see *Chapter 3, Warehouse Structure for Your Logical Data Model* in the *Project Design Guide*.

If primary key columns are included in at least one of the tables of your query, joins are not automatically created based solely on column names and data types. In this case, joins are only automatically created between the primary key columns and any columns that match the primary key columns' column names and data types.

Display foreign key relationships with dotted lines

The check box at the bottom of the Query Builder Options dialog box allows you to show and hide the display of dotted lines between tables to indicate foreign key relationships.

Select columns of data to return

Query Builder reports return data from columns included in the SELECT clauses of your queries. The SELECT clause chooses which columns are returned in the result set of the report. You can include columns in the SELECT clause by adding the columns to the Selections pane (the bottom pane on the right). The Selections pane is displayed in the image below.

Select	Expression	Alias	Object	Form	Type
☒	LU_YEAR.YEAR_ID		 Year	 ID	Number
☒	LU_CATEGORY.CATEGORY_ID		 Category	 ID	Number
☒	LU_CATEGORY.CATEGORY_DESC		 Category	 DESC	Text
☒	Sum(YR_CATEGORY_SLS.TOT_DOLLAR_SALES)	Column1	 Revenue		Number

The Selections pane allows you to modify the column selections in the following ways:

- [Include columns in the SELECT statement, page 713](#)
- [Define the column expression, page 713](#)
- [Define an alias, page 714](#)

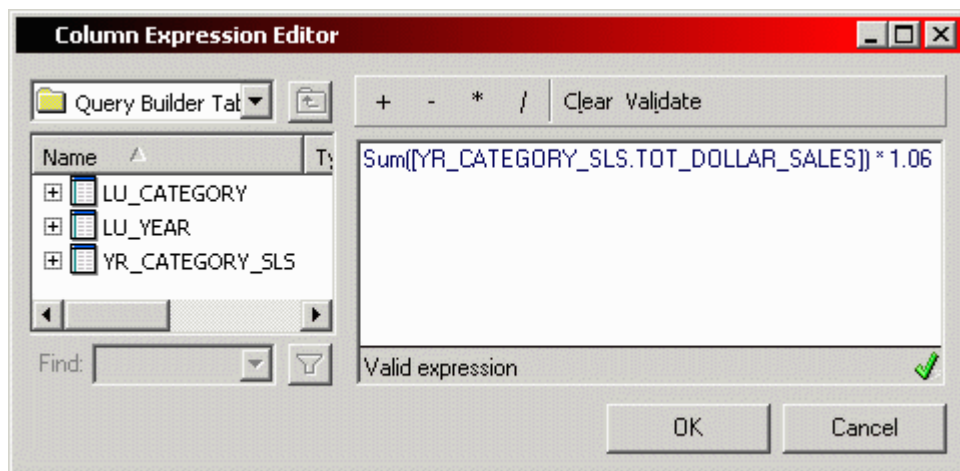
- [Map columns to metadata objects, page 714](#)
- [Define attribute forms, page 716](#)
- [Define the form data type, page 716](#)

Include columns in the SELECT statement

The check box in the Select column of the Selections pane determines whether the table column will be included in the `SELECT` clause. If the check box is selected, the column will be included in the `SELECT` clause. If the check box is cleared, the column is not included in the `SELECT` clause, but if it is a non-aggregated column it still appears in the `GROUP BY` clause.

Define the column expression

A column expression represents the column data, as well as any modifications to the data, that is returned for your report. You can right-click a column in the Selections pane and select **Edit Column Expression** to modify a column expression. The Column Expression Editor allows you to include functional and arithmetic expressions in the column expression. The image below shows the modifications available in the Column Expression Editor.



Define an alias

You can specify a column alias to use within the SQL statement of your queries. When viewing the SQL, a descriptive alias can help identify where the column is used in your queries.



The alias of a column is not displayed on a report run in grid, graph, or grid and graph view.

Map columns to metadata objects

When a column is added to the Selections pane, the column is mapped as either an attribute or metric, depending on its data type.



You can modify this default mapping in the Query Builder Editor by browsing to the Freeform Objects folder and dragging either <Add New Attribute Form> or <Add New Metric> onto the column in the Selections pane.

Query Builder supports history mapping of column expressions to simplify and improve the process of mapping column expressions to attributes or metrics. When you map a column expression to an attribute or metric in a Query Builder report, a connection between the column expression and the attribute or metric is saved. If you build another report with Query Builder and select the same column expression, the column expression is automatically mapped to the same metadata object you used in your previous Query Builder report.

For example, in the MicroStrategy Tutorial you can select the column `REGION_ID` from the `LU_REGION` table in Query Builder. If you map this column expression in Query Builder to the attribute `Region`, the `REGION_ID` column expression is automatically mapped to the attribute `Region` the next time you use the Query Builder feature and select the `REGION_ID` column.

Automatically mapping column expressions to attributes and metrics have the following behaviors:

- Each column expression can have multiple history mappings to choose from. For example, you can map a fact column `TOT_DOLLAR_SALES` from the `YR_CATEGORY_SLS` table to two different metrics, `Revenue` and `Sales`. The next time you select the `TOT_DOLLAR_SALES` column, the drop-down list includes the metrics `Revenue` and `Sales` to choose from.
- Attributes and metrics can have only one column expression saved as a history mapping. Only the most recent column expression mapped to the

attribute or metric is saved and used to automatically map attributes and metrics.

For example, consider the fact column `TOT_COST` from the `DAY_CTR_SLS` table. For the first Query Builder report, you map the expression as `Sum(DAY_CTR_SLS.TOT_COST)` to the metric `Cost`. The next time you build a report with Query Builder, you map the expression `Sum(DAY_CTR_SLS.TOT_COST*1.06)` to the metric `Cost`. This updates the `Cost` history mapping to be associated with the new expression. The next time you map the expression `Sum(DAY_CTR_SLS.TOT_COST)`, the metric `Cost` is not automatically mapped. However, if you change the expression to `Sum(DAY_CTR_SLS.TOT_COST*1.06)` the metric `Cost` is automatically mapped.

- You can choose to map a column to a different metadata object than is chosen by the automatic mapping.

If you previously defined a ROLAP schema by modeling attributes and metrics for your project, columns can be manually mapped to existing project attributes. For example, within the MicroStrategy Tutorial project, the `YEAR_ID` column from the `LU_YEAR` table can be mapped to the existing `Year` attribute. Mapping columns to attributes that are part of a ROLAP schema allows Query Builder reports to use the following MicroStrategy features:

- Security filters
- Element list prompts
- Shortcut-to-a-report qualification. For more information on using a Query Builder report as a shortcut-to-a-report qualification, see [Using Query Builder reports to filter other reports, page 726](#).

Columns in your Query Builder reports that are mapped to metrics can be mapped only to managed object metrics. Query Builder reports cannot use metrics previously created within your project that are built from an attribute and fact schema. However, managed object metrics can be mapped to columns in multiple Freeform SQL reports.

If a column is not mapped to an existing attribute or it is mapped to a metric, a managed object is created with the name you enter into the Object field. Once a managed object is created, you can map it to columns in other Query Builder reports by specifying the name of the managed object in the Object field. For example, if there is a managed object named `Year`, you can map the `YEAR_ID` column from the `LU_YEAR` table to this existing object by typing “Year” into the Object field. A new managed object is created if you do not enter the exact name in the Object field, such as “Years.” For more

information on managed objects related to Query Builder, see [Map data to non-project objects: Managed objects, page 735](#).

Define attribute forms

Each column expression which is mapped to an attribute must be mapped to an attribute form. The Form drop-down list includes the following options:

- ID
- DESC
- Any other forms that have been defined for the attribute



Each attribute must have a column mapped to the ID form of the attribute.

You can also type in a new form name to map the column to a new attribute form.



For more information on attribute forms, see the *Project Design Guide*.

Define the form data type

Attributes and metrics must map a form data type to each column expression. The Type drop-down list includes the following options:

- Number
- Text
- Datetime
- Date
- Time
- URL
- Email
- HTML Tag
- Picture

- Big Decimal



For more information on attribute form data types, see the *Project Design Guide*.

Reporting analysis features

This section discusses the following MicroStrategy reporting analysis features as they relate to Query Builder reports:

- [Filters](#)
- [Prompts](#)
- [Drilling](#)
- [Using Query Builder reports to filter other reports, page 726](#)

Filters

A filter specifies the conditions that data must meet to be included in the report results. For more information on filters, refer to [Chapter 3, Advanced Filters](#), in this guide.

You cannot use existing filters in a Query Builder report; however, you can filter report data by including a qualification that will be used in either the `WHERE` or `HAVING` clause of the SQL statement. The Conditions pane (the middle pane on the right) allows you to create qualifications for your queries.

The Conditions pane reflects whether your qualification is used in the `WHERE` or `HAVING` clause of the SQL query.

WHERE clause

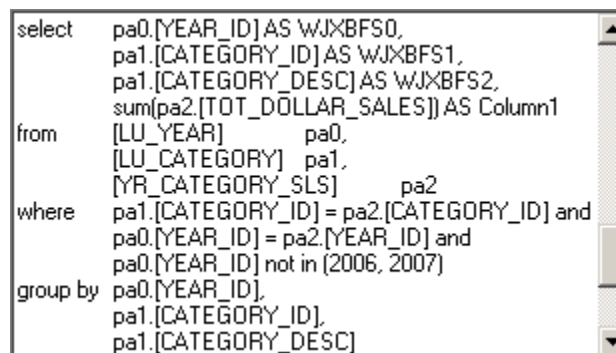
Qualifications within the `WHERE` clause of a SQL statement are evaluated before any aggregation is performed. For example, if your report includes Year, Category, and Revenue you can include a qualification that restricts your result set to data for the year 2006 only. This restriction is performed first, and then Revenue data is aggregated only for the year 2006, instead of aggregating data across all years.

If the qualification does not contain an aggregated column expression, it is included in the `WHERE` clause and appears at the top of the Conditions pane.

To create a WHERE clause qualification

This procedure uses the Query Builder Demo report created in the section [Create a Query Builder report to access relational databases, page 700](#).

- 1 In MicroStrategy Developer, right-click the **Query Builder Demo** report and select **Edit**. The Report Editor opens.
- 2 From the **Data** menu, select **Query Builder Definition**. The Query Builder Editor opens.
- 3 In the Conditions pane (the middle pane on the right), select **Click here to start a new qualification**. The Field, Operator, and Value drop-down lists become available.
- 4 From the **Field** drop-down list, select **LU_YEAR.YEAR_ID**.
- 5 From the **Operator** drop-down list, select **Not in list**.
- 6 From the **Value** drop-down list, select **Type Values**.
- 7 Type **2005, 2006** in the text field and click **OK**.
- 8 Click **OK** to exit the Query Builder Editor. The Report Editor is displayed.
- 9 From the **File** menu, select **Save as** and save the report as **QB Demo - Where Clause Filter**.
- 10 Run the report in SQL view. The WHERE clause of the SQL statement restricts the result set to exclude data for years 2006 and 2007, as shown in the image below.



```

select  pa0.[YEAR_ID] AS WJXBFS0,
        pa1.[CATEGORY_ID] AS WJXBFS1,
        pa1.[CATEGORY_DESC] AS WJXBFS2,
        sum(pa2.[TOT_DOLLAR_SALES]) AS Column1
from    [LU_YEAR]      pa0,
        [LU_CATEGORY] pa1,
        [YR_CATEGORY_SLS] pa2
where   pa1.[CATEGORY_ID] = pa2.[CATEGORY_ID] and
        pa0.[YEAR_ID] = pa2.[YEAR_ID] and
        pa0.[YEAR_ID] not in (2006, 2007)
group by pa0.[YEAR_ID],
        pa1.[CATEGORY_ID],
        pa1.[CATEGORY_DESC]
  
```

You can use the view filter functionality for Query Builder reports in the same way as for regular reports. For more information on view filters, see the *In-memory Analytics Guide*.

HAVING clause

Qualifications within the `HAVING` clause of a SQL statement are evaluated after any aggregation statement is performed. Therefore, any column expression with an aggregation function applied to it should be in the `HAVING` clause. For example, if your report includes Revenue aggregated with the sum function, you can include a qualification that restricts your result set to data with Revenue greater than \$200,000. This restriction is performed after the Revenue data is aggregated.

If the qualification contains an aggregated column expression, it is included in the `HAVING` clause and appears at the bottom of the Conditions pane (the middle pane on the right).

To create a HAVING clause qualification

This procedure uses the Query Builder Demo report created in the section [Create a Query Builder report to access relational databases, page 700](#).

- 1 In MicroStrategy Developer, right-click the **Query Builder Demo** report and select **Edit**. The Report Editor opens.
- 2 From the **Data** menu, select **Query Builder Definition**. The Query Builder Editor opens.
- 3 Select **Click here to start a new qualification**. The Field, Operator, and Value drop-down lists become available.
- 4 From the **Field** drop-down list, select **Column1**.



Column1 is the alias that has been automatically assigned to the column expression

`SUM(YR_CATEGORY_SLS.TOT_DOLLAR_SALES)`.
- 5 From the **Operator** drop-down list, select **Greater than**.
- 6 From the **Value** drop-down list, select **Type a value**. Type **200000** in the Value field.

- 7 Click **OK** to exit the Query Builder Editor. The Report Editor is displayed.
- 8 From the **File** menu, select **Save as** and save the report as **QB Demo - Having Clause Filter**.
- 9 Run the report in SQL view. The **HAVING** clause of the SQL statement restricts the result set to data with Revenue greater than \$200,000, as shown in the image below.

```

Pass0 - Duration: 0:00:00.00
select    pa1.[YEAR_ID] AS WJXBFS0,
          pa0.[CATEGORY_ID] AS WJXBFS1,
          pa0.[CATEGORY_DESC] AS WJXBFS2,
          sum(pa2.[TOT_DOLLAR_SALES]) AS Column1
from      [YR_CATEGORY_SLS]      pa2,
          [LU_CATEGORY]      pa0,
          [LU_YEAR]      pa1
where     pa2.[CATEGORY_ID] = pa0.[CATEGORY_ID] and
          pa2.[YEAR_ID] = pa1.[YEAR_ID]
group by  pa1.[YEAR_ID],
          pa0.[CATEGORY_ID],
          pa0.[CATEGORY_DESC]
having    sum(pa2.[TOT_DOLLAR_SALES]) > 200000

```

You can use the view filter functionality for Query Builder reports in the same way as for regular reports. For more information on view filters, see the *In-memory Analytics Guide*.

Prompts

A prompt is a MicroStrategy object that allows user interaction at report run time. For general information on prompts, refer to [Chapter 6, Advanced Prompts](#), in this guide.

For Query Builder reports, only two types of prompts are supported, which includes value prompts and element list prompts. Prompts can be included using different methods for the following panes in the Query Builder Editor:

- **Conditions pane** (the middle pane on the right): You can include element list prompts and value prompts in qualifications for your Query Builder reports. These prompts are inserted in the **WHERE** or **HAVING** clauses of your SQL query, depending on the type of qualification they are included in. For more information on creating qualifications in the Query Builder Editor, see [Filters, page 717](#).
- **Selections pane** (the bottom pane on the right): Element list prompts and value prompts can be referenced within a column expression in the

Selections pane, and are then used in the `SELECT` clause of the SQL query.

Include element list prompts in attribute qualifications

Element list prompts must be created prior to adding it to your Query Builder report. For more information on creating element list prompts, see [Choose from an attribute element list, page 245](#).

You can use existing element list prompts to qualify on a column which has been mapped to an attribute in the project. For more information on mapping columns to attributes in the project, see [Map columns to metadata objects, page 714](#).



Only project attributes can be used to create element list prompts in Query Builder reports.

Using an element list prompt in the WHERE clause

This procedure uses the Query Builder Demo report created in the section [Create a Query Builder report to access relational databases, page 700](#).

- 1 In MicroStrategy Developer, create an element list prompt that lists all elements of the attribute Year without restrictions. Save the prompt as **Year Element List Prompt**.
- 2 Right-click the **Query Builder Demo** report, and select **Edit**. The Report Editor opens.
- 3 From the **Data** menu, select **Query Builder Definition**. The Query Builder Editor opens.
- 4 Select **Click here to start a new qualification**. The Field, Operator, and Value drop-down lists become available.
- 5 From the **Field** drop-down list, select **LU_YEAR.YEAR_ID**.
- 6 From the **Operator** drop-down list, select **In list**.
- 7 From the **Value** drop-down list, select **Select a Prompt**. The Open dialog box opens.

- 8 Browse to the Year Element List Prompt you created and click **Open**. The prompt is inserted into the **Value** field.
- 9 Click **OK** to exit the Query Builder Editor. The Report Editor is displayed.
- 10 From the **File** menu, select **Save as** and save the report as **Query Builder Demo - Where Clause Element List Prompt**.
- 11 Run the report in grid view, selecting the Year element 2008 in the Prompt Answer Wizard. The report output is restricted to Year 2008, as shown in the image below:

		Metrics	Revenue
Year	Category		
2008	Books		1,121,696
	Electronics		10,342,798
	Movies		1,740,847
	Music		1,653,523



Notice that revenue data is displayed with a general format by default. You can format the Revenue metric's values to apply a currency format to the revenue data.

Include value prompts in attribute and metric qualifications

Value prompts can either be created prior to creating a Query Builder report, or value prompts can be created within the Query Builder Editor itself. For more information on creating value prompts, see [Value prompts, page 249](#).

These prompts can be used to qualify on a column which has been mapped to an attribute or metric. Value prompts do not require that attributes or metrics be included in the project schema. Therefore, you can use value prompts with columns which are mapped to managed objects. For more information on managed objects, see [Map data to non-project objects: Managed objects, page 735](#).

Creating a value prompt in the HAVING clause

This procedure uses the Query Builder Demo report created in the section [Create a Query Builder report to access relational databases, page 700](#).

- 1 In MicroStrategy Developer, right-click the **Query Builder Demo** report and select **Edit**. The Report Editor opens.

- 2 From the **Data** menu, select **Query Builder Definition**. The Query Builder Editor opens.
- 3 Select **Click here to start a new qualification**. The Field, Operator, and Value drop-down lists become available.
- 4 From the **Field** drop-down list, select **Column1**.
 Column1 is the alias that has been automatically assigned to the column expression
`SUM(YR_CATEGORY_SLS.TOT_DOLLAR_SALES)`.
- 5 From the **Operator** drop-down list, select **Greater than**.
- 6 From the **Value** drop-down list, select **Prompt a Value**. The Prompt Generation Wizard opens.
- 7 Step through the creation of the value prompt and click **Finish** to create the prompt.
- 8 Click **OK** to exit the Query Builder Editor. The Report Editor is displayed.
- 9 From the **File** menu, select **Save as** and save the report as **Query Builder Demo - Having Clause Value Prompt**.
- 10 Run the report in grid view by entering **2000000** in the Prompt Answer Wizard and clicking **Finish**. The report output is restricted to Revenue greater than \$2,000,000, as shown in the image below:

Year	Category	Metrics	Revenue
2006	Electronics		6,027,843
2007	Electronics		8,020,662
2008	Electronics		10,342,798

-  Notice that revenue data is displayed with a general format by default. You can format the Revenue metric's values to apply a currency format to the revenue data.

Include prompts in your column selections

You can include element list prompts and value prompts in the column selections for your Query Builder reports. These prompts are inserted in the **SELECT** clause of your SQL query, allowing you to control how data is selected and modified in the SQL statement before being returned for the

resulting report. For more information on selecting columns for your Query Builder report, see [Select columns of data to return, page 712](#).

You must create the element list prompt or value prompt prior to adding the prompt to a Query Builder report. For more information on creating prompts, see [Chapter 6, Advanced Prompts](#). To include a prompt in a column expression, you must type the prompt name in the expression using the following syntax: `?[prompt name]`

How the prompt is used in the column expression is dependent on the column expression itself. An element list prompt can only be created with a project attribute but it can be used in the column expression of a managed object. For example, you can map the column expression `YEAR_ID - ?[Choose from a list of years]` to a new managed object rather than the Year attribute used to build the prompt.

 Upon report execution, element list prompts are replaced by a comma-delimited list of attribute ID's. For example, if you select 2004 and 2005 in a Choose from a list of years prompt built from the Year attribute, then the SQL syntax includes "2004, 2005" where the prompt was placed.

 It is recommended that you do not include optional prompts in a column expression. Prompts that are included in the `SELECT` clause of a SQL query should require a prompt answer to avoid execution errors due to unanswered prompts.

Using a value prompt in the `SELECT` clause

This procedure uses the Query Builder Demo report created in the section [Create a Query Builder report to access relational databases, page 700](#).

- 1 In MicroStrategy Developer, create a numeric value prompt and save the prompt as **Select Clause Value Prompt**.
- 2 Right-click the **Query Builder Demo** report and select **Edit**. The Report Editor opens.
- 3 From the **Data** menu, select **Query Builder Definition**. The Query Builder Editor opens.
- 4 In the Selections pane (the bottom pane on the right), right-click the column expression **Sum(YR_CATEGORY_SLS.TOT_DOLLAR_SLS)** and select **Edit Column Expression**. The Column Expression Editor opens.

5 Modify the column expression to include your value prompt:

```
(Sum(YR_CATEGORY_SLS.TOT_DOLLAR_SALES) * ?[Select
Clause Value Prompt])
```

6 Click **OK** to return to the Query Builder Editor.

7 Click **OK** to exit the Query Builder Editor. The Report Editor is displayed.

8 From the **File** menu, select **Save as** and save the report as **Query Builder Demo - Select Clause Value Prompt**.

9 Run the report in SQL view by entering **1.06** in the Prompt Answer Wizard and clicking **Finish**.

The select statement includes

`(Sum(YR_CATEGORY_SLS.TOT_DOLLAR_SALES) * ?[Select Clause Value Prompt])` and replaces the prompt string with the value entered for the prompt in the **SELECT** clause, as shown in the syntax below:

```
Pass0 - Duration: 0:00:00.04
select  pa1.[YEAR_ID] AS WJXBFS0,
        pa0.[CATEGORY_ID] AS WJXBFS1,
        pa0.[CATEGORY_DESC] AS WJXBFS2,
        (sum(pa2.[TOT_DOLLAR_SALES]) * 1.06) AS Column1
from    [YR_CATEGORY_SLS]      pa2,
        [LU_CATEGORY]         pa0,
        [LU_YEAR]             pa1
where   pa2.[CATEGORY_ID] = pa0.[CATEGORY_ID] and
        pa2.[YEAR_ID] = pa1.[YEAR_ID]
group by pa1.[YEAR_ID],
        pa0.[CATEGORY_ID],
        pa0.[CATEGORY_DESC]
```

Drilling

Drilling allows you to look at specific data at levels other than what is originally displayed on the grid or graph. For standard reports, you can drill in different directions, such as down, up, or across attributes, based on the drill map. For Query Builder reports, support for drilling is limited to attributes within what is known as a personal Intelligent Cube. For a description of what a personal Intelligent Cube is, see the section [Intelligent Cubes, page 288](#).

Drilling in Query Builder reports works in the following ways:

- This functionality is controlled by the Drill within Intelligent Cube privilege, and requires an OLAP Services license.

- You can only drill to attributes that are included in the Report Objects pane but are not on the report grid.
- You can only use drill maps with project attributes, since managed object attributes do not have associated drill maps.
- You can drill from managed objects to any other project objects that are included in the Report Objects pane but are not on the report grid. You cannot drill from a managed object to another managed object.



For more information on managed objects, see [Map data to non-project objects: Managed objects, page 735](#).

- You can drill down and drill up on project attributes. For example, a Freeform SQL report has the Year and Quarter attributes on the report grid. When you move Quarter off the report to the Report Objects pane, you can drill down from Year to Quarter on the report. Additionally, if Quarter remains on the report and you move Year off the report, you can drill up from Quarter to Year.
- You can drill to attributes that are not in the same hierarchy as the attribute you are drilling from. For example, if Quarter from the Time hierarchy is on the report and Category from the Product hierarchy is only in the Report Objects pane, you can drill from Quarter to Category.



Managed objects do not have an associated hierarchy, so they are able to drill to any other project attribute that resides in the Report Objects pane but is not on the report grid.

Using Query Builder reports to filter other reports

Query Builder reports can be used to filter other standard reports in MicroStrategy. This can be supported by using a Query Builder report as a shortcut-to-a-report qualification on a standard report. For information on shortcut-to-a-report qualifications, see [About the report-as-filter, page 283](#).

However, there are additional configurations and requirements to allow Query Builder reports to be used as a shortcut-to-a-report qualification, as described below:

- The Query Builder report must map data to project attributes rather than managed object attributes. This is required in order for the standard report to recognize the data that is available on the Query Builder report. For information on mapping data in Query Builder reports to project attributes, see [Map columns to metadata objects, page 714](#).

- If the Query Builder report returns data from a data source that is not the primary data warehouse, you must have the Execute Multiple Source Report privilege. This privilege is available as part of MicroStrategy MultiSource. For information on MultiSource, see the *Project Design Guide*.

To add a Query Builder report as a shortcut-to-a-report qualification in a standard report, open the standard report in MicroStrategy Developer. In the **Report Filter** area, double-click the arrow to add a new qualification. In the **Filtering Options**, select **Add a Shortcut to a Report**. Click **OK**. Click the ... (browse button) to select a Query Builder report, and click **OK**. Click **OK** again to create the filter qualification.

Security for data access

MicroStrategy has a robust security model that enables you to create users and groups, determine how they are authenticated, control what data they can see, and what functional privileges they can have. For detailed information on these features, refer to the *System Administration Guide*. This section discusses the access control list (ACL) and security filters that relate to Query Builder reports only.

Access control list

An access control list (ACL) is a list of users and groups and the access permission that each one has to objects in a MicroStrategy project. Different users may have different permissions on the same object.

When you use existing objects (including project objects) in Query Builder column mapping, the ACL's of these objects are used. However, new attributes and metrics created in Query Builder reports inherit the default ACL defined in the Project Configuration Editor. You can modify the default ACL in MicroStrategy Developer by right-clicking a project and selecting Project Configuration. In the Project Configuration window point to Project definition, then Security, and then for Set Freeform SQL and MDX objects

default security select Modify. The Properties[XDA Objects] dialog box is displayed. The Permissions list has the following settings:

User	Children
Administrator	Full Control
Everyone	View
Public/Guest	View

 The user who creates the new attributes and metrics in Query Builder is automatically given the Full Control permission of the new objects.

In the Properties[XDA Objects] dialog box, you can change the settings of the default ACL for different groups of users.

 The changed settings will only affect the new attributes and metrics created subsequently in Query Builder reports, but not those objects created prior to the change.

Security filters in Query Builder

In MicroStrategy, a security filter is a filter object that is used to narrow down the result set that users or groups can view when they execute reports or browse elements. Usually assigned by administrators, security filters control what warehouse data users can see within MicroStrategy.

Once a security filter is created for users or groups, it is automatically applied when those users or groups view report data or browse attribute elements. However, you must perform some additional configuration to apply security filters to a Query Builder report. You configure your Query Builder report to apply security filter qualifications by inserting and configuring a placeholder for a user or group's security filter. Placeholders of this type created in Query Builder reports are referred to as "security filter placeholders" in this section.

Query Builder works with project security filters in the following ways:

- By default, Query Builder reports ignore security filters. The Report Designer has to insert a security filter placeholder as a qualification in the Conditions pane (the middle pane on the right) of a Query Builder report and configure it; otherwise, any user can run the Query Builder report without being limited in the data they see. For more information on creating security filter placeholders, see [Create security filter placeholders, page 729](#).

- Security filter qualifications are performed on Query Builder reports only if the attribute in the qualification is mapped to a column in the Query Builder report using a security filter placeholder. For more information on mapping attributes in security filter placeholders, see [Attribute Mappings, page 729](#).
- Security filter qualifications are ignored in Query Builder reports if the attributes in the qualifications are explicitly ignored in a security filter placeholder. For more information on ignoring security filter attributes in security filter placeholders, see [Ignored Attributes, page 730](#).
- Security filters which qualify on multiple attributes can be used with Query Builder reports only if every attribute in the security filter is either mapped to a column or ignored in a security filter placeholder. Query Builder reports fail for users with security filters containing multiple attribute qualifications that are not mapped or set to be ignored by a security filter placeholder.
- A security filter can restrict the attributes a user can view in relation to the level at which attributes are found within a MicroStrategy hierarchy. For more information on allowing top range and bottom range attributes in security filter placeholders see [Allow security filters with Top and Bottom levels to be evaluated based on the select level of this report, page 730](#). The two attribute range options are as follows:
 - **Top range attribute:** specifies the highest level of detail that the security filter allows the user to view.
 - **Bottom range attribute:** specifies the lowest level of detail that the security filter allows the user to view.

For more information on security filters in general, refer to the *Setting Up User Security* chapter in the *System Administration Guide*.

Create security filter placeholders

Security filter placeholders allow security filter qualifications to be included or ignored in the Query Builder report. You can create a Security filter placeholder as a qualification by right-clicking the Conditions pane (the middle pane on the right) and selecting **Insert Security Filter**. Within the Query Builder Security Filter Dialog that opens, you can create a security filter placeholder using the following options:

- **Attribute Mappings**

The Attribute Mapping pane is located on the upper right side of the Query Builder Security Filter Dialog. This is where you map attributes to

columns in the query to connect to security filter qualifications. For every attribute qualification to be recognized in security filters, you need to provide the form, table, and column mapped to that attribute in the query. For example,

Attribute	Form	Table	Column
Year	ID	LU_YEAR	YEAR_ID

- **Ignored Attributes**

The Ignored Attributes pane is located below the Attribute Mapping pane in the Query Builder Security Filter Dialog. This is where you specify attributes that may be ignored by the Engine when the report is being generated, even if they appear in a security filter qualification. This option can be useful if your Query Builder report does not include attributes that are included in security filters. By ignoring attributes that are not included in the Query Builder report, it allows security filters to apply qualifications for mapped attributes without creating any security holes. For example, if you define the following:

- **Attribute Mappings:** Customer is mapped to the CUSTOMER_ID column
- **Ignored Attributes:** Year is chosen to be ignored because it is not part of the Query Builder report
- **Security filter definition:** Year = 2006 and Customer = Bob

Users with the security filter defined above are able to run the report with the Customer = Bob qualification applied to the Query Builder report. There is no security hole because the Year attribute does not appear on the report.

You can ignore attributes that are included in the Query Builder report, but this can cause security holes because all qualifications on these attributes within security filters are ignored.

- **Allow security filters with Top and Bottom levels to be evaluated based on the select level of this report**

This check box option is located in the lower part of the Query Builder Security Filter Dialog. This option explicitly defines the **Select Level** of the report (displayed in the line just above this option) as the true level of

the data that is retrieved for the report when Top and Bottom criteria are evaluated.



Exercise caution with this option:

- Not selecting this option when the user has Top and Bottom levels defined will cause the report to fail.
- Select this option only when you are sure that your query does not retrieve data outside the restriction defined by the security filters. This means that you may need to check the security filters for individual users one by one and see how each one may interact with the query.

A security filter can be applied to a Query Builder report only if every attribute in the security filter is either mapped to a column or ignored. If only a subset of the attributes used in a security filter are either mapped to columns or ignored within the security filter placeholder, then the report will fail.

For example, a security filter that includes qualifications on Year and Category is applied for a group of users. In the Query Builder report, Year is the only attribute that has been mapped to a column in the security filter placeholder. Since Category is not included either as an attribute mapping or an ignored attribute, the report will fail for all users with this security filter.

Connect to databases, Excel files, and text files

With Freeform SQL and Query Builder, you can access multiple data sources in the same project and use MicroStrategy reporting features to analyze your data. Freeform SQL and Query Build also enable you to access Excel and text files as data sources.



For information on using the Freeform SQL Editor to report on third-party web services, see [Reporting on third-party web services with XQuery, page 638](#).

Before you can analyze data from databases, Excel files, or text files with Freeform SQL and Query Builder, you must connect to your desired data source. The sections listed below describe the steps required to connect Freeform SQL and Query Builder to your desired data source:

- [Connect to databases, page 732](#)
- [Connect to Excel files, page 733](#)

- [Connect to text files, page 734](#)

Connect to databases

The process of connecting Freeform SQL or Query Builder to a database involves the following general steps:

To configure a database instance

- 1 In MicroStrategy Developer, create a database instance for the data source that the Freeform SQL or Query Builder report will run against.
 You must create a database instance before defining a Freeform SQL or Query Builder report. For more information on creating a database instance, see the *Installation and Configuration Guide*.
- 2 To make the database instance available for Freeform SQL and Query Builder reports, right-click a project and select **Project Configuration**. The Project Configuration Editor opens.
- 3 Expand **Database instances**, and select **SQL Data warehouses**.
- 4 From the list of available database instances, select the check box for the database instance.
- 5 Click **OK** to save your changes and close the Project Configuration Editor.

For details on how to create a Freeform SQL report to access a database, see [Create a Freeform SQL report to access databases, Excel files, or text files, page 613](#).

For details on how to create a Query Builder report to access a database, see [Create a Query Builder report to access relational databases, page 700](#).

Connect to Excel files

The Freeform SQL and Query Builder reporting features enable you to create reports using data from Excel files. The connection process to Excel files involves the following steps:

- To connect to Excel file data, you must first prepare the Excel file as a valid data source and create a data source name for the Excel file. The steps to perform these configurations are included in *Appendix A: Connecting to Databases* of the *Installation and Configuration Guide*.
- [Create a database instance for an Excel file, page 733](#)

Create a database instance for an Excel file

To use an Excel file as a data source, you must create a database instance in MicroStrategy. This database instance is used to connect to the Excel file. Follow the steps below to create a database instance for an Excel file.

To create a database instance for the Excel file

- 1 In MicroStrategy Developer, create a new database instance that points to the DSN for the Excel file.
 It is recommended that you use Microsoft Excel 2000/2003 as the Data Connection Type for the database instance. For more information on setting up a database instance, see the *Installation and Configuration Guide*.
- 2 To make the database instance available for Freeform SQL reports, right-click a project and select **Project Configuration**. The Project Configuration Editor opens.
- 3 Expand **Database instances**, and select **SQL Data warehouses**.
- 4 From the list of available database instances, select the check box for the Excel file database instance.
- 5 Click **OK** to save your changes and close the Project Configuration Editor.

For details on how to create a Freeform SQL report to access your Excel file, see [Create a Freeform SQL report to access databases, Excel files, or text files, page 613](#).

For details on how to create a Query Builder report to access your Excel file, see [Create a Query Builder report to access Excel files, page 703](#).

Connect to text files

The Freeform SQL and Query Builder reporting features enable you to create reports using data from text files. The creation process involves the following steps:

- To connect to text file data, you must first prepare the text file as a valid data source and create a data source name for the text file. The steps to perform these configurations are included in *Appendix A: Connecting to Databases* of the *Installation and Configuration Guide*.
- [Create a database instance for a text file, page 734](#)

Create a database instance for a text file

To use a text file as a data source, you must create a database instance in MicroStrategy. This database instance is used to connect to the text file. Follow the steps below to create a database instance for a text file.

To create a database instance for the text file

- 1 In MicroStrategy Developer, create a new database instance that points to the DSN for the text file.
 It is recommended that you use Generic DBMS for the Data Connection Type. For more information on setting up a database instance, see the *Installation and Configuration Guide*.
- 2 To make the new database instance available for your Freeform SQL report, right-click a project and select **Project Configuration**, the Project Configuration Editor opens.
- 3 Expand **Database instances**, and select **SQL Data warehouses**.
- 4 From the list of available database instances, select the check box for the text file database instance.
- 5 Click **OK** to save your changes and close the Project Configuration Editor.

For details on how to create a Freeform SQL report to access a text file, see [Create a Freeform SQL report to access databases, Excel files, or text files, page 613](#).

For details on how to create a Query Builder report to access a text file, see [Create a Query Builder report to access text files, page 704](#).

Map data to non-project objects: Managed objects

A managed object is just like a normal object in MicroStrategy except that it is created by the system and is stored in a special system folder. When you create a Freeform SQL or Query Builder report, you can use existing attributes, including project attributes, to map to the column selections in the SQL query in the Freeform SQL Editor or Query Builder Editor. Alternatively, you can create new attributes, attribute forms, and metrics on the fly in Freeform SQL or Query Builder to which to map the columns in the SQL query; these new objects are managed objects.

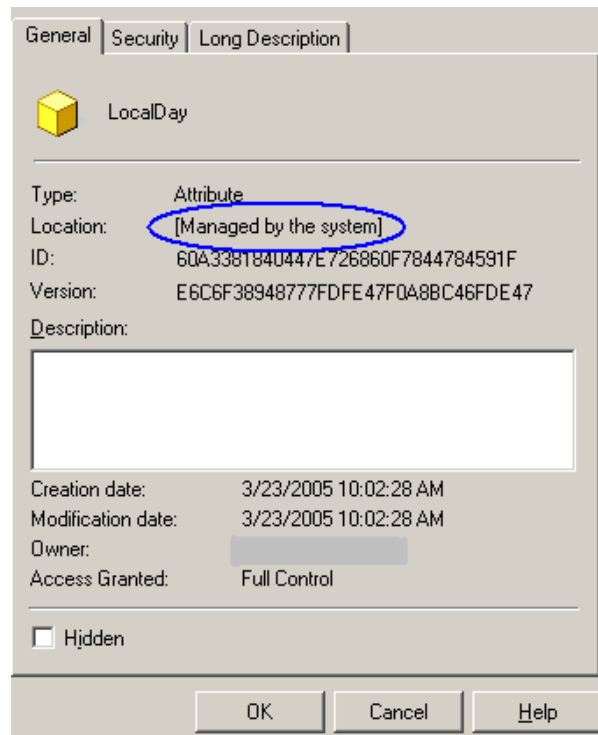


Managed objects are also created when integrating MDX cube sources into MicroStrategy. MDX cube sources include SAP BI, Microsoft Analysis Services 2000, 2005, and 2008, and Hyperion Essbase. For information on integrating MDX cube sources into MicroStrategy, see the *MDX Cube Reporting Guide*.

Column selections in your Freeform SQL or Query Builder reports that are mapped to metrics can be mapped only to managed object metrics. Freeform SQL and Query Builder reports cannot use metrics previously created within your project that are built from an attribute and fact schema. However, managed object metrics can be mapped to column selections in multiple Freeform SQL and Query Builder reports.

You can find out whether a given attribute or metric is a managed object or not by using the Properties dialog box. If it is a managed object, the Location

field on the General tab indicates Managed by the system as shown in the image below.



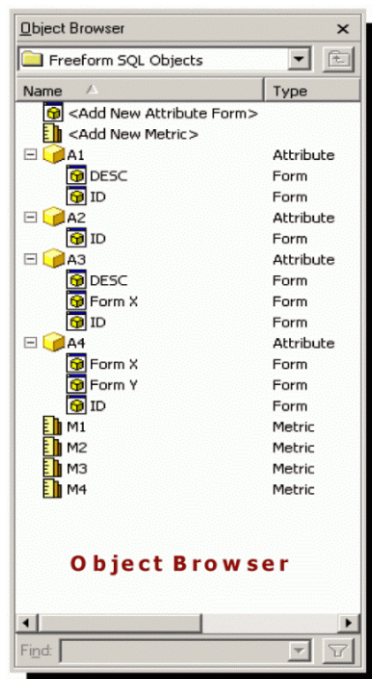
For every managed object (attribute or metric) created in Freeform SQL or Query Builder reports, you can perform the following tasks by using the right-mouse click function:

- From the **Search for Objects** dialog box, you can:
 - Edit
 - Rename
 - Delete
 - Search for dependents
 - Display in graphical viewer (for logical tables only)
 - Check or redefine settings
- From the **Report Editor**, you can:
 - Rename
 - Remove from report
 - Edit

- Search for dependents
- Check or redefine settings
- From the **Freeform SQL Editor** or **Query Builder Editor**, you can use the Object Browser to browse to the **Freeform Objects** folder and:
 - Search for dependents
 - Check or redefine settings

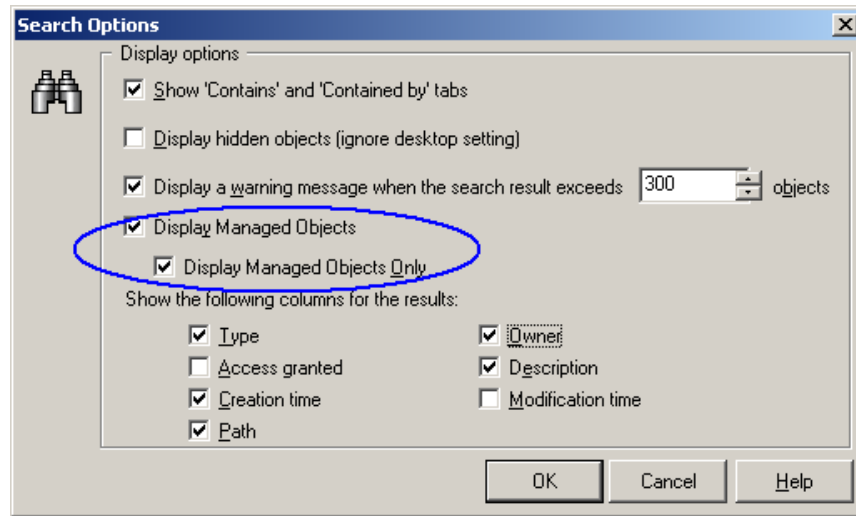
Access managed objects

You can view managed objects through the Freeform SQL Editor or Query Builder Editor, where you are defining your report. The Object Browser displays a Freeform Objects folder, as shown in the image below, that contains all the managed objects created for all your Freeform SQL or Query Builder reports. All the managed objects for a project are shown regardless of which database instance is used for which report. These objects do not have mappings outside of the Freeform SQL and Query Builder reports; therefore, they cannot be used in any standard reports.

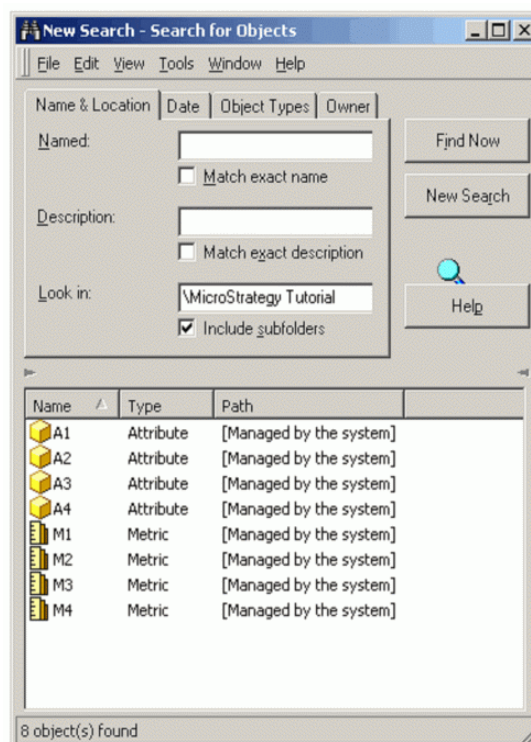


You can access managed objects by using the Search for Objects function. Make sure you select the **Display Managed Objects** option (in the Search

for Objects dialog box, select **Options** from the **Tools** menu) so managed objects will be displayed in the search result, as shown in the image below.



Once the managed objects are listed in the Search for Objects results shown in the image below, you can delete, rename, or edit any one of them by right-clicking its name.



Edit managed objects

In the Search for Objects dialog box, you have the option to edit a managed object (attribute or metric) when you right-click its name. The Attribute Editor or Metric Editor is displayed, where you can make the modification. You can also edit any managed attribute or metric directly from the Report Objects window of the Report Editor by following the steps in the procedure below.

Editing a managed object

- 1 After creating the Freeform SQL or Query Builder report in the Freeform SQL Editor or Query Builder Editor, the report is displayed in the Report Editor.
- 2 Save the report.
- 3 In the Report Objects window, right-click the name of the managed object that you want to modify and select **Edit**.

The Edit option is only available after the report has been saved and has not been changed since the last Save operation.

- 4 Depending on the object you selected, the Attribute Editor or Metric Editor is displayed.
- 5 Edit the object as needed.

When you close the Attribute Editor or Metric Editor, the Report Editor refreshes the object's definition, which is reflected in the Freeform SQL or Query Builder report the next time you run it.

Delete managed objects

If you decide that you no longer want to use some Freeform SQL or Query Builder reports, along with deleting the reports you can also delete their associated managed objects.

When you delete a Freeform SQL or Query Builder report, all the associated managed attributes and metrics used in the report are not deleted automatically. You can either delete managed objects one-by-one or delete Freeform SQL and Query Builder database instances along with all unused

managed objects. For more information on deleting managed objects and Freeform SQL and Query Builder database instances, see the *Managing Your Projects* chapter of the *System Administration Guide*.

Creating Intelligent Cubes

The Freeform SQL Editor and the Query Builder Editor provide alternative methods to create reports in MicroStrategy. In addition to creating reports directly, you can use the Freeform SQL Editor or Query Builder Editor to create an Intelligent Cube. This stores the data as an Intelligent Cube, which allows you to take advantage of various Intelligent Cube features, including the improved response time of reporting against Intelligent Cubes.

Once an Intelligent Cube is created, reports can be created based on the Intelligent Cube. These reports can analyze the Freeform SQL or Query Builder data, while also taking advantage of OLAP Services analysis features such as derived elements.

The steps below show you how to create an Intelligent Cube using Freeform SQL or Query Builder.

Prerequisites

- You must make the data sources to report on available for use with Freeform SQL and Query Builder. You can make data sources available for the following tasks:
 - To access data sources such as databases, Microsoft Excel files, and text files using the Freeform SQL Editor, see [Access and analyze multiple data sources with Freeform SQL, page 613](#).
 - To access web services using the Freeform SQL Editor, see [Allowing connections to web services in a project, page 639](#).
 - To access data sources such as databases, Microsoft Excel files, and text files using the Query Builder Editor, see [Access and analyze multiple data sources with Query Builder, page 699](#).
- You need the Use Intelligent Cube Editor privilege to create Intelligent Cubes. This privilege is part of OLAP Services.

To create an Intelligent Cube using Freeform SQL or Query Builder

- 1 Using MicroStrategy Developer, log in to a project connected to an MDX cube source.
- 2 From the **File** menu, select **New**, and then select **Intelligent Cube**. The New Intelligent Cube dialog box opens.
- 3 On the **Freeform Sources** tab, select one of the following options:
 - To create an Intelligent Cube by creating SQL statements using the Freeform SQL Editor, select **Create Freeform SQL report**.
 - To create an Intelligent Cube by creating XQuery statements using the Freeform SQL Editor, select **Create Freeform XQuery report**.
 - To create an Intelligent Cube using the Query Builder Editor, select **Create Query Builder report**.
- 4 In the **Source** area, select the database instance that connects to the data source to report on using Freeform SQL or Query Builder, and then click **OK**. The Freeform SQL Editor or Query Builder Editor opens.
- 5 You can now begin to create the Intelligent Cube using the standard techniques to create a Freeform SQL report or Query Builder report, which include:
 - *Create a Freeform SQL report to access databases, Excel files, or text files, page 613*
 - *Create a Freeform SQL report using a stored procedure, page 616*
 - *Creating a report to analyze web service data, page 659*
 - *Create a Query Builder report to access relational databases, page 700*
 - *Create a Query Builder report to access Excel files, page 703*
 - *Create a Query Builder report to access text files, page 704*
- 6 Once you have finished developing the Intelligent Cube, in the Report Editor, from the **File** menu, select **Save**. The Save Intelligent Cube As dialog box opens.
- 7 In the **Object name** field, type a descriptive name for the Intelligent Cube.

- 8 Click **OK** to save the Intelligent Cube and return to the Report Editor.
- 9 To make the Intelligent Cube available to be reported on, you must publish the Intelligent Cube. This can be done by selecting **Run Report** in the toolbar. For important details on publishing Intelligent Cubes, including prerequisites to consider before publishing an Intelligent Cube, refer to the *In-memory Analytics Guide*.
- 10 Once the Intelligent Cube is published, reports can be created on the Intelligent Cube. For steps on how to report on Intelligent Cubes, refer to the *In-memory Analytics Guide*.

ACCESSING SUBSETS OF DATA: DATA MARTS

Introduction

A data mart is a data repository where you store the results of a report as a relational table in a data warehouse. After creating a data mart, you can use it as a source table in your projects, and execute reports against it.

The primary objective of building a data mart is to provide an alternative location for storing frequently-used data that pertains to a particular business process or unit. This enables reports to pull data more effectively and efficiently for analysis and decision making. You can use data marts to achieve many business needs, including the following:

- Create aggregate fact tables. Aggregate fact tables improve the response time for reports that request summarized data from the warehouse.
- Create tables for very large results sets and then use other applications such as Microsoft Excel or Microsoft Access to access the data. This allows you to create a single table to feed a mass-mailing software tool or a similar closed-loop application.
- Create tables for offline analysis using direct SQL queries.
- If you have data in an MDX source, such as an SAP BW cube, you can transfer that data to a data mart and use features that cannot be used

directly on MDX reports. These features include, but are not limited to custom groups, consolidations, and metrics with complex definitions.

For information on using MDX data sources, see the *MDX Cube Reporting Guide*.

This chapter contains information on how to create a data mart in MicroStrategy Developer, by creating and executing a data mart report which then creates a data mart table. It also shows you how to run standard MicroStrategy reports against your new data mart table, using the data mart table as a source table in the project in which the data mart was created.

Getting started with data marts

Creating data marts involves creating two components:

- **Data mart report:** A metadata object that you create using the Report Editor. When this object is executed, it creates the data mart table in the warehouse of your choice. The data mart report contains attributes, metrics, and other application objects that translate into columns in the data mart table.
- **Data mart table:** A relational table that is created in the data warehouse when a data mart report is executed. You can use multiple data mart reports to modify a data mart table.

Creating data marts

To create a data mart table, you first create a data mart report that defines the columns of the data mart table. You then create the data mart table and populate it with data.

The steps below walk you through the process of creating a data mart report and then executing the report to create a data mart table. The steps also include an example for most steps, based on Tutorial sample data in the MicroStrategy Tutorial project.

Prerequisites

- You must specify a data warehouse in which to create the data mart table, using a database instance. The database instance specifies warehouse connection information, such as the DSN, user name, password, and other warehouse-specific information. You can use the following database instances as the data mart's database instance:

- Your project's primary database instance

To identify the project's primary database instance, in the Folder List, right-click the project's name, and select **Project Configuration**. Expand the **Database instances** category, and choose **SQL Data Warehouses**. Under the **Select the Primary Database Instance for the Project** drop-down, note which database instance is selected.

- A database instance other than your project's primary database instance, which points to the same warehouse as the primary database instance. You must ensure that the database instance has been created, and is available to the project, as described below.

You can also enable data mart optimization for this database instance, which can improve the performance of reports that use the data mart. For information on enabling data mart optimization, see [Enabling data mart optimization, page 754](#).

- A database instance other than your project's primary database instance, which points to a different data warehouse than the primary database instance. You must ensure that the database instance has been created, and is available to the project.



You must have the MicroStrategy MultiSource Option installed to use data marts that you save to a different database instance than your project's primary database instance.

- Ensure that any metric column aliases follow the naming conventions for your database, as they are used as column names in the data mart table. To change the metric column alias, open a metric in the Metric Editor. Under **Tools**, select **Advanced Settings** and select **Metric Column Options**. For information on column aliases, see [Metric column aliases: SQL identifiers and data types, page 123](#).
- If you need to run SQL statements before or after the data mart is executed, you must have the Use SQL Statements tab in Datamart/Bulk Export editors privilege in the Developer privileges.

To create a data mart

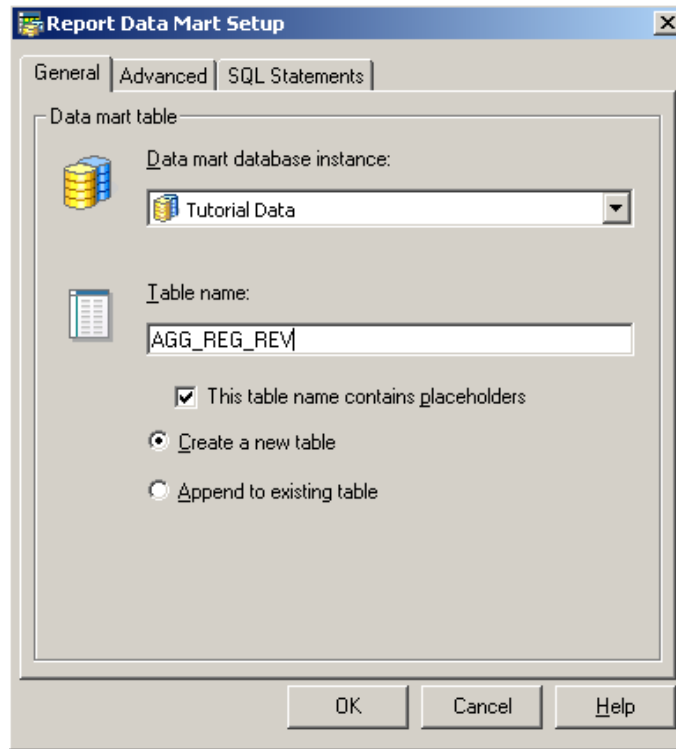
- 1 In MicroStrategy Developer, create a new report or select an existing report to use as the data mart table. The report should contain the attributes, metrics, and other objects that you want to use as columns in the data mart table and which will populate the data mart table when the data mart report is executed.



Your report cannot be used as a data mart if it contains any of the following:

- View filters
 - Report Objects that are not included in the template
 - Derived metrics
- For this example, use the sample Tutorial project to create a new report with Customer Region as the attribute and Revenue as the metric.
- 2 From the **File** menu, select **Save**, and select an appropriate folder in which to save the report.
 - For this example, save the report with the name My_Report, in a folder of your choice.

- 3 To use the report as a data mart report, from the **Data** menu, select **Configure Data Mart**. The Report Data Mart Setup dialog box opens, as shown below:



- 4 On the General tab, from the **Data mart database instance** drop-down list, select a database instance for the data mart table that will be created. The data mart table will be stored in this space.
 - For this example, choose **Tutorial Data**.
- 5 In the **Table name** field, type a table name that you want to associate with the database instance that you specified. This table name must be compliant with the naming conventions supported by your database.
 - The table name that you enter in this tab is not validated by the system. Ensure that the table name follows the naming convention rules for your database platform. If you do not use a valid table name, an error message is displayed.
 - For this example, name the table `AGG_REG_REV`.
- 6 To use a placeholder in the table name, select the **This table name contains placeholders** check box.

Placeholders allow you to modify table names dynamically according to your needs. The available placeholders for data mart table names are listed in the following table:

Placeholder	Replacement Options
!u	User name
!d	Date on which the table was created
!o	Report name
!j	Job ID
!r	Report GUID
!t	Timestamp
!p	Project Name
!z	Project GUID
!s	User session GUID

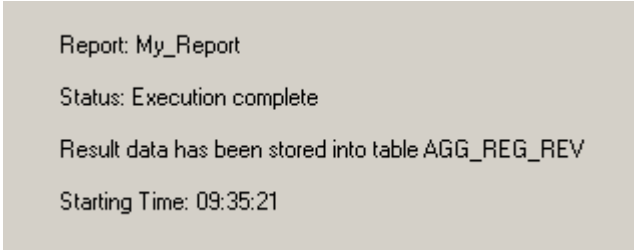
If you use a placeholder character other than those listed in the table, the placeholder is deleted from the table name.

- For this example, disable the **This table name contains placeholders** check box.
- 7 Select whether to **Create a new table** or **Append to existing table**, described below:
 - **Create a new table:** Select this option to replace the existing table each time the data mart report is run. The SQL statements drop and create the table each time the report is run.
 - **Append to existing table:** Select this option to add the data mart report results to an existing table.
 - For this example, select **Create a new table**.
 - 8 If you need to specify governors, table creation settings, and custom SQL for table creation, see [Specifying governors, table creation options, and custom SQL statements, page 757](#)
 - 9 Click **OK**.

Create the data mart table

- 10 Execute the data mart report. MicroStrategy creates the data mart table in the database you selected.

When the data mart table is created, the system displays a message that includes the data mart table name and a notification that the data mart table creation was successful, as shown in the example message below:



Report: My_Report
Status: Execution complete
Result data has been stored into table AGG_REG_REV
Starting Time: 09:35:21

Using a data mart table as a source table



The following procedures assume that you have an advanced knowledge about fact, metric, and attribute creation. For background information, see the *Project Design Guide*.

After creating a data mart table, you can use it as a source table in your projects. To do this, you must update the Warehouse Catalog for the project, verify that the new data mart table is identified as a source table for the attributes used in the data mart report, add the table as a data source for the appropriate fact, and update the project schema. The steps to do this are below.

To use a data mart as a source table in a project

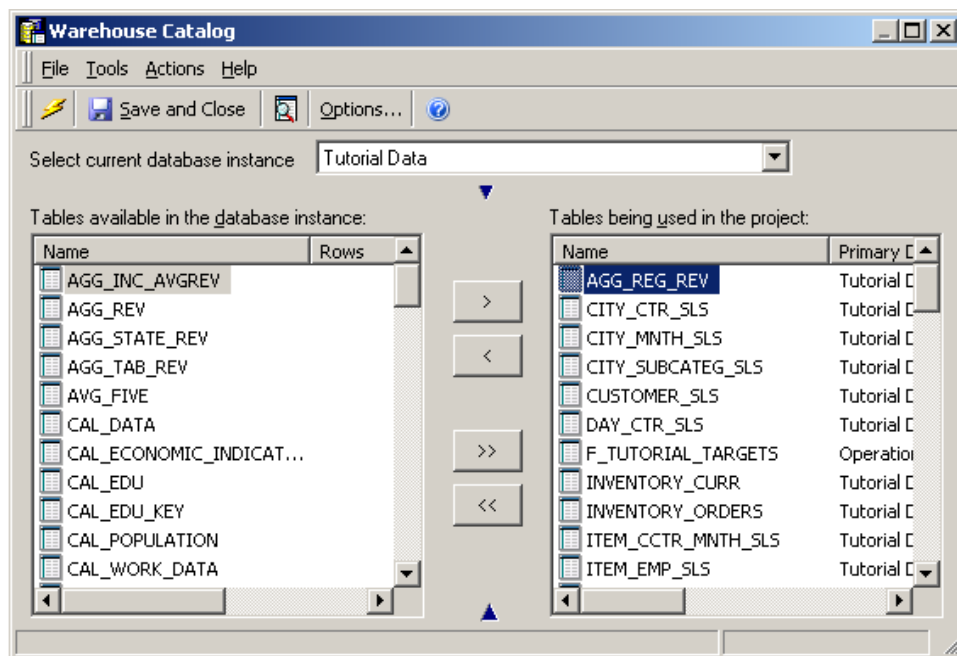
Update the Warehouse Catalog

- 1 In MicroStrategy Developer, from the **Schema** menu select **Warehouse Catalog**. The Warehouse Catalog opens.

- 2 From the **Select current database instance** drop-down list, select the database instance to which you saved the data mart table.

 You must have the MicroStrategy MultiSource Option installed to use data marts that you save to a different database instance than your project's primary database instance.

- 3 Select the newly created data mart table in the left pane. Then use the > arrow to move it to the right pane.
 - For this example, select the newly created data mart table AGG_REG_REV and move it to the right pane.

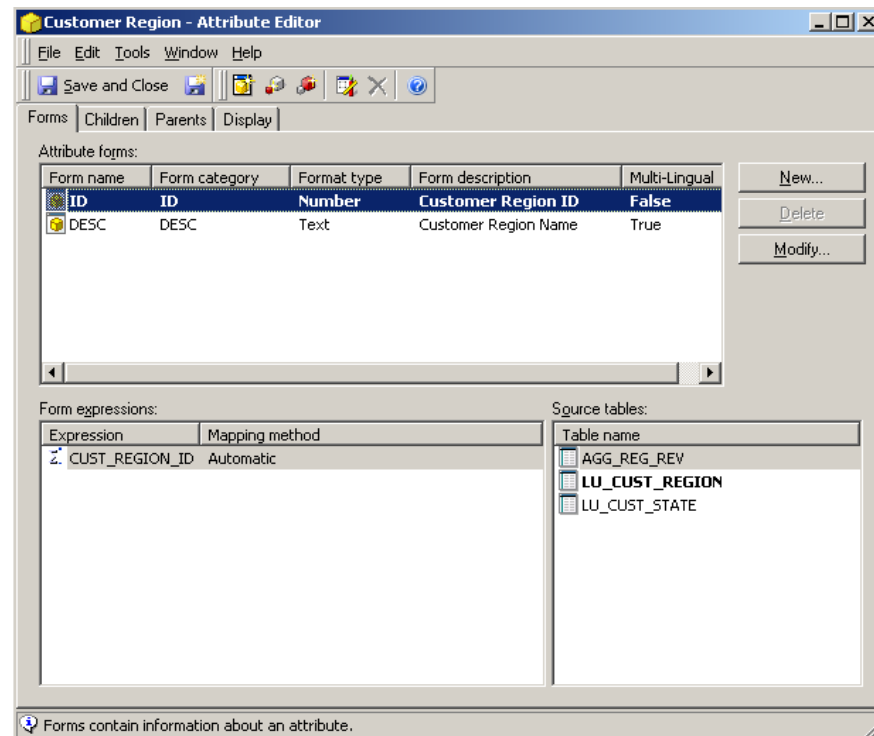


- 4 Click **Save and close** in the Warehouse Catalog to save the changes and close the Warehouse Catalog.

Verify that the new data mart table is identified as a source table for the attribute used in the data mart report

- 5 In MicroStrategy Developer, in the Folder List on the left, navigate to the folder which contains the attribute that you used in the data mart report. Then double-click the attribute. The Attribute Editor opens, as shown in the image below.
 - For this example, to verify that the new data mart table AGG_REG_REV is identified as a source table for the Customer Region attribute in the My_Report data mart report, navigate to the

Schema Objects folder, open the Attributes folder, and open the Customer folder. Double-click the Customer Region attribute.



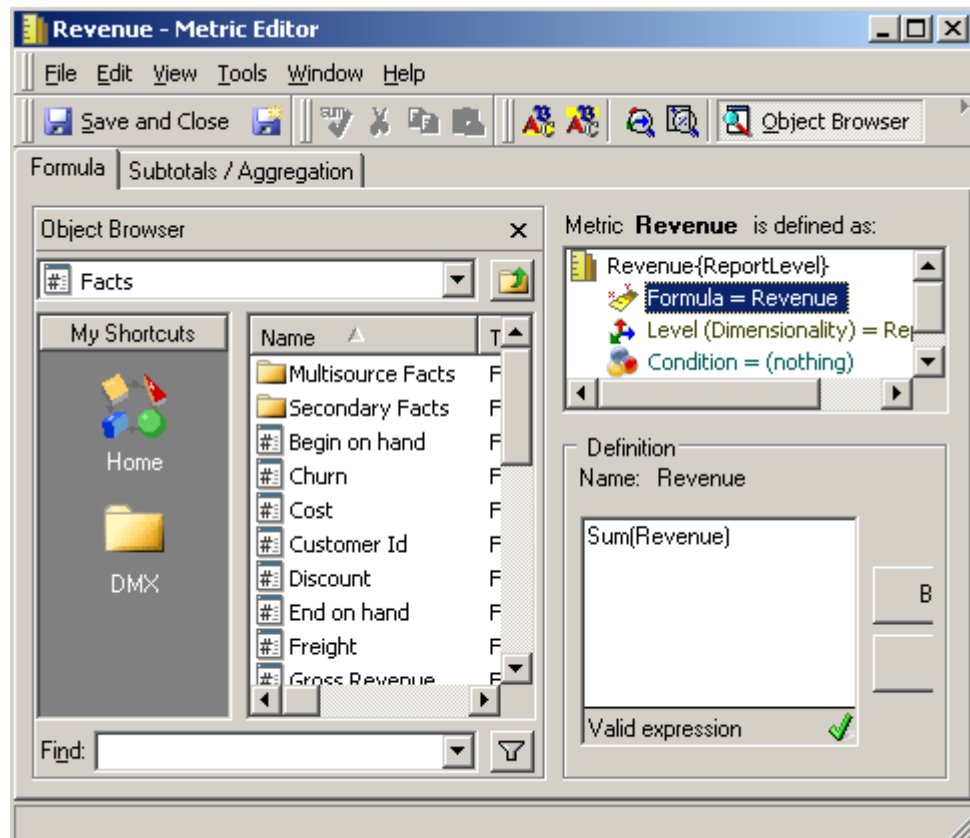
- 6 Look at the Source Tables pane at the bottom right. Ensure that your new data mart table name appears in the list of table names.

Add the data mart table as a source table for facts

To use the data mart table as a source table from which to execute reports, you must update the fact expressions on which the metrics used to create the data mart table are based.

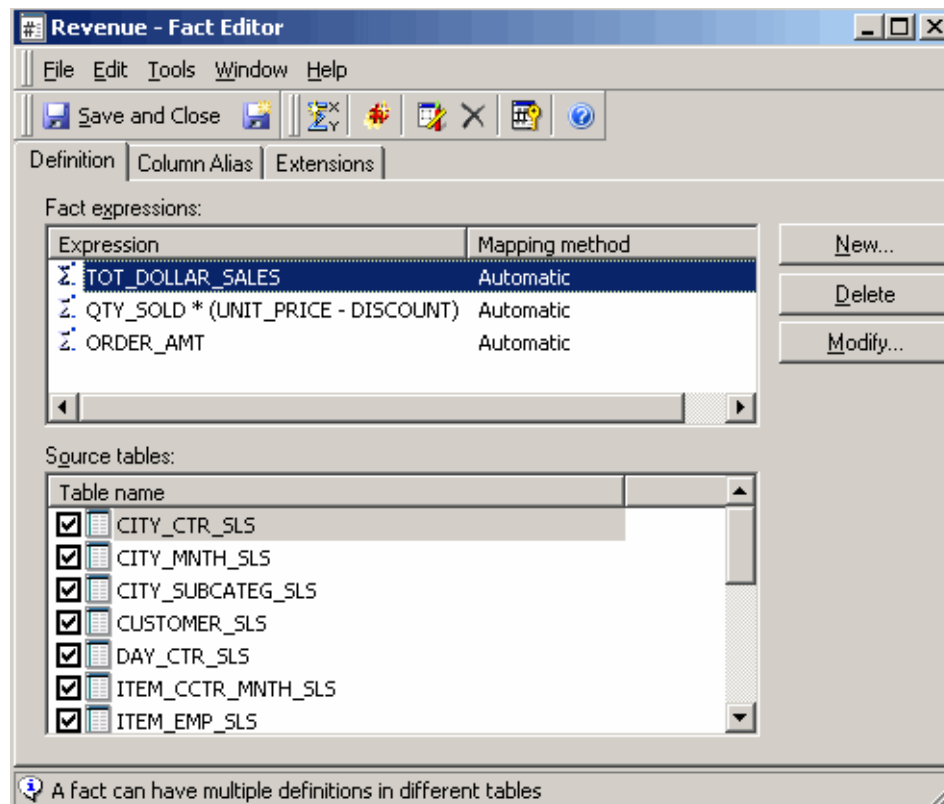
- 7 In MicroStrategy Developer, in the Folder List on the left, navigate to the folder which contains the metric that you used in the data mart report. Double-click the metric to open it in the Metric Editor. Note the fact on which the metric is based.

- For this example, navigate to the Metrics Folder and double-click the Revenue metric to open it in the Metric Editor. Note that the Revenue metric is based on the Revenue fact, as shown in the image below.



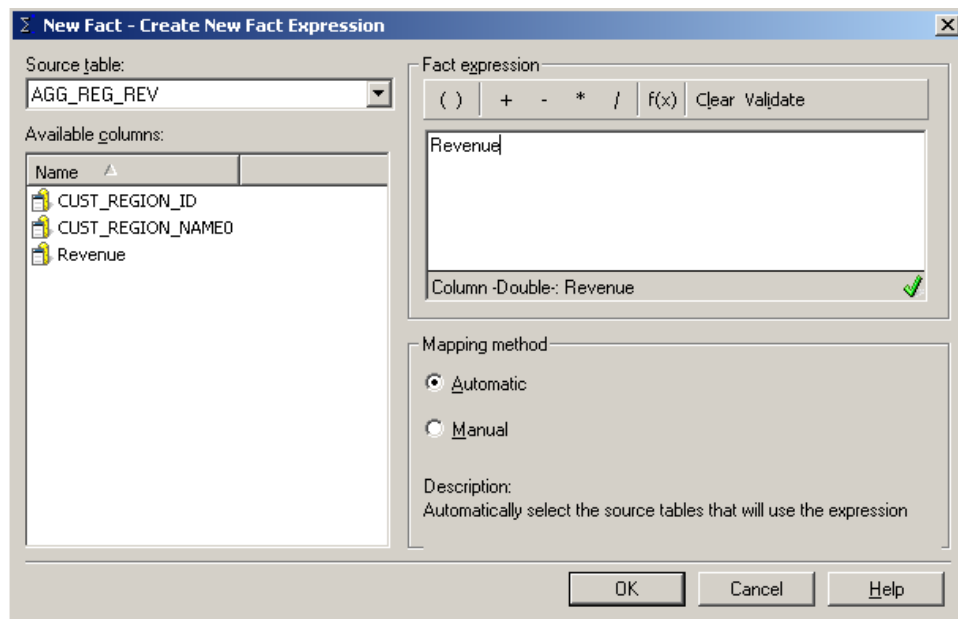
- In MicroStrategy Developer, in the Folder List on the left, navigate to the folder which contains the fact on which the data mart metric is based. Double-click the fact to open it in the Fact Editor.

- For this example, navigate to Schema Objects, then to the Facts folder. Double-click Revenue to open it in the Fact Editor.



- 9 Click **New**. This opens the Create New Fact Expression Editor.
- 10 From the **Source table** drop-down list, select the data mart table as the source table. Then double-click the fact associated with the data mart table, to add it to the Fact Expression pane.

- For this example, select AGG_REG_REV as the source data mart table. Then double-click Revenue to add it to the Fact Expression pane, as shown below:



11 Under Mapping method, select **Automatic**. This ensures that if more data marts are created with the same metric, they are automatically added as source tables for the fact.

12 Click **OK** to return to the Fact Editor.

13 Click **Save and Close** to save the changes.

Update the project schema

14 In MicroStrategy Developer, from the **Schema** menu, select **Update Schema**.

15 When prompted to select update options, ensure that all check boxes are selected and click **Update**.

Enabling data mart optimization

If you created the data mart table using a different database instance than the primary database instance used by your project, but which points to the same data warehouse, you can enable data mart optimization for the

database instance. This can improve the performance of reports that access the data mart.

For example, you have built a data mart report to create an aggregate fact table in your data warehouse. You have specified a database instance that is different from your project's primary database instance, but points to the same data warehouse as the latter. When you execute the report without using data mart optimization, Intelligence Server treats the database instance as one that points to a different database, and the following operations are performed:

- The results for the data mart are fetched from the warehouse into Intelligence Server's memory.
- A new table for the data mart is created in the data warehouse.
- The results are loaded into the table, one row at a time.

However, if you use data mart optimization, the following operations are performed:

- A new table for the data mart is created in the data warehouse.
- The results for the data mart are loaded into the new table directly, without being loaded into Intelligence Server's memory.

In this example, using data mart optimization thus results in simpler SQL queries, and can conserve Intelligence Server's resources.

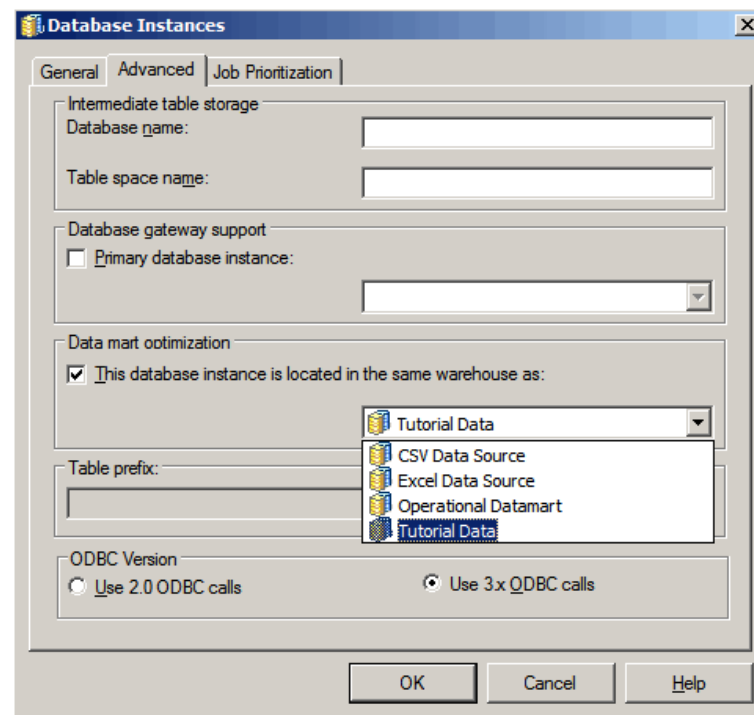


Do not use data mart optimization unless you are sure that the database instance for the data mart points to the same data warehouse as the project's primary database instance.

To enable data mart optimization

- 1 In MicroStrategy Developer, log in to your project source with administrative privileges.
- 2 In the Folder List, expand **Administration**, then expand **Configuration Managers**, and select **Database Instances**.
- 3 Double-click the database instance for which you want to enable data mart optimization. The database instance opens for editing.
- 4 Click the **Advanced** tab.

- 5 Under **Data mart optimization**, select **This database instance is located in the same warehouse as**, and from the drop-down list, select the primary database instance for your project.



Running a standard report against a data mart

To run a standard report against the data mart

- 1 Create a new report with the same definition as the data mart report.
 - For this example, create a new report with Customer Region as the attribute and Revenue as the metric.
- 2 Execute the report.
- 3 In the Report Viewer, from the **View** menu, select **SQL View**.

In the SQL view of the report, the `from` clause should show that the aggregated data is directly obtained from the recently created data mart table.

- For this example, the SQL view of the report shows that the SQL is accessing the data mart table AGG_REG_REV for data, as shown below:

Report details	
Report: new_report	
Report Cache Used: Yes	
Number of Columns Returned:	3
Number of Temp Tables:	0
Total Number of Passes:	1
Number of SQL Passes:	1
Number of Analytical Passes:	0
Tables Accessed:	
AGG_REG_REV	
LU_CUST_REGION	
SQL Statements:	
<pre>select a11.[CUST_REGION_ID] AS CUST_REGION_ID, a12.[CUST_REGION_NAME] AS CUST_REGION_NAME, a11.[Revenue] AS Revenue from [AGG_REG_REV] a11, [LU_CUST_REGION] a12 where a11.[CUST_REGION_ID] = a12.[CUST_REGION_ID]</pre>	
[Analytical engine calculation steps:	
1. Perform cross-tabbing	
]	

- Click **Save and Close**.

Specifying governors, table creation options, and custom SQL statements

You can use the information in this section to configure specialized or custom options when creating data mart tables. These advanced options can be modified at any time. If you change these options after the data mart report is initially saved and executed, you must save and re-execute the data mart report to re-create the data mart table with the new options.

Specifying governors and table creation options

The Advanced tab of the Report Data Mart Setup dialog box allows you to customize settings for data mart report creation. Here you can specify data mart governors and data mart table creation settings. These settings apply to the CREATE statement for the data mart table.

The settings for a data mart report override any settings at the project level.

To access the Advanced tab, open the data mart report in the Report Editor. From the **Data** menu, select **Configure Data Mart**. In the Report Data Mart Setup dialog box, select the Advanced tab.

The screenshot shows the 'Report Data Mart Setup' dialog box with the 'Advanced' tab selected. The dialog has three tabs: 'General', 'Advanced', and 'SQL Statements'. The 'Advanced' tab is active and contains two main sections: 'Governing' and 'Table Creation Properties'. In the 'Governing' section, 'Maximum execution time (sec):' is set to 0 and 'Maximum number of rows (Analytical Engine):' is set to 65536. The 'Table Creation Properties' section includes fields for 'Table Qualifier:', 'Table Descriptor:', 'Table Option:', 'Create Table Post String:', 'Table Space:', 'Table Prefix:', and 'Table Type:'. The 'Table Type:' dropdown is set to 'Permanent'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

- ⚡ If a field is left empty, the value for that setting is taken from the database instance you selected on the General tab.
- ⚡ The values entered on this tab are not validated by the system. Be sure you adhere to the syntax for your database.

Specifying performance parameters: Governing settings

These performance parameters include the maximum execution time and the number of rows to be sent to the Analytical Engine:

- **Maximum Execution Time:** Sets a time limit on report execution (in seconds). By default the Maximum execution time is set to 0 (zero), which means that the project level settings will be used.
- **Maximum Number of Rows (Analytical Engine):** This setting governs the maximum number of rows to send to the Analytical Engine at one time when the table is created. The default value is set to 65536.

Data mart VLDB properties: Table creation settings

These settings collectively are the VLDB properties for data marts. VLDB properties offer specialized configurations to support or optimize your reporting and analysis requirements. For background information on VLDB properties, see the *MicroStrategy Supplemental Reference for System Administration*.



The Table Creation settings are only applicable to permanent tables.

- **Table Qualifier:** Specify a string to be placed before the TABLE keyword in the CREATE TABLE statement.
- **Table Descriptor:** Specify a string to be placed after the TABLE keyword in the CREATE TABLE statement.
- **Table Option:** Specify a string to be placed after the table name in CREATE TABLE statements.
- **Create Table Post String:** Append a string after the CREATE TABLE statement.
- **Table Space:** Append a string at the end of the CREATE TABLE statement, but before any index definitions.
- **Table Prefix:** Specify a prefix for a table name in CREATE TABLE statements.
- **Table type:** Set whether the table is permanent or true temporary.

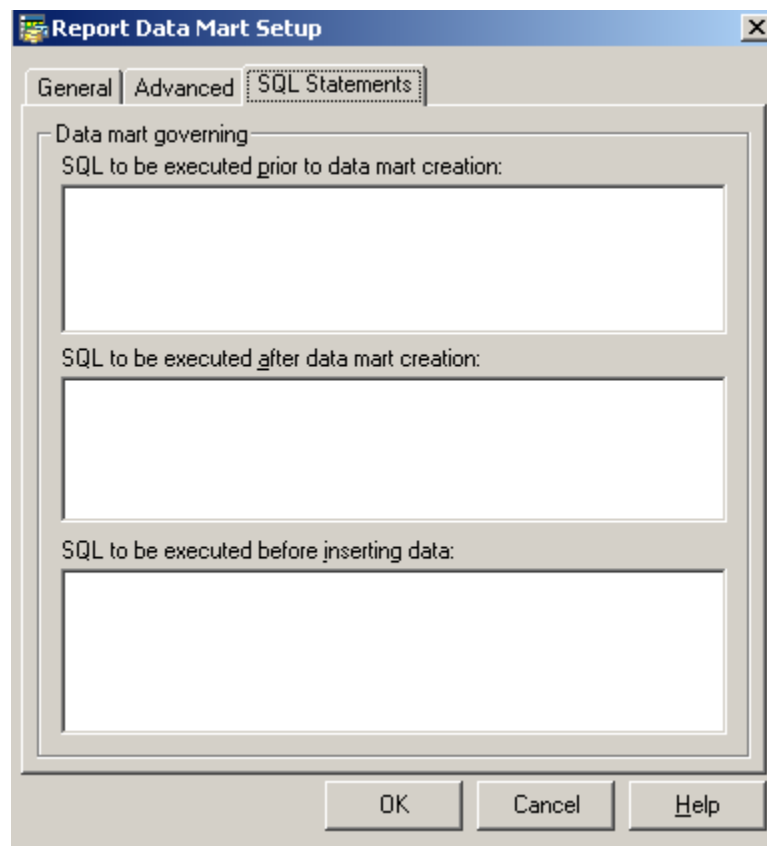
Specifying custom SQL statements

The SQL Statements tab of the Report Data Mart Setup dialog box allows you to enter custom SQL statements for a data mart table. You can add any

pre-creation or post creation SQL statements to your data mart table definition.

★ The SQL Statements tab is only visible if you have the Use SQL Statements tab in Datamart/Bulk Export editors privilege in the Developer privileges.

To access the SQL Statements tab, open the data mart report in the Report Editor. From the **Data** menu, select **Configure Data Mart**. In the Report Data Mart Setup dialog box, select the SQL Statements tab.



★ Custom SQL statements should be separated by semicolons. Press Enter after each SQL statement so that they are displayed on separate lines in the boxes on the SQL statements tab.

★ These SQL statements are not validated by the system. Be sure to follow the syntax rules for your database.

Type the SQL statements to be executed as described below:

- **SQL to be executed prior to data mart creation:** To execute SQL prior to the CREATE TABLE statement, type the SQL that you want to be executed.

- **SQL to be executed after data mart creation:** To execute SQL after the CREATE TABLE statement, type the SQL that you want to be executed.
- **SQL to be executed before inserting data:** To execute SQL after the CREATE TABLE statement and before the INSERT statement, type the SQL that you want to be executed.

You can also use placeholders to dynamically modify your queries. The following is a list of valid placeholders for custom SQL statements:

Placeholder	Replacement Options
!u	User name
!d	Date on which the table was created
!o	Report name
???	Temporary table names
!!!	All column names
!a	Attribute column names
!!	Exclamation (!)
!!=	Does not equal (!=)
!j	Job ID
!r	Report GUID
!t	Timestamp
!p	Project Name
!z	Project GUID
!s	User session GUID

DATA MINING SERVICES

Introduction

Data mining generally refers to examining a large amount of data to extract valuable information. The data mining process involves using the predictive models based on existing and historical data to project potential outcome for business activities and transactions. MicroStrategy Data Mining Services facilitates the development and deployment of these predictive models.

This chapter introduces MicroStrategy Data Mining Services, which includes these features:

- Using MicroStrategy to create multi-variate linear regressions, multi-variate exponential regressions, logistic regressions, decision trees, clustering predictive models, association rules, and time series models
- Support for importing third-party predictive models using the PMML industry standard
- A Predictive Model Viewer that visualizes the predictive model
- Data mining examples included in the MicroStrategy Tutorial project, in
 \Public Objects\Reports\
 MicroStrategy Platform Capabilities\
 MicroStrategy Data Mining Services

In addition, this chapter describes the process of how to create and use predictive models with MicroStrategy and provides a business case for illustration.

Data Mining Services

The ultimate goal of data mining is to find hidden predictive information from a large amount of data. The data mining process involves using existing information to gain new insights into business activities by applying predictive models, using analysis techniques such as regression, classification, clustering, and association. By extending MicroStrategy's powerful analytical, query, and reporting capabilities, MicroStrategy Data Mining Services can help organizations use their data to forecast future outcomes. Data Mining Services can be widely used in different industries and business areas, ranging from forecasting future results and customer behavior to classifying customers and estimating risk. A good example is an important area in marketing called campaign management. The goal of campaign management is to reduce the costs of marketing campaigns while increasing the positive response.

First, you gather data about the customers targeted for past campaigns, including information such as their age, gender, income, education, household size, and whether they responded positively or negatively to the campaigns. Next, you develop a MicroStrategy report to generate a result set, which is then analyzed to determine if positive responders shared any factors. Once these predictive factors are identified and a predictive model is developed, a MicroStrategy metric is created that embodies this predictive model. This metric forecasts who is likely to respond positively to similar, future campaigns.

By using this metric in a report, you only need to contact those customers on the narrowed-down list. This lowers your company's direct marketing costs and increases the effectiveness of the campaign. For details of this example, see [Campaign management example \(using logistic regression\)](#), page 836.

Approaches for data mining with MicroStrategy

You can incorporate MicroStrategy into the data mining workflow in a number of ways. These alternatives are grouped based on how the model is ultimately deployed and where the scoring takes place. The common

approaches for incorporating data mining within MicroStrategy are described below:

- [*Scoring the database, page 765*](#): Records are scored in batches and saved as tables or columns.
- [*Database does the scoring, page 766*](#): The database scores records in response to queries.
- [*MicroStrategy does the scoring, page 767*](#): MicroStrategy scores records using metrics and reports.

While MicroStrategy supports all three approaches, each has positive and negative aspects. The next sections describe each approach in detail.

Scoring the database

In this approach, records are scored and inserted into the database either as new tables or as new columns in existing tables. Most often, a third-party scoring engine receives a result set and scores the records. Then the scores are added to the database. Once they are part of the database, MicroStrategy attributes or metrics can reference those scores, just like any other data in the database. Historically, this approach has been the most common. Its pros and cons are described below.

Pros

- Since an external scoring engine performs the scoring calculation, model complexity and performance is hidden within the scoring engine. Thus, the scoring process does not require any database resources and does not impact other business intelligence work.
- At run time, data is simply read from the database without having to calculate the score on the fly. Scoring on the fly can slow analysis especially if millions of scores are involved.
- MicroStrategy can use this approach by just creating metrics or attributes for the scored data.

Cons

- This approach requires database space and the support of a database administrator.
- New records that are inserted after the batch scoring are not scored.

- Updating the model or scores requires more database and database administrator overhead.
- In many companies, adding or updating information in the enterprise data warehouse is not done easily or whenever desired. The cross functional effort required to score the database limits the frequency of scoring and prevents the vast majority of users from trying new models or changing existing ones.

This approach is really no different than adding other entities to a MicroStrategy project. For more information, see the *Project Design Guide*.

Database does the scoring

In this approach, data mining features of the database system are used to perform the scoring. Nearly all major databases have the ability to score data mining models. The most common approach persists the model in the database and then generates scores by using extensions to the SQL queries processed by the database to invoke the model. A key feature of this approach is that the model can be scored in a system that is different from the data mining tool that developed the model.

The model can be saved in the database as a Predictive Model Markup Language (PMML) object, or, less frequently, in some form of executable code. For more information on PMML, see [PMML overview, page 768](#). Persisting the model in this way is possible since the sophisticated algorithms needed to create the model are not required to score them. Scoring simply involves mathematical calculations on a set of inputs to generate a result. The ability to represent the model and score it outside of the model creation tool is relatively new, but more companies are adopting this approach. Its advantages and disadvantages are described below.

Pros

- Scores can be calculated on the fly even if new records are added.
- Updating the model is easier than in the Score the database option.
- This approach requires less database space than the score the database option.
- When the database supports accessing its data mining features via SQL, MicroStrategy can take advantage of this approach using its SQL Engine.

Cons

- This approach requires support from a database administrator and application knowledge of the database's data mining tool. However, the database administrator usually does not have this knowledge.
- The database data mining tool is typically an additional cost.

MicroStrategy has documented how to implement this approach for the IBM DB2 Intelligent Miner product. Contact MicroStrategy Technical Support for this Tech Note.

MicroStrategy does the scoring

In this approach, predictive models are applied from within the Business Intelligence platform environment, without requiring support from the database and from database administrators to implement data mining models. This direct approach reduces the time required, the potential for data inconsistencies, and cross-departmental dependencies.

MicroStrategy Data Mining Services uses enterprise data resources without significantly increasing the overhead. MicroStrategy Data Mining Services allows sophisticated data mining techniques to be applied directly within the business intelligence environment. Just as the other approaches, it also has advantages and disadvantages, as described below:

Pros

- MicroStrategy stores the predictive model in its metadata as a predictive metric that can be used just like any other metric.
- Scores can be done on the fly even if new records are added.
- The predictive model can be viewed in MicroStrategy Developer.
- The predictive model is easily updated using MicroStrategy Developer.
- This approach does not require database space or support from a database administrator.
- MicroStrategy can take advantage of this approach by using the Analytical Engine.

Cons

- This approach does not take advantage of the database data mining features.
- Predictor inputs need to be passed from the database to Intelligence Server. For large result sets, databases typically handle data operations more efficiently than moving data to MicroStrategy and scoring it there.

A key enabler of this process is MicroStrategy's ability to import predictive models using PMML. Therefore, it is necessary for you to have a basic understanding of PMML and how it is used. This is provided in the next section.

PMML overview

PMML (Predictive Model Markup Language) is an XML standard that represents data mining models. It was developed by the Data Mining Group (DMG), an independent consortium consisting of over two dozen companies including MicroStrategy. The language thoroughly describes how to apply a predictive model. It allows different model types, including the following:

- Regression
- Neural networks
- Clustering
- Trees
- Rule set
- Support vector machine
- Ensembles of models
- Association rules
- Time series

It also supports data transformation and descriptive statistics. PMML is generated by many data mining applications, including ANGOSS®, FairIsaac®, KXEN®, MicroStrategy, IBM®, Salford Systems®, SAS®, SPSS®, StatSoft®, and others. MicroStrategy can import PMML and also generate PMML for certain types of models.

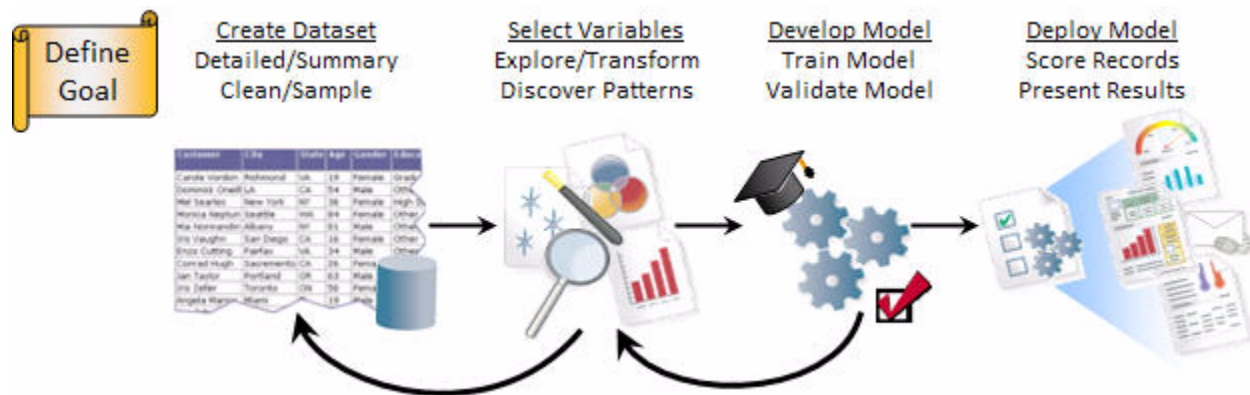
PMML is a major advance for the industry, as it allows the sophisticated, and sometimes esoteric, work of statisticians and data analysts to be easily

deployed to other environments. PMML closes the loop between data mining tools and the applications that use data mining models. Several data mining and database vendors have announced integrations based on PMML. MicroStrategy is the first business intelligence platform to support the standard. By allowing predictive metrics to be accessible to all users in the enterprise, MicroStrategy makes data mining for the masses possible.

For more information on PMML, check the website of DMG at <http://www.dmg.org> and other related documentation.

The Data Mining Services workflow

The process of creating a predictive model and incorporating it into the MicroStrategy business intelligence platform involves the following steps:



The process begins with the important step of defining a goal or objective for the analysis. The goal should be defined in business terms, identifying the results desired to improve some aspect of the organization's performance. Some common goals include improving the response to marketing campaigns, reducing customer attrition, forecasting trends, segmenting customers, and understanding relationships. The goal should be specific since the business objectives determine the approach used for predictive analysis and the data requirements.

With a thoroughly defined goal, the steps listed below are required to create and deploy a data mining model:

- 1 Create a dataset report to be used to develop the predictive model. See [Creating a dataset report, page 771](#) for details.

- 2 Create and train a predictive model from the dataset report, using MicroStrategy or a third-party application. For details, see [Creating a predictive model, page 785](#).
- 3 Create the predictive metric using MicroStrategy Developer. For details, see [Importing the predictive model, page 823](#).
- 4 Deploy the predictive metric in MicroStrategy reports to predict new outcomes, a process called scoring. See [Using the predictive metric, page 829](#) for information.

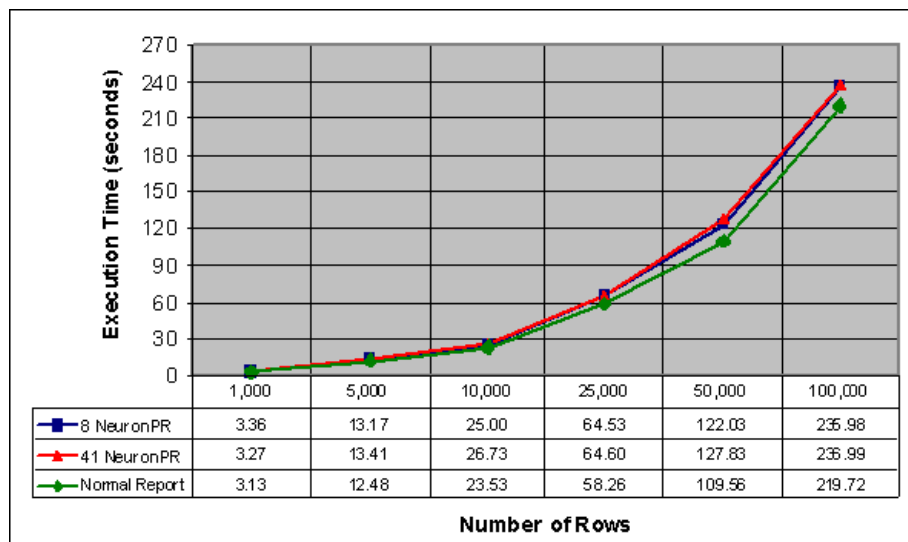


The predictive metric can also be used in filters, custom groups, or wherever other MicroStrategy metrics are used.

Once you have created a predictive model, you can view it using the MicroStrategy Predictive Model Viewer. See [Predictive Model Viewer, page 830](#) for details.

Predictive metrics and performance

When it comes to scoring complex predictive models like neural networks, people often have the impression that it takes a long time to perform these analytical calculations. Actually, scoring predictive metrics does not significantly increase the time required to run reports. The chart below shows the impact of two different types of neural network models, one with 8 neurons and one with 41. Compared to an equivalent report without predictive metrics, the reports with predictive metrics only take about 10% longer or less to run.



For more information, contact MicroStrategy Technical Support for the Tech Note that describes the impact of predictive metrics on performance.

Creating a dataset report

The first step in creating a predictive model is to develop a dataset report in MicroStrategy. It is recommended that you use MicroStrategy as the data mining source for the following reasons:

- **Enterprise data warehouse:** MicroStrategy projects typically use the enterprise data warehouse, which often contains clean, high-quality data. Since creating a good dataset report is typically 50-75% of the data mining effort, using MicroStrategy can greatly reduce that effort.
- **Analytical functions:** It is easy to create new metrics using MicroStrategy's vast library of over 200 analytical functions, which range from OLAP functions such as Rank, Percentile, and Moving Averages to advanced statistical functions. Raw facts in the database are easily turned into simple, and sometimes sophisticated, predictive inputs using these functions.
- **Relational databases:** MicroStrategy is optimized for all the major relational database vendors, which means that dataset reports are optimized to take advantage of these database capabilities.
- **Security model:** MicroStrategy's robust security model ensures that users can only access the data that they are permitted to access. The MicroStrategy business intelligence platform can address privacy-preserving issues as the predictive model is developed and also when it is deployed.
- **Easy handling of reports:** The dataset report can be easily created, refreshed, and accessed, even if it is large or contains complex calculations. Users do not have to be database administrators nor do they have to code queries to access the data they need.
- **Consistent use of variable definitions:** The exact definitions of variables that were used to develop the model are re-used when the predictive model is deployed. This process is performed automatically, which ensures the results are consistent with the way the model was developed.
- **Data sampling:** The dataset report can be sampled from a large number of records.

Data mining dataset reports

Data mining dataset reports have a very simple structure. The data usually focuses on a specific subject or attribute, for example, customers, transactions, or products; this information is used to develop a predictive model.

A dataset report is like a table in a database and usually has the following features:

- Each **row** represents a specific attribute, such as customer, transaction, or product.
- The **first column** is a unique identifier for the specific record, such as customer name, customer identification number, transaction number, or product SKU number.
- Each of the **remaining columns** of the dataset report contains data that describes the item in that row, such as customer age or annual purchases, transaction location or amount, or product color or cost. These columns can be either of the following:
 - Inputs to the predictive model, referred to as *predictive inputs* and also called *independent variables*
 - Representations of outcomes worth predicting, also called *dependent variables*

The following is an example of a part of a dataset report for customer information:

1	West	1 below 20	2 Female	4 60,001 - 80,000	3 Graduate	3 3	2 Owner	1 Single (Never	12
2	East	2 21-40	2 Female	1 0 - 20,000	1 High School	3 3	2 Owner	1 Single (Never	13
2	East	1 below 20	1 Male	1 0 - 20,000	4 Other	5 5	1 Renter	3 Divorced	2
2	East	4 61-80	1 Male	2 20,001 - 40,000	3 Graduate	3 3	1 Renter	1 Single (Never	13
2	East	1 below 20	1 Male	2 20,001 - 40,000	2 Undergraduate	1 1	1 Renter	2 Married	2
1	West	3 41-60	1 Male	3 40,001 - 60,000	4 Other	3 3	1 Renter	1 Single (Never	12
1	West	5 81 and above	2 Female	3 40,001 - 60,000	4 Other	2 2	1 Renter	3 Divorced	2
1	West	2 21-40	2 Female	5 80,001 upward	3 Graduate	2 2	1 Renter	3 Divorced	1
2	East	4 61-80	1 Male	2 20,001 - 40,000	4 Other	1 1	2 Owner	2 Married	2
2	East	2 21-40	1 Male	1 0 - 20,000	3 Graduate	7 7 or more	1 Renter	3 Divorced	1
2	East	1 below 20	2 Female	3 40,001 - 60,000	1 High School	4 4	2 Owner	2 Married	1
2	East	1 below 20	1 Male	5 80,001 upward	3 Graduate	5 5	2 Owner	2 Married	1
2	East	5 81 and above	1 Male	3 40,001 - 60,000	2 Undergraduate	1 1	1 Renter	3 Divorced	2
2	East	2 21-40	2 Female	3 40,001 - 60,000	3 Graduate	4 4	2 Owner	2 Married	1
1	West	5 81 and above	1 Male	3 40,001 - 60,000	2 Undergraduate	3 3	2 Owner	2 Married	1
1	West	2 21-40	1 Male	5 80,001 upward	3 Graduate	6 6	1 Renter	2 Married	1
1	West	4 61-80	1 Male	4 60,001 - 80,000	3 Graduate	2 2	1 Renter	1 Single (Never	12
2	East	5 81 and above	1 Male	2 20,001 - 40,000	1 High School	7 7 or more	2 Owner	1 Single (Never	12
2	East	2 21-40	2 Female	5 80,001 upward	2 Undergraduate	2 2	1 Renter	1 Single (Never	12
1	West	2 21-40	1 Male	5 80,001 upward	1 High School	2 2	2 Owner	1 Single (Never	12
2	East	4 61-80	2 Female	2 20,001 - 40,000	2 Undergraduate	5 5	1 Renter	3 Divorced	2
1	West	1 below 20	1 Male	2 20,001 - 40,000	3 Graduate	7 7 or more	1 Renter	1 Single (Never	12
2	East	3 41-60	2 Female	5 80,001 upward	3 Graduate	1 1	1 Renter	3 Divorced	1
3	Central	3 41-60	1 Male	3 40,001 - 60,000	3 Graduate	6 6	2 Owner	2 Married	1
1	West	1 below 20	1 Male	1 0 - 20,000	2 Undergraduate	1 1	1 Renter	3 Divorced	3
3	Central	5 81 and above	2 Female	3 40,001 - 60,000	2 Undergraduate	6 6	2 Owner	3 Divorced	1
2	East	2 21-40	1 Male	2 20,001 - 40,000	2 Undergraduate	6 6	2 Owner	3 Divorced	1
2	East	1 below 20	2 Female	5 80,001 upward	2 Undergraduate	2 2	1 Renter	2 Married	1
1	West	1 below 20	2 Female	5 80,001 upward	2 Undergraduate	1 1	2 Owner	1 Single (Never	12
2	East	2 21-40	2 Female	3 40,001 - 60,000	2 Undergraduate	5 5	1 Renter	3 Divorced	2
1	West	2 21-40	2 Female	2 20,001 - 40,000	4 Other	2 2	2 Owner	2 Married	2
2	East	5 81 and above	2 Female	4 60,001 - 80,000	4 Other	6 6	1 Renter	1 Single (Never	1
2	East	1 below 20	2 Female	1 0 - 20,000	1 High School	7 7 or more	2 Owner	1 Single (Never	12

Notice that each attribute, such as Customer Age Range, has two attribute forms on the report—the ID and the description. Some data mining software works better using numbers, such as the ID, while the description is included for ease of use.

Once the dataset report is ready, it can be used in a data mining analysis, usually in one of two ways:

- **Creating a predictive metric using MicroStrategy:** The dataset report is used to create a data mining model using MicroStrategy. More information on this approach can be found in [Creating a predictive model, page 785](#).
- **Creating a predictive metric with a third-party data mining tool:** The dataset report is made available for analysis by an external application, usually in one of the following ways:
 - The dataset report is created in the database as a table using MicroStrategy's Data Mart feature. Third-party data mining

applications can easily access databases using ODBC and SQL. This setup also promotes consistency between the dataset report used to develop the predictive model, especially for the variable names and data types. It does require database accessibility and database storage space. For more information on data marts, see [Chapter 12, Accessing Subsets of Data: Data Marts](#).

- The dataset report is exported to a particular file format using MicroStrategy's export capabilities. Third-party data mining applications can access many file formats such as Microsoft Excel, text files, and so on. Exporting files requires that the data type of each variable is determined on-the-fly by the data mining application. This interpretation may need correction by the user. On the other hand, this approach is usually easier for most people and does not require help with database administration. See the MicroStrategy online help for more information on exporting reports from MicroStrategy.

Guidelines for creating a dataset report

Before creating the dataset report, you need to make sure the attributes and metrics to be used as predictive inputs in the dataset report can also be used as inputs to the predictive metric. Recall that the predictive metric is the metric created from the predictive model after it is imported into MicroStrategy. By defining predictive inputs properly when you build the dataset report, you are guaranteed that the predictive model developed from that data receives the same inputs regardless of how that model is used in MicroStrategy.

You can also use an MDX cube as the dataset for your predictive model. This allows you to perform predictive analysis for your MDX cube data. The same guidelines for dataset reports listed below also apply to using MDX cubes as a dataset for your predictive model. For steps to integrate MDX cubes into MicroStrategy, see the *MDX Cube Reporting Guide*.

The following guidelines are intended to help you create your dataset report.

- **Use a flat report template.**

In a flat report template, attributes must be placed on rows and metrics on columns. Since the dataset report is a table consisting of a single header row followed by rows of data, placing attributes in the columns usually creates multiple header rows on each column, which cannot be easily represented in a database table.

- **Use attributes and metrics as predictive inputs.**

To create predictive metrics, you can use both metrics and attributes as inputs, as described in:

- [Attributes as inputs for predictive metrics, page 776](#)
- [Level metrics as inputs for predictive metrics, page 779](#)
- [Conditional metrics as inputs for predictive metrics, page 782](#)

You can use metrics and attributes as inputs for predictive metrics by creating a training metric (see [Creating a training metric with the Training Metric Wizard, page 808](#)).

- **Match each metric's level with the attribute used on the rows of the dataset report.**

The attributes on the rows of the dataset report control the level, or dimensionality, of the dataset report. If a metric is used in the predictive model without a level, the metric results change based on the attributes of the report using the predictive metric. Creating another type metric for the predictive metric, which sets the metric level, can resolve this problem. For more information, see [Level metrics as inputs for predictive metrics, page 779](#).

- **Set a condition on the metric that provides the proper grouping, when grouping a metric's results by an attribute.**

A metric condition is essentially a filter that allows an attribute to qualify metrics. For example, you can display customer revenue by payment method. For more information, see [Conditional metrics as inputs for predictive metrics, page 782](#).

- **Set the attribute join type for the report.**

Setting the attribute join type ensures that any missing data does not cause rows to be deleted. To do this, complete the following procedure.

To set the attribute join type for the report

- 1 Open the report in the Report Editor.
- 2 Select **Report Data Options** from the **Data** menu.
- 3 Under **Categories**, expand **Calculations**, and then select **Attribute Join Type**.

- 4 Clear the **Use Defaults** check box.
- 5 Set the attribute join type of the report to either of the following:
 - **Preserve lookup table elements joined to the final pass result table based on the template attributes with filter** keeps all attribute elements and applies all related filtering conditions.
 - **Preserve lookup table elements joined to the final pass result table based on the template attributes without filter** keeps all attribute elements and ignores all related filtering conditions.
- 6 Click **OK**.

Inputs for predictive metrics

A predictive metric can be created using attributes and metrics as its inputs. How you define the attributes and metrics you use as inputs for your predictive metrics affects the resulting predictive metrics, as described in:

- [Attributes as inputs for predictive metrics, page 776](#)
- [Level metrics as inputs for predictive metrics, page 779](#)
- [Conditional metrics as inputs for predictive metrics, page 782](#)

Attributes as inputs for predictive metrics

Attributes can be used as inputs for predictive metrics. Data mining often analyzes non-numeric, demographic, and psychographic information about customers, looking for attributes that are strong predictors.

For example, your MicroStrategy project contains a Customer attribute with related attributes for age, gender, and income. You can include an attribute, such as the Customer attribute, directly in a training metric, as described in [Creating a training metric with the Training Metric Wizard, page 808](#).

By including an attribute directly in a training metric, a predictive metric is then created that includes the attribute as one of its inputs. When using attributes directly in training metrics to create predictive metrics, be aware of the following:

- The ID attribute form for the attribute is used by the training metric to include the attribute information in a predictive metric. If attributes

include additional attribute forms other than the ID form that are to be used as inputs for predictive metrics, you can create metrics based on these attribute forms. Once these metrics are created, they can then be used as inputs for predictive metrics. This scenario for creating attribute-based predictive metrics is described in [Creating metrics to use additional attribute forms as inputs for predictive metrics](#) below.

- Attribute forms must use a text or numeric data type. If the attribute form uses a date data type, the data cannot be correctly represented when creating the predictive metric. If an attribute form uses date values, you must convert the date values into a numeric format to use the attribute form to create predictive metrics.

Creating metrics to use additional attribute forms as inputs for predictive metrics

If attributes include additional attribute forms other than their ID form that are to be used as inputs for predictive metrics, you can create metrics based on these attribute forms. The resulting metric can then be used as an input for a predictive metric, thus allowing the attribute information to be included in a predictive metric.

The steps below show you how to create a metric based on an attribute form. The resulting metric, which contains the attribute information, can then be used to create a predictive metric.

Prerequisites

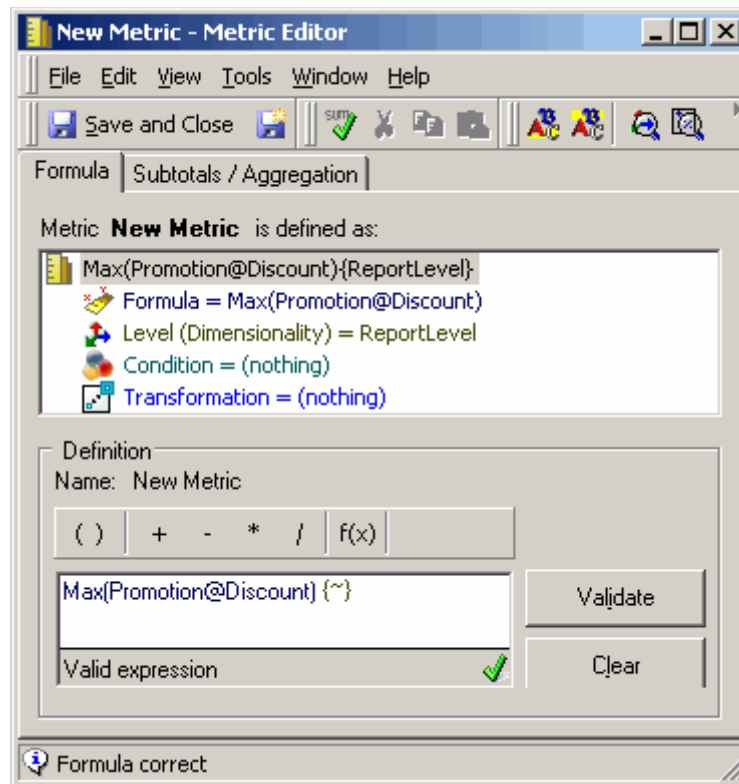
- This procedure assumes you are familiar with the process of creating a metric. For steps on how to create metrics, see [Chapter 2, Advanced Metrics](#).

To create metrics to use additional attribute forms as inputs for predictive metrics

- 1 Using the Metric Editor, create a new metric expression. All metric expressions must have an aggregation function. To support including attribute information in the metric expression, in the **Definition** area, type `Max ( )` to use the Max aggregation function.

- 2 Within the parentheses of the `Max()` aggregation function, specify the desired attribute form using the `AttributeName@FormName` format, where:
 - *AttributeName*: Is the name of the attribute. If there are spaces in the attribute name, you can enclose the attribute name in square brackets (`[]`).
 - *FormName*: Is the name of the attribute form. Be aware that this is different than the attribute form category. If there are spaces in the attribute form name, you can enclose the attribute form name in square brackets (`[]`).

For example, in the image shown below the Discount form of the Promotion attribute is included in the metric.



- 3 Add the attribute as a metric level so that this metric always returns results at the level of the attribute.
- 4 If the predictive metric is to be used to forecast values for elements that do not exist in your project, you must define the join type for the metric used as an input for the predictive metric to be an outer join. For example, the predictive metric is planned to forecast values for one year in the future. Since this future year is not represented in the project, you

must define the join type for the metric used as an input for the predictive metric to be an outer join so that values are returned.

To enable outer joins to include all data:

- a Select **Metric Join Type** from the **Tools** menu. The Metric Join Type dialog box opens.
 - b Clear the **Use default inherited value** check box.
 - c Select **Outer**.
 - d Click **OK** to close the dialog box.
- 5** If you plan to export predictive metric results to a third-party tool, you should define the column alias for the metric used as an input for the predictive metric. This ensures that the name of the metric used as an input for the predictive metric can be viewed when viewing the exported results in the third-party tool.

To create a metric column alias to ensure the column name matches the metric's name:

- a Select **Advanced Settings** from the **Tools** menu, and then select **Metric Column Options**. The Metric Column Alias Options dialog box opens.
 - b In the **Column Name** field, type the alias.
 - c Click **OK** to close the dialog box.
- 6** Save the metric, using the alias from the previous step as the metric name. You can now include the metric in a training metric to create a predictive metric, as described in [Creating a training metric with the Training Metric Wizard, page 808](#).

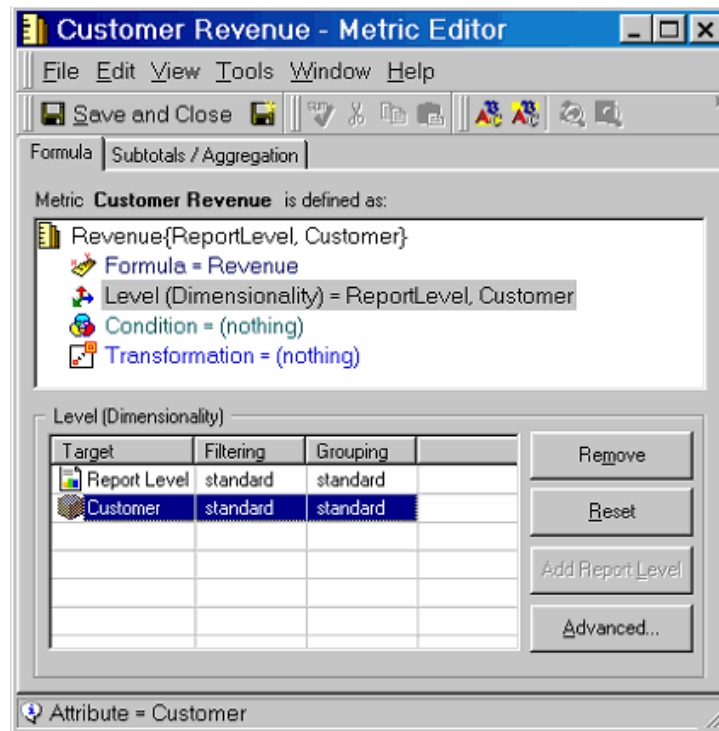
Level metrics as inputs for predictive metrics

The attribute used on the rows of the dataset report sets the level of the data by restricting the data to a particular level, or dimension, of the data model.

For example, if the Customer attribute is placed on the rows and the Revenue metric on the columns of a report, the data in the Revenue column is at the customer level. If the Revenue metric is used in the predictive model without any levels, then the data it produces changes based on the attribute of the report using the predictive metric. If Year is placed on the rows of the report described previously, the predictive metric calculates yearly revenue rather

than customer revenue. Passing yearly revenue to a predictive model based on customer revenue yields the wrong results.

This problem can be easily resolved by creating a separate metric, which is then used as an input for the predictive metric. This separate metric can be created to match the metric definition for Revenue, but also define its level as Customer. This approach is better than adding a level directly to the Revenue metric itself because the Revenue metric may be used in other situations where the level should not be set to Customer. Such a metric would look like the following.



Prerequisites

- This procedure assumes you are familiar with the process of creating a metric. For steps on how to create metrics, see [Chapter 2, Advanced Metrics](#).

To create level metrics to use as inputs for predictive metrics

- 1 In the Metric Editor, open the metric that requires a level.
- 2 Clear any Break-by parameters that may exist on the metric's function:
 - a Highlight the function in the **Definition** pane to select it.

- b Right-click the function and then select **Function_Name parameters**. The Parameters dialog box opens.
 - c On the **Break By** tab, click **Reset**.
 - d Click **OK** to close the dialog box.
- 3 Add the necessary attributes as metric levels:
- a Click **Level (Dimensionality)** on the Metric component pane.
 - b In the Object Browser, double-click each attribute to add as a level.
- 4 If the predictive metric is to be used to forecast values for elements that do not exist in your project, you must define the join type for the metric used as an input for the predictive metric to be an outer join. For example, the predictive metric is planned to forecast values for one year in the future. Since this future year is not represented in the project, you must define the outer join type for the metric used as an input for the predictive metric so that values are returned.

To enable outer joins to include all data:

- a Select **Metric Join Type** from the **Tools** menu. The Metric Join Type dialog box opens.
 - b Clear the **Use default inherited value** check box.
 - c Select **Outer**.
 - d Click **OK** to close the dialog box.
- 5 If you plan to export predictive metric results to a third-party tool, you should define the column alias for the metric used as an input for the predictive metric. This ensures that the name of the metric used as an input for the predictive metric can be viewed when viewing the exported results in the third-party tool.

To create a metric column alias to ensure the column name matches the metric's name:

- a Select **Advanced Settings** from the **Tools** menu, and then select **Metric Column Options**. The Metric Column Alias Options dialog box opens.
- b In the **Column Name** field, type the alias.
- c Click **OK** to close the dialog box.

- 6 Save the metric with the alias name from the previous step. You can now include the metric in a training metric to create a predictive metric, as described in [Creating a training metric with the Training Metric Wizard, page 808](#).

Conditional metrics as inputs for predictive metrics

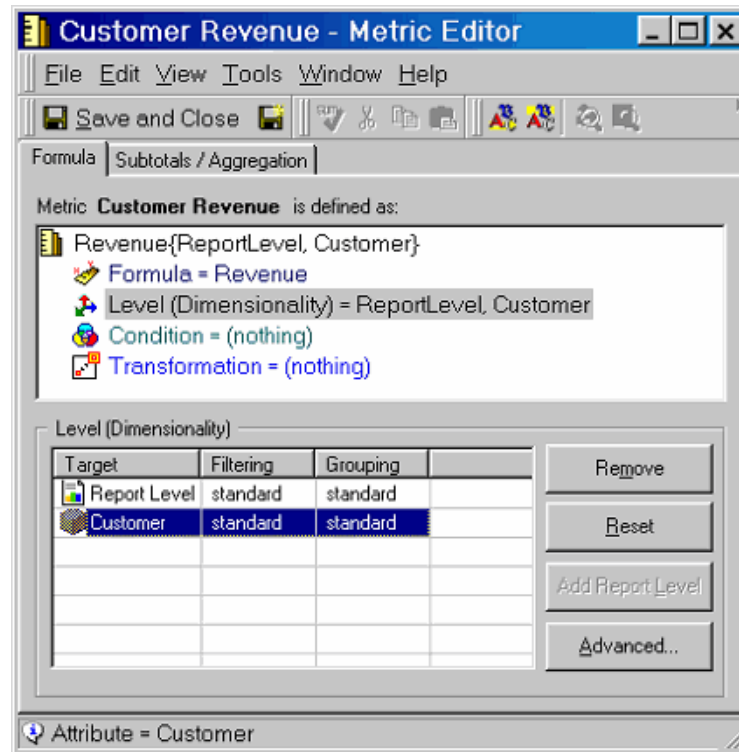
To group a metric's results by an attribute, create a conditional metric for each category. For example, you want to use customer revenue grouped by payment method in your data mining analysis. If you place the Customer attribute on the rows of the report, the Revenue metric on the columns, and the Payment Method attribute on the columns, you get the following report as a result:

Customer	Metrics	Revenue				
	Payment Method	Vis a	Amex	Check	MC	Club Card
Maxwell Aaronson		\$485	\$460	\$13	\$123	\$274
Hugh Abarca		\$142	\$639	\$979	\$832	\$163
Hazel Abelson		\$260	\$50	\$87	\$92	
Brooks Abern		\$739	\$29	\$104	\$63	\$993
Ross Abram		\$332	\$130	\$18	\$142	\$511
Wylie Abrams		\$107	\$25	\$28	\$524	\$16
Don Addison		\$403	\$121	\$1,129	\$459	\$117
Merrell Adess		\$510	\$769	\$153	\$141	\$107
Keith Adler		\$84	\$997	\$116	\$169	\$806
Daniel Aguilar		\$1,116	\$120	\$77	\$329	\$1,724
Deborah Aguirre		\$1,067	\$287	\$146	\$235	\$413
Dean Ahern		\$1,229	\$1,473	\$36	\$165	\$244
Elton Al-Timinu		\$487	\$132	\$355	\$63	\$101
Dorthy Alcaraz		\$64	\$266	\$129	\$73	\$27
Sarah Aldo		\$663	\$343	\$51	\$406	\$72
Peggy Alein		\$1,680	\$959	\$180	\$1,151	\$812
Maxwell Alexander		\$79	\$713	\$112	\$880	\$176

However, this report presents problems if it is used as a dataset report because multiple headings are generated for all the columns, specifically, Revenue and each Payment Method. Additionally, each column is revenue for a particular payment method and unless there is a metric that matches this definition, it is difficult to successfully deploy any model that uses one of these columns.

To solve this problem, create a separate metric, which is then used as an input for a predictive metric, that filters Revenue for each Payment Method.

This has the same definition as the original Revenue metric, but its conditionality is set to filter Revenue by a particular Payment Type.



Prerequisites

- This procedure assumes you are familiar with the process of creating metrics and filters. For steps on how to create metrics, see [Chapter 2, Advanced Metrics](#). For steps on how to create filters, see [Chapter 3, Advanced Filters](#).

To create a conditional predictive metric

- 1 Create a separate filter for each of the necessary attribute elements. For the example above, they are Payment Method = Visa, Payment Method = Amex, Payment Method = Check, and so on.
- 2 For each metric, create a separate metric to be used as an input for a predictive metric, as explained in the section above.
- 3 Add the filters you created as conditions of the metric-based predictive input metric. Save the metric. You can now include the metric in a training metric to create a predictive metric, as described in [Creating a training metric with the Training Metric Wizard, page 808](#).

The following report uses conditional metrics to generate the same results as the first report but in a dataset report format.

Customer	Metrics	Customer Revenue (Visa)	Customer Revenue (Amex)	Customer Revenue (Check)	Customer Revenue (MC)	Customer Revenue (Club Card)
Maxwell Aaronson		\$485	\$460	\$13	\$123	\$274
Hugh Abarca		\$142	\$639	\$979	\$832	\$163
Hazel Abelson		\$260	\$50	\$87	\$92	
Brooks Abern		\$739	\$29	\$104	\$63	\$993
Ross Abram		\$332	\$130	\$18	\$142	\$511
Wylie Abrams		\$107	\$25	\$28	\$524	\$16
Don Addison		\$403	\$121	\$1,129	\$459	\$117
Merrell Adess		\$510	\$769	\$153	\$141	\$107
Keith Adler		\$84	\$997	\$116	\$169	\$806
Daniel Aguilar		\$1,116	\$120	\$77	\$329	\$1,724
Deborah Aguirre		\$1,067	\$287	\$146	\$235	\$413
Dean Ahern		\$1,229	\$1,473	\$36	\$165	\$244
Elton Al-Timinu		\$487	\$132	\$355	\$63	\$101
Dorthy Alcaraz		\$64	\$266	\$129	\$73	\$27
Sarah Aldo		\$663	\$343	\$51	\$406	\$72
Peggy Alein		\$1,680	\$959	\$180	\$1,151	\$812
Maxwell Alexander		\$79	\$713	\$112	\$880	\$176

Using non-MicroStrategy dataset reports

While there are benefits to starting with a dataset report from MicroStrategy, there are cases where the dataset report used to create predictive models came from other sources. This could happen for a number of reasons, such as the data mining process may have started with data pulled from multiple systems. Alternatively, the data mining tool transformed or processed the data, or the data mining process was used to determine which data should be added to the database.

In any case, models that were developed from other data can still be used with MicroStrategy, as long as the following requirements are met:

- The data that forms the inputs to the data mining model must be present in MicroStrategy as metrics or attributes, which can then be used to create predictive metrics.
- The meaning of the data is unchanged when used in MicroStrategy. This includes any level requirements. For example, if the data mining model expects Recent Customer Transaction Count as input, the definitions for recent, customer, transaction, and count must be the same in

MicroStrategy. They must also yield the same results under the same conditions used when developing the model.

As you can see in [Importing the predictive model, page 823](#), if the model expects an input that is not already defined in MicroStrategy, the user is warned and then prompted to select a metric that matches. If one does not exist, the import process is aborted and that metric needs to be added to the project before the model can be imported.

Creating a predictive model

After you have created a dataset report, the next step is to create a predictive model. The predictive model will ultimately become a predictive metric within MicroStrategy metadata. You can create a predictive model in either of the following ways:

- [Using MicroStrategy, page 785](#)
- [Using third-party data mining applications, page 823](#)

Using MicroStrategy

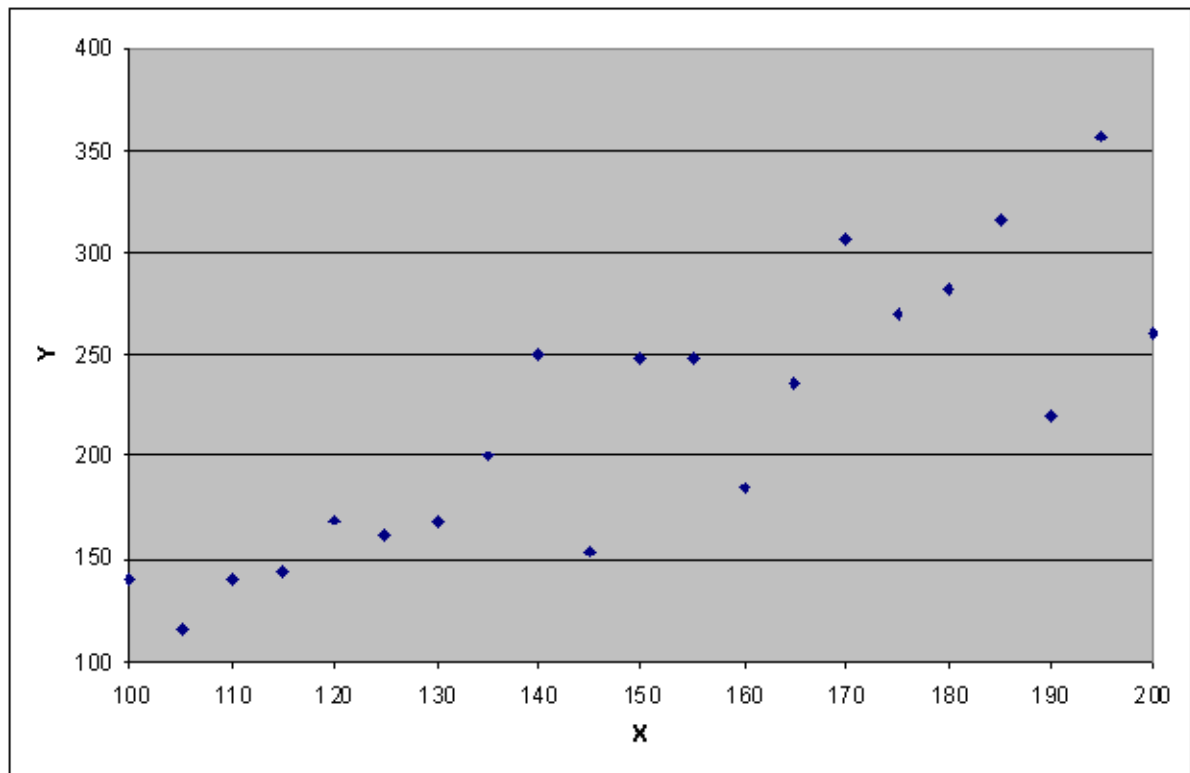
MicroStrategy Data Mining Services has been evolving to include more data mining algorithms and functionality. One key feature is MicroStrategy Developer's Training Metric Wizard. The Training Metric Wizard can be used to create several different types of predictive models including linear and exponential regression, logistic regression, decision tree, cluster, time series, and association rules.

Linear and exponential regression

The linear regression data mining technique should be familiar to you if you have ever tried to extrapolate or interpolate data, tried to find the line that best fits a series of data points, or used Microsoft Excel's LINEST or LOGEST functions.

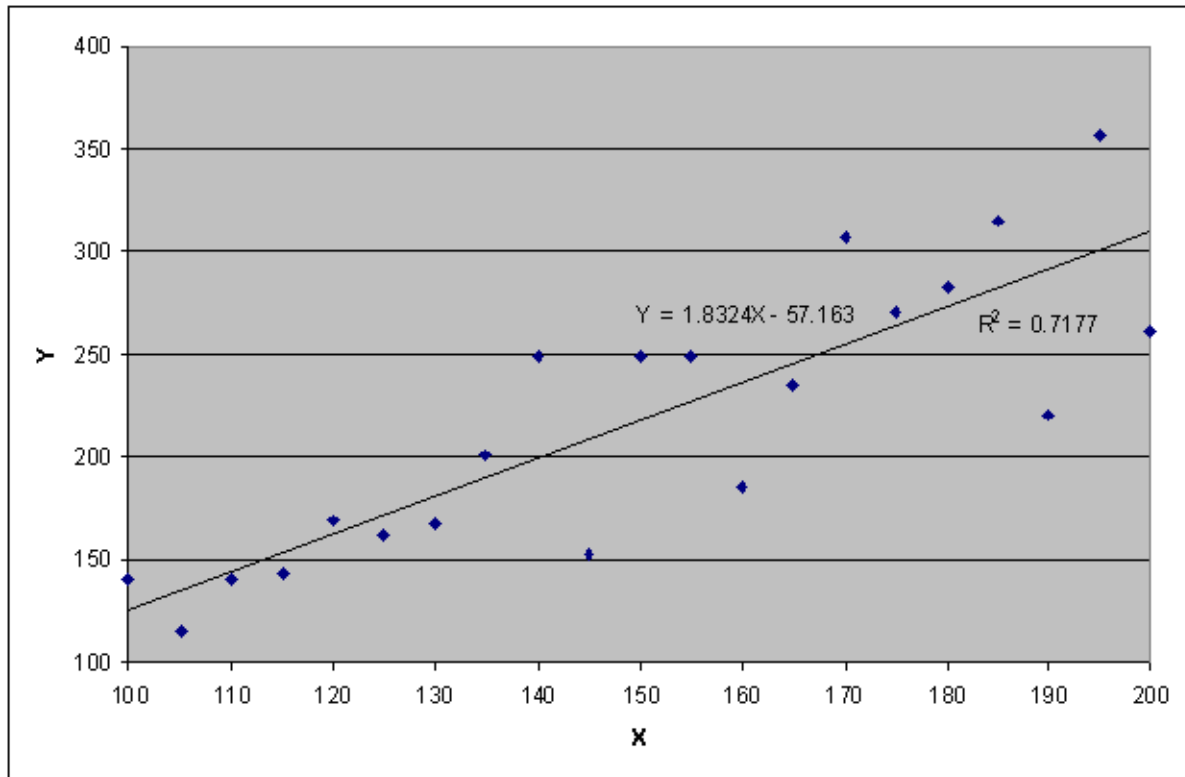
Regression analyzes the relationship between several predictive inputs, or independent variables, and a dependent variable that is to be predicted. Regression finds the line that best fits the data, with a minimum of error.

For example, you have a dataset report with just two variables, X and Y, which are plotted as in the following chart:

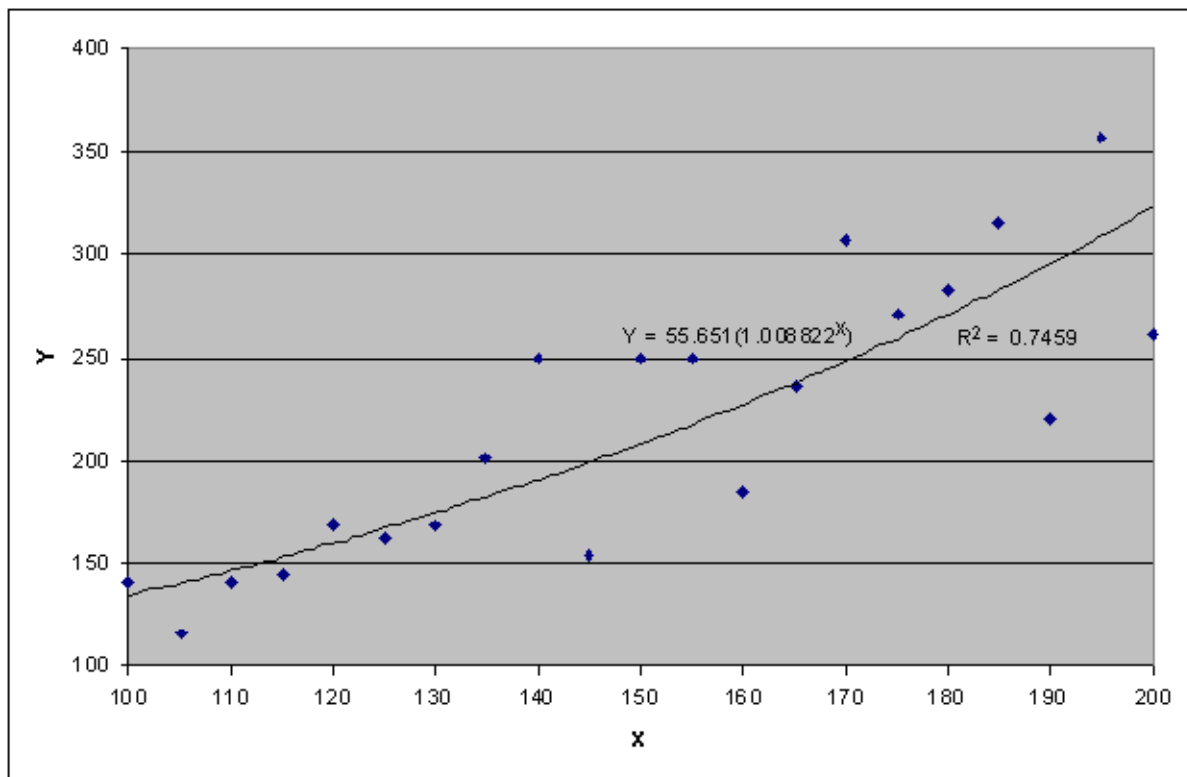


Using the regression technique, it is relatively simple to find the straight line that best fits this data, as shown below. The line is represented by a linear

equation in the classic $y = mx + b$ format, where m is the slope and b is the y-intercept.



Alternatively, you can also fit an exponential line through this data, as shown in the following chart. This line has an equation in the $y = b m^x$ format.

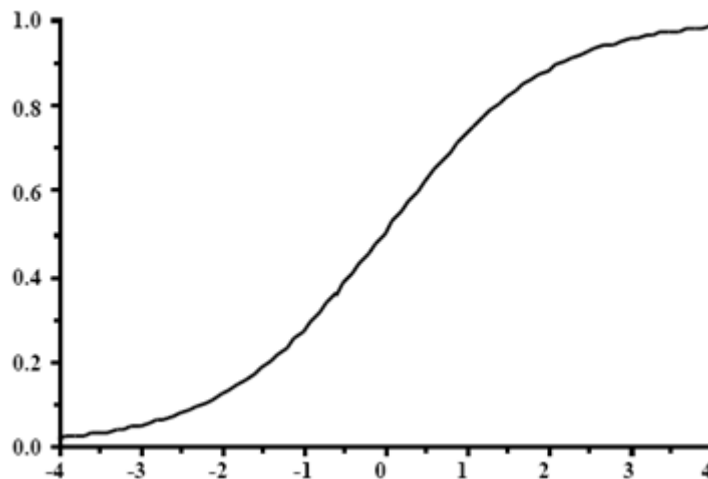


So, how can you tell which line has the better fit? Many statistics are used in the regression technique. One basic statistic is an indicator of the goodness-of-fit, meaning how well the line fits the relationship among the variables. This is also called the *Coefficient of Determination*, whose symbol is R^2 . The higher that R^2 is, the better the fit. The linear predictor has an R^2 of 0.7177 and the exponential predictor has an R^2 of 0.7459; therefore, the exponential predictor is a better fit statistically.

With just one independent variable, this example is considered a univariate regression model. In reality, the regression technique can work with any number of independent variables, but with only one dependent variable. While the multivariate regression models are not as easy to visualize as the univariate model, the technique does generate statistics so you can determine the goodness-of-fit.

Logistic regression

Logistic regression is a classification technique used to determine the most likely outcome from a set of finite values. The term logistic comes from the logit function, shown below:



Notice how this function tends to push values to zero or one; the more negative the x-axis value becomes, the logit function approaches a value of zero; the more positive the x-axis value becomes, it approaches a value of one. This is how regression, a technique originally used to calculate results across a wide range of values, can be used to calculate a finite number of possible outcomes.

The technique determines the probability that each outcome will occur. A regression equation is created for each outcome and comparisons are made to select a prediction based on the outcome with the highest likelihood.

Evaluating logistic regression using a confusion matrix

As with other categorical predictors, the success of a logistic regression model can be evaluated using a confusion matrix. The confusion matrix highlights the instances of true positives, true negatives, false positives, and false negatives. For example, a model is used to predict the risk of heart disease as Low, Medium, or High. The confusion matrix would be generated as follows:

Confusion Matrix	Predicted Result		
	High	Medium	Low
Actual Result			
High	320 (32.4%)	24 (2.4%)	9 (0.9%)
Medium	3 (0.3%)	117 (11.9%)	5 (0.5%)
Low	0 (0.0%)	0 (0.0%)	509 (51.6%)

Proper analysis of the matrix depends on the predictive situation. In the scenario concerning heart disease risk, outcomes with higher false positives for Medium and High Risk are favored over increased false negatives of High Risk. A false positive in this case encourages preventative measures, whereas a false negative implies good health when, in fact, significant concern exists.

Cluster analysis

Cluster analysis offers a method of grouping data values based on similarities within the data. This technique segments different items into groups depending on the degree of association between items. The degree of association between two objects is maximal if they belong to the same group and minimal otherwise. A specified or determined number of groups, or clusters, is formed, allowing each data value to then be mathematically categorized into the appropriate cluster.

Cluster analysis is seen as an unguided learning technique since there is no target or dependent variable. There are usually underlying features that determine why certain things appear related and others unrelated. Analyzing clusters of related elements can yield meaningful insight into how various elements of a dataset report relate to each other.

MicroStrategy employs the k-Means algorithm for determining clusters. Using this technique, clusters are defined by a center in multidimensional space. The dimensions of this space are determined by the independent variables that characterize each item. Continuous variables are normalized to the range of zero to one (so that no variable dominates). Categorical variables are replaced with a binary indicator variable for each category (1 = an item is in that category, 0 = an item is not). In this fashion, each variable spans a similar range and represents a dimension in this space.

The user specifies the number of clusters, k , and the algorithm determines the coordinates of the center of each cluster.



In MicroStrategy, the number of clusters to be generated from a set of data can be specified by the user or determined by the software. If determined by the software, the number of clusters is based on multiple iterations to reveal the optimal grouping. A maximum is set by the user to limit the number of clusters determined by the software.

The table below shows the result of a model that determines which of five clusters a customer belongs to.

Field Name	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Age Range=24 and under	0.06	0.06	0.08	0.05	0.07
Age Range=25 to 34	0.16	0.18	0.17	0.16	0.18
Age Range=35 to 44	0.33	0.34	0.37	0.38	0.38
Age Range=45 to 54	0.18	0.16	0.16	0.15	0.14
Age Range=55 and over	0.27	0.27	0.23	0.26	0.23
Education=Graduate	0.22	0.20	0.22	0.24	0.24
Education=High School	0.40	0.41	0.41	0.38	0.40
Education=Other	0.08	0.09	0.11	0.10	0.10
Education=Undergraduate	0.30	0.30	0.27	0.28	0.26
Gender=Female	0.46	0.40	0.46	0.48	0.43
Housing Type=Dependent	0.00	1.00	0.00	0.00	0.00
Housing Type=Owner	0.00	0.00	0.00	1.00	1.00
Housing Type=Renter	1.00	0.00	1.00	0.00	0.00
Marital Status=Married	0.00	0.33	0.69	0.71	0.00
Marital Status=Previously Married	0.00	0.18	0.31	0.29	0.00
Marital Status=Single	1.00	0.49	0.00	0.00	1.00

The center of each cluster is represented in the five cluster columns. With this information, it is possible to learn what the model says about each cluster. For example, since the center of Cluster 1 is where “Marital Status = Single” is one and all other marital statuses are zero, single people tend to be in this cluster. On the other hand, there is not a lot of variability across the clusters for the Age Range dimensions, so age is not a significant factor for segmentation.

Creating a cluster model is the first step, because while the algorithm will always generate the proper clusters, it is up to the user to understand how each cluster differs from the others.

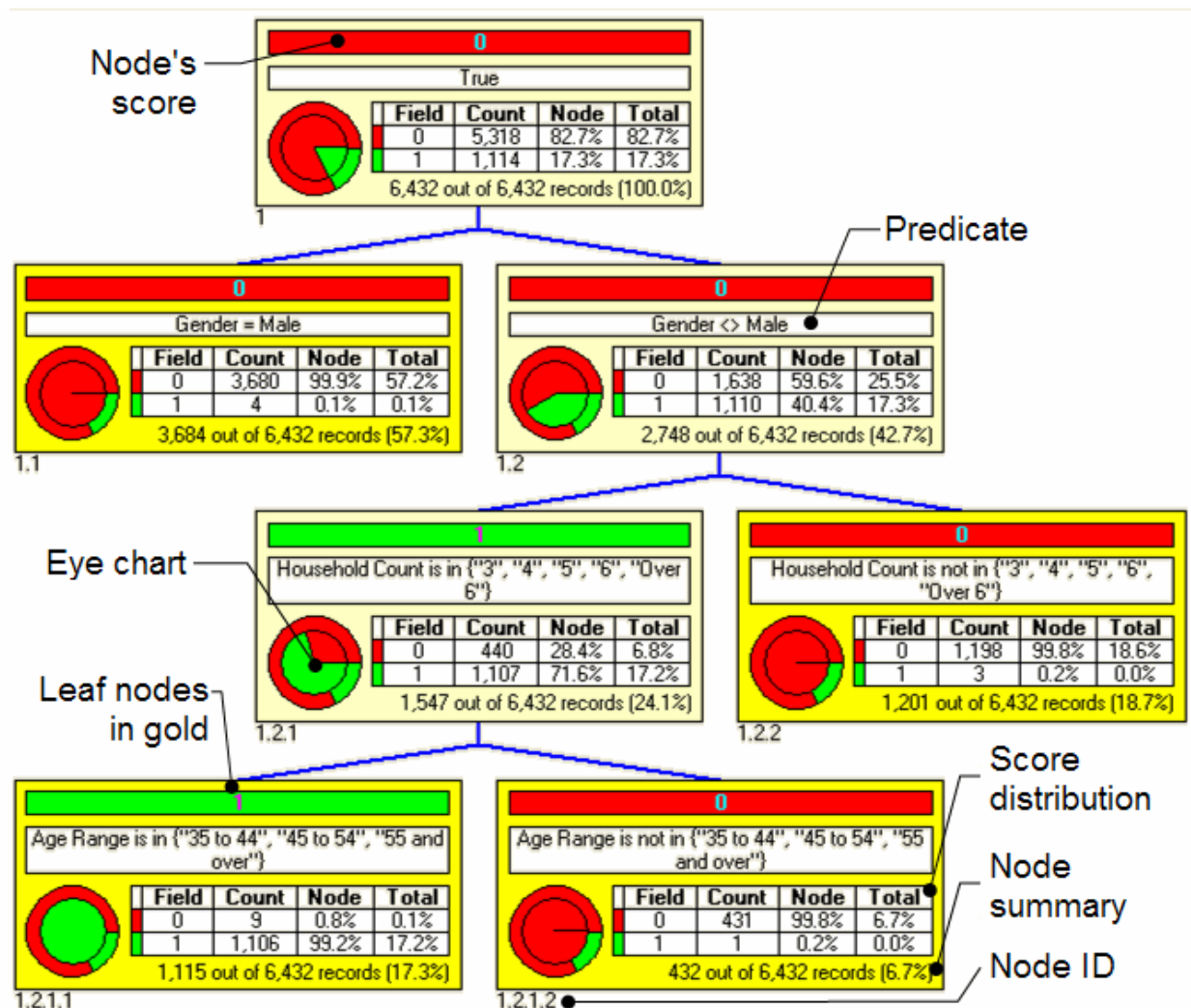
Decision tree analysis

Decision trees are one of the most intuitive types of data mining model since they display an “if-then-else” structure that even the most novice user can understand. A decision tree is simple to follow and allows for very sophisticated analysis.

Decision trees separate a given dataset report into distinct subsets. Rather than employing the unguided learning approach used in cluster analysis (see [Cluster analysis, page 790](#)), decision tree analysis uses a guided algorithm to create these subsets. The subsets share a particular target characteristic, which is provided by the dependent variable. Independent variables provide the other characteristics, which are used to divide the original set of data into

subsets. Typically, the independent variable with the most predictive power is used first, then the next most powerful, and so on. MicroStrategy implements the Classification And Regression Tree (CART) algorithm to construct decision trees.

The image below displays a basic decision tree that includes its root node at the top. The connected nodes of the decision tree are traversed from top-to-bottom primarily and then left-to-right. Processing typically ends at the first leaf node (a leaf node is a node with no children) encountered with a predicate that evaluates to True.



Each node includes the following information:

- **Score:** The most common (or dominant) result of the data records for the node.

- **Predicate:** A logical statement that is used to separate data records from a node's parent. Data records can belong to a node if the predicate evaluates to True. Predicates can be a single logic statement or a combination of multiple logic statements using operators such as AND, OR, XOR, and so on.
- **Eye chart:** A graphical representation of the distribution of scores, which is only shown if the PMML contains score distribution information. The outer ring chart shows the distribution of scores for all the data records in the training dataset report. The inner pie chart shows the distribution of scores for the data records in this node. The largest slice of the inner pie chart is the score for this node.
- **Score distribution:** A table showing the breakdown of training data records associated with the node, which serves also as the legend for the eye chart. The score distribution contains the actual count of training data records in this node, represented by each target class. The proportion of each class of data records is displayed as a percentage of the total counts for this node as well as a percentage of the total population. The node percentage for the dominant score can be considered as the confidence in the node's score. Score distributions are not required by PMML and, if not present, this score distribution table cannot be shown.
- **Node summary:** Percentage of all the training data records associated with this node. This information is descriptive only (not used to predict results directly) and can only be shown if the PMML contains score distribution information.
- **Node ID:** A reference for the node. MicroStrategy uses a level-depth format where the ID is a series of numbers representing the left-based position for each level.

The PMML specification for decision trees includes strategies for missing value handling, no true child handling, weighted confidences, and other features. For more details on these strategies, see the Data Mining Group website at <http://www.dmg.org>.

Association rules analysis

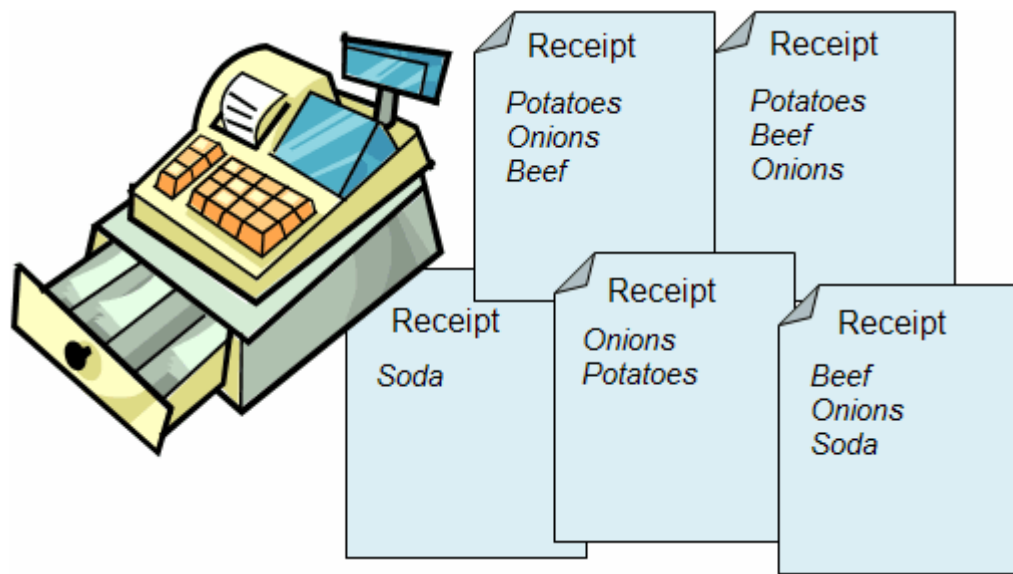
Association rules look for relationships between items. The most common example of this is market basket analysis.

Market basket analysis studies retail purchases to determine which items tend to appear together in individual transactions. Retailers use market basket analysis for their commercial websites to suggest additional items to purchase before a customer completes their order. These recommendations

are based on what other items are typically purchased together with the items already in the customer's order. Market basket analysis provides the valuable ability to upsell a customer at the time of purchase, which has become a key requirement for any retailer.

The key to this type of analysis is the ability to find associations amongst the items in each transaction. This can include associations such as which items appear together the most frequently, and which items tend to increase the likelihood that other items will appear in the same transaction.

For example, five transactions from a grocery store are shown in the image below.



These transactions are summarized in the table below:

☀ A 1 denotes that the item is included in the transaction, while a 0 denotes that the item is not included in the transaction.

Transaction ID	Soda	Potatoes	Onions	Beef
1	1	0	0	0
2	0	1	1	1
3	0	1	1	0
4	1	0	1	1
5	0	1	1	1

By reviewing the transaction table above, you can determine that beef appears in three out of the five transactions. In other words, 60% of all transactions support the item beef.

Support is a key concept that describes the relative frequency of transactions that contain an item or set of items, called an itemset. Itemset is another key concept in association rules since you can calculate associations not only for individual items but also between groups of items.

The table below shows the support for all possible combinations of one, two or three items per itemset (in other words, a maximum of three items per itemset).

Itemset	Transaction Count	Support
Beef	3	60%
Onions	4	80%
Potatoes	3	60%
Soda	2	40%
Beef, Onions	3	60%
Beef, Potatoes	2	40%
Onions, Potatoes	3	60%
Beef, Soda	1	20%
Onions, Soda	1	20%
Potatoes, Soda	0	0%
Beef, Onions, Potatoes	2	40%
Beef, Onions, Soda	1	20%
Beef, Potatoes, Soda	0	0%
Onions, Potatoes, Soda	0	0%

With this information, you can define rules that describe the associations between items or itemsets. The rules can be described as: The antecedent itemset implies the consequent itemset. In other words, the antecedent is a combination of items that are analyzed to determine what other items might be associated with this combination. These implied items are the consequent of the analysis.

For example, consider the rule Potatoes and Onions itemset implies Beef. This rule describes how transactions containing both potatoes and onions are related to those transactions that also contain beef. An association rule

contains this qualitative statement but it is also quantified with additional statistics.

From the transaction table, you can determine that three out of five transactions contain the itemset potatoes and onions (antecedent), and two out of five transactions contain both the antecedent and the consequent. In other words, two out of three of the transactions containing potatoes and onions also contain beef.

Each association rule contains a confidence statistic that estimates the probability of a transaction having the consequent given the antecedent. In this example, after analyzing the five transactions, the confidence in this rule Potatoes and Onions itemset implies Beef is 67%. Confidence is calculated as follows:

$$\text{Confidence} = \frac{\text{Support}(A + C)}{\text{Support}(A)} = \frac{\text{Support}(\text{Potatoes, Onions, Beef})}{\text{Support}(\text{Potatoes, Onion})} = \frac{40\%}{60\%} = 67\%$$

Therefore, if a customer purchases both potatoes and onions, you can be 67% confident that beef would also be purchased, based on the five transactions analyzed.

By analyzing all combinations of these itemsets, an association rules model contains rules that describe the relationships between itemsets in a given set of transactions. The table below shows the confidence in all the rules found in this example scenario.

	Consequent			
Antecedent	Beef	Onions	Potatoes	Soda
Beef		100%	67%	33%
Onions	75%		75%	25%
Potatoes	67%	100%		0%
Soda	50%	50%	0%	
Beef, Onions			67%	33%
Beef, Potatoes				0%
Onions, Potatoes	67%			0%
Beef, Soda				
Onions, Soda	100%			

This model contains 22 different rules based on only four different items and five transactions. Since a typical retailer can have thousands of items and

potentially millions of transactions, this type of analysis can generate a large number of rules. It is typical that the vast majority of these rules will have a low confidence (notice in the table above that the lowest confidences are rules that have a consequent containing soda).

In order to limit the volume of rules generated, you can control:

- **Maximum number of items per antecedent:** This setting defines the maximum number of items that can be included in each antecedent itemset (consequent itemsets can only contain a single item). For example, if set to three, items for each antecedent will be grouped into itemsets containing one, two, or three items. In the example above that includes the four items beef, onions, potatoes, and soda, a maximum of two creates antecedents with no more than two items, while still including each item in the analysis.
- **Minimum confidence:** The minimum probability that qualifying rules should have. For example, if set to 10%, then an association rule must have a confidence of 10% or more to appear in the model.
- **Minimum support:** The minimum number of transactions an itemset must occur in to be considered for an association rule. For example, if set to 1%, then itemsets must appear, on average, in one transaction out of 100.
- **Maximum consequent support:** The maximum support of the consequent allowed for qualifying rules. This can be used to avoid including obvious recommendations in the resulting rules. For example, if set to 99%, then rules that have a consequent support greater than 99% are not included in the resulting model.

In addition to support and confidence, association rules can also include the following statistics:

- **Lift:** Lift is a ratio that describes whether the rule is more or less significant than what one would expect from random chance. Lift values greater than 1.0 indicate that transactions containing the antecedent tend to contain the consequent more often than transactions that do not contain the antecedent. The table below shows the lift of the rules in our

example model. Note that onions are an above average predictor of the purchase of beef and potatoes.

	Consequent			
Antecedent	Beef	Onions	Potatoes	Soda
Beef		1.25	1.11	0.83
Onions	1.25		1.25	0.63
Potatoes	1.11	1.25		0.00
Soda	0.83	0.63	0.00	
Beef, Onions			1.11	0.83
Beef, Potatoes				0.00
Onions, Potatoes	1.11			0.00
Beef, Soda				
Onions, Soda	1.67			

- Leverage:** Leverage is a value that describes the support of the combination of the antecedent and the consequent as compared to their individual support. Leverage can range from -0.25 to 0.25, and a high leverage indicates that there is a relationship between the antecedent and the consequent. For example, if 50% of the transactions contain the antecedent and 50% of the transactions contain the consequent, you would expect 25% of the transactions to contain both the antecedent and the consequent if they were completely independent; this would correspond to a leverage of zero. If more than 25% of the transactions contain the antecedent and consequent together, then there is a positive leverage (between 0 and 0.25). This positive leverage indicates that the antecedent and consequent appear more frequently than you would expect if they were completely independent, and can hint at a relationship.
- Affinity:** Affinity is a measure of the similarity between the antecedent and consequent itemsets, which is referred to as the Jaccard Similarity in statistical analysis. Affinity can range from 0 to 1, with itemsets that are similar approaching the value of 1.

Time series analysis

Time series represents a broad and diverse set of analysis techniques which use a sequence of measurements to make forecasts based on the intrinsic

nature of that data. While most other data mining techniques search for independent variables that have predictive power with respect to a particular dependent variable, time series analysis has just one variable. The past behavior of the target variable is used to predict its future behavior.

This past behavior is measured in a time-based sequence of data, and most often that sequence is a set of measurements taken at equal intervals in time. By analyzing how values change over time, time series analysis attempts to find a model that best fits the data.

There is also an implied assumption that the data has an internal structure that can be measured, such as level, trend, and seasonality.

Time series forecasts are only statistically valid for projections that are just one time period beyond the last known data point. The techniques used to determine the model parameters are focused on reducing the one-step-ahead error. Forecasts two or more steps ahead are less reliable since they go beyond the forecast horizon considered when the model was developed.

In time series analysis, many models are tested and each model is structured to match a certain data profile.

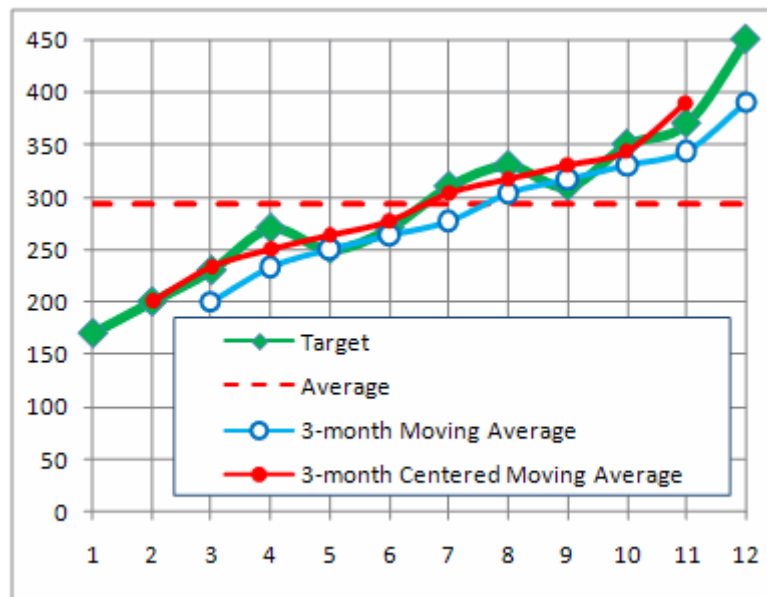
For example, consider this set of data:

Month	Target	Average	Three-Month Moving Average	Three-Month Centered Moving Average
1	170	293		
2	200	293		200
3	230	293	200	233
4	270	293	233	250
5	250	293	250	263
6	270	293	263	277
7	310	293	277	303
8	330	293	303	317
9	310	293	317	330
10	350	293	330	343
11	370	293	343	390
12	450	293	390	

The first two columns contain the data to be modeled. The target, which could be anything from sales revenue to units sold can be used to create a few simple models:

- The Average column is simply the average of all twelve data points.
- Each data point in the Three-Month Moving Average is the average of the previous three months target data. Since this model contains a series of subset averages, it does a better job of following the data than the simple average. But since it averages the past three months, it tends to lag behind the target data. The last value is at time t but it can never catch the upward trend of the target.
- The right most column is a centered version of the Three-Month Moving Average. Each data point in this column is an average of the prior, current, and next month. This is an improvement over the non-centered version since the lag has been eliminated, but at the cost of delaying the forecast. The last value is at time $t-1$ but it tracks the upward trend closely.

The differences between the three models becomes clear when the data is plotted on the chart shown below.



The Average model simply captures the level of the data. If our Target data was flat without a significant upward or downward trend, a simple average might make a sufficient model. But in this case, with the data trending strongly upward, it is not a good fit.

Both moving average models do a better job. The centered moving average avoids the lag problem by delaying its calculation and therefore is the best fit.



The technical definition of best fit is the model with the lowest Root Mean Square Error (RMSE). The RMSE is calculated by taking the square root of the average difference between the actual data and the forecast at each time period.

In all three models, the averages have the effect of smoothing the data by diminishing the peaks and valleys. The simple 12-month average is a constant and therefore completely smooth. On the other hand, the moving three-month averages still have some ups and downs, but not as strongly as the original target data. The challenge for finding the best model is determining the proper time horizon to forecast over. This is especially true when the data is not stationary but has a significant trend, since older values can hide more recent changes.

A common solution is a technique called exponential smoothing. In order to understand how exponential smoothing works, it is helpful to see how the technique is derived from moving averages. The generic formula for calculating averages is:

$$\text{Average} = \frac{1}{n} \sum_{i=1}^n y_i = \left(\frac{1}{n}\right)y_1 + \left(\frac{1}{n}\right)y_2 + \dots + \left(\frac{1}{n}\right)y_n$$

One way of thinking of an average is that it is the sum of each value weighted by $1/n$. Our simple 12-month average gave equal weight (one twelfth) to each month. The three-month moving averages gave each month a weight of one-third.

Exponential smoothing gives more recent values greater influence on the result. Older values are given exponentially decreasing weight. The generic exponential smoothing formula is:

$$S_t = \alpha y_{t-1} + (1 - \alpha)S_{t-1}$$

Where:

- S_t = Smoothed Observation
- α = A smoothing constant, determines how quickly or slowly weights decrease as observations get older



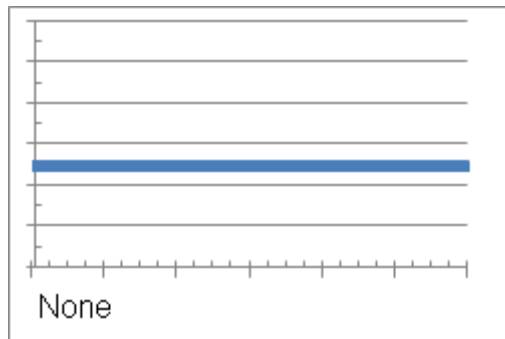
Over many observations, the smoothing effect follows an infinite series that approximates an exponential function.

In exponential smoothing, the algorithm determines the smoothing constant α that results in the lowest RMSE for a given time series profile. The state of

the art approach to exponential smoothing, described by Everette S. Gardner, attempts to find the best profile that matches the data used to train the model. In particular, each profile can be described by two aspects, trend and seasonality.

Trend is the dominant direction that the data is heading towards, as a function of time. Exponential Smoothing includes the following trend types:

- **None:** This is the simplest case, in which there is no trend. This means that the data is not dominated by either an upward or a downward progression, as shown in the example below:

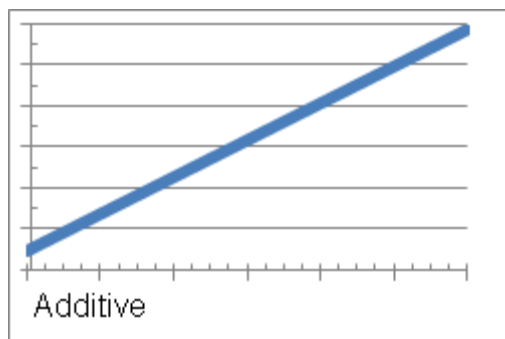


The following equation describes this trend line:

$$y_t = b_0$$

Where y_t is the value being forecast, and b_0 is a constant representing the level of the data.

- **Additive:** An additive trend is evident when the time series data changes an equal amount per time period, as shown in the example below:



The following equation describes this trend line:

$$y_t = b_0 + b_1 t$$

Where y_t is the value being forecast, b_0 is the level, b_1 is slope or constant trend, and t is the number of periods ahead of the last known data point in the time series.

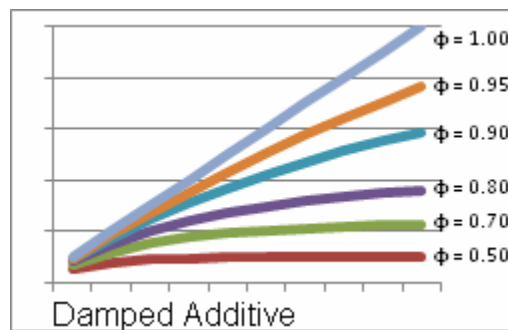
- **Damped additive:** With this trend, the amount of change decreases each successive time period. This can help to reflect most real world systems and processes, in which a trend is constrained from progressing indefinitely without change.

The rate of this damping is governed by the damping parameter ϕ (phi). The value of phi can vary from zero to one, with the damping effect becoming smaller as phi approaches one.



The additive trend is the same as a damped additive trend with $\phi = 1$.

The value of phi is the one that results in the lowest RMSE. This trend is shown in the following example:



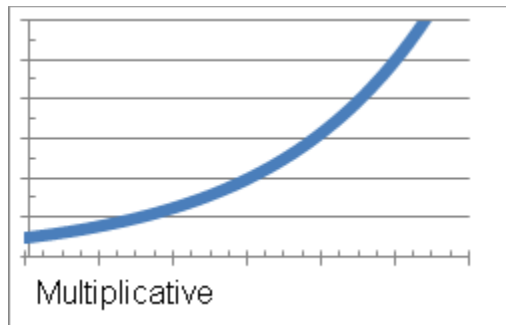
The following equation describes this trend line:

$$y_t = b_0 + b_1 \text{Damping}(t, \phi)$$

Where y_t is the value being forecast, b_0 is the level, b_1 is slope or constant trend, and t is the number of periods ahead of the last known data point in the time series. The damping is calculated using the following formula:

$$\text{Damping}(t, \phi) = \sum_{k=1}^t \phi^k$$

- **Multiplicative:** A multiplicative trend is one where the trend is not constant, but the trend changes at a fixed rate, as shown in the example below:

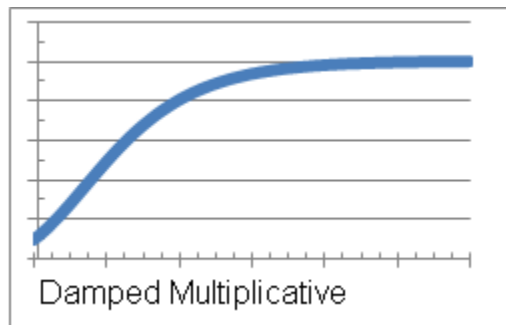


The following equation describes this trend line:

$$y_t = b_0(b_1)^t$$

Where y_t is the value being forecast, b_0 is the level, b_1 is slope or constant trend, and t is the number of periods ahead of the last known data point in the time series.

- **Damped multiplicative:** This trend is one where the rate of change in the trend decreases over time, subject to the damping parameter ϕ , as shown in the example below:



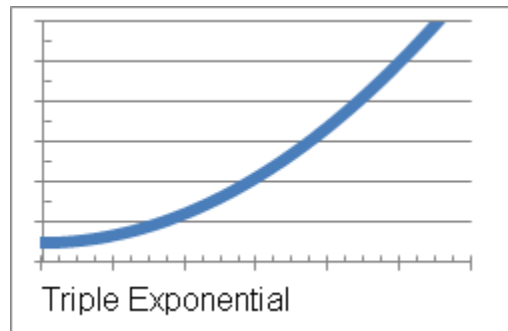
The following equation describes this trend line:

$$y_t = b_0(b_1)^{\text{Damping}(t, \phi)}$$

Where y_t is the value being forecast, b_0 is the level, b_1 is the rate at which the undamped trend changes, and t is the number of periods ahead of the last known data point in the time series. The damping is calculated using the following formula:

$$\text{Damping}(t, \phi) = \sum_{k=1}^t \phi^k$$

- **Triple exponential:** This trend is evident when the time series data follows a parabolic profile, as shown in the example below:



The following equation describes this trend line:

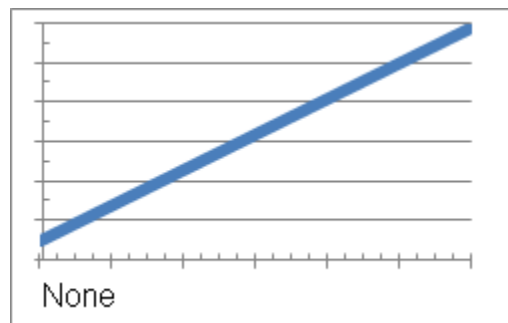
$$y_t = b_0 + b_1t + b_2t^2$$

Where y_t is the value being forecast, b_0 is the level, b_1 is the slope or constant trend, b_2 is the rate of acceleration of the trend, and t is the number of periods ahead of the last known data point in the time series.

The other aspect that describes the time series profile is seasonality. Seasonality accounts for seasonal differences that appear in the time series data. When creating models with MicroStrategy using the Training Metric Wizard, the user defines the number of time periods in the seasonal cycle. For example, with monthly data you could define each month as its own season by using 12 as the value for the seasonality. Similarly, with quarterly data you could define each quarter as its own season by using four as the value for the seasonality.

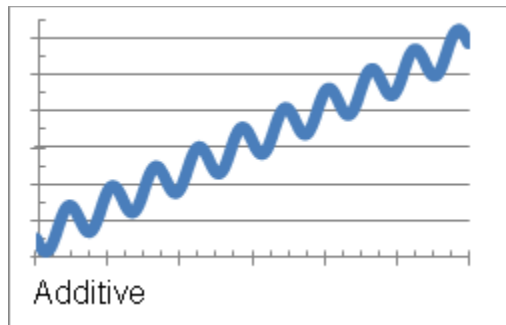
Exponential Smoothing includes the following seasonality types:

- **None:** There are no seasonal factors in the time series, as shown in the example below:



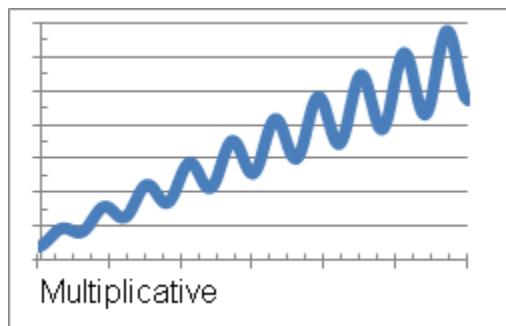
To use this type of seasonality, in the Training Metric Wizard, specify zero as the number of time periods for the seasonal cycle.

- **Additive:** Seasonal factors are added to the trend result. This means that each season's effect on the profile is constant, as shown in the example below:



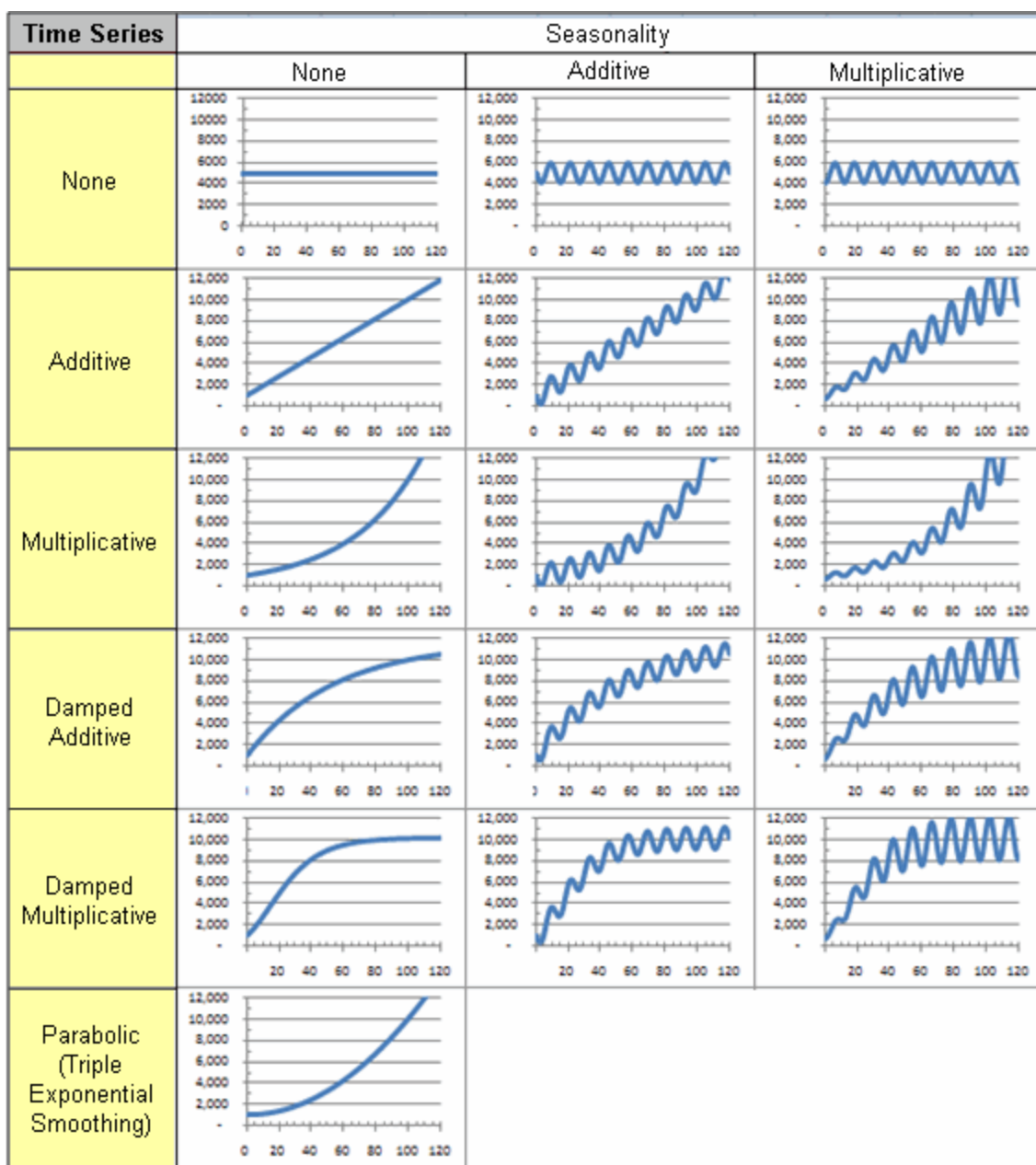
This profile is considered only when the number of time periods in the seasonal cycle, specified in the Training Metric Wizard, is two or more.

- **Multiplicative:** Seasonal factors are multiplied by the trend result. This means that seasonal effects increase as the trend of the data increases, as shown in the example below:



This profile is considered only when the number of time periods in the seasonal cycle, specified in the Training Metric Wizard, is two or more.

By combining trend and seasonality, the following are all of the possible time series profiles:



Predictive model workflow

To create a predictive model, you must do the following, in order:

- 1 Create the training metrics using the Training Metric Wizard. For information, see [Creating a training metric with the Training Metric Wizard, page 808](#).
- 2 Create a training report containing the training metric. This training report will generate the dataset used to train the predictive models.
- 3 Create the predictive metric from the training metric by executing the training report. For information, see [Create predictive metrics from training metrics, page 820](#).
 - Since Exponential Smoothing gives greater weight to more recent data, time series models should be updated whenever new data is available. While this can be said for other types of model as well, this is particularly true for time series models since recent results can significantly influence the model. Therefore, for time series models, it is common to deploy the training metrics instead of the predictive metrics. This guarantees the time series forecast results reflect the most recent data.
 - If the option to automatically create predictive metrics is selected when the training metric is created, simply execute the training report. The predictive metric is generated according to the parameters of the training metric. You can now use the predictive metric on reports or other MicroStrategy objects. For information, see [Using the predictive metric, page 829](#).
 - If the option to automatically create predictive metrics is not selected when the training metric is created, execute the training report and then use the Create Predictive Metrics option. For steps, see [To use the Create Predictive Metrics option in the Report Editor, page 822](#). You can then use the predictive metric, as discussed in [Using the predictive metric, page 829](#).

Creating a training metric with the Training Metric Wizard

Recall that the Data Mining Services workflow develops a predictive model based on a dataset report. When you use MicroStrategy, this dataset report is

generated in the form of a MicroStrategy report or MDX cube, with the basic format shown in the following report sample:

Quarter	Metrics	QuarterIndex	Revenue
2004 Q1		1	\$3,856,799
2004 Q2		2	\$4,243,328
2004 Q3		3	\$4,930,369
2004 Q4		4	\$5,443,579
2005 Q1		5	\$5,164,830
2005 Q2		6	\$5,014,413
2005 Q3		7	\$5,713,240
2005 Q4		8	\$6,509,331
2006 Q1		9	\$6,271,951
2006 Q2		10	\$6,657,190
2006 Q3		11	\$6,979,821
2006 Q4		12	\$7,581,619

The dataset report shown above can be used to develop a predictive model to forecast on-line sales per quarter. The process of developing this predictive model is called *training the model*. Data Mining Services uses a particular type of metric called a training metric to develop the predictive model. When the training metric is added to the dataset report, it analyzes the data and generates a forecast, as shown below.

Quarter	Revenue	Training metric for 'Revenue Predictor (Linear)'
2004 Q1	\$3,856,799	\$4,032,525
2004 Q2	\$4,243,328	\$4,335,194
2004 Q3	\$4,930,369	\$4,637,864
2004 Q4	\$5,443,579	\$4,940,533
2005 Q1	\$5,164,830	\$5,243,202
2005 Q2	\$5,014,413	\$5,545,871
2005 Q3	\$5,713,240	\$5,848,540
2005 Q4	\$6,509,331	\$6,151,210
2006 Q1	\$6,271,951	\$6,453,879
2006 Q2	\$6,657,190	\$6,756,548
2006 Q3	\$6,979,821	\$7,059,217
2006 Q4	\$7,581,619	\$7,361,886
2007 Q1		\$7,664,555
2007 Q2		\$7,967,225
2007 Q3		\$8,269,894
2007 Q4		\$8,572,563

This training metric uses a powerful function that forecasts the results shown in the previous report, given the proper parameters. It also generates a predictive model that can be used to forecast future outcomes. The MicroStrategy Training Metric Wizard allows you to easily create training metrics.

Prerequisites

- You must have Architect privileges to access the Training Metric Wizard.
- If you are creating a derived training metric, you must also have a license for MicroStrategy OLAP Services.
- If you are creating a training metric within an MDX cube, you must have imported an MDX cube into your MicroStrategy project. For steps to import an MDX cube as well as additional requirements to import MDX cubes into MicroStrategy, see the *MDX Cube Reporting Guide*.
- If you are creating a derived training metric on a document, you must add a dataset report as a Grid/Graph on the document.

To create a training metric

- 1 In MicroStrategy Developer, you can create the following types of training metrics:
 - Stand-alone training metric: Stand-alone training metrics exist in metadata and are the standard type of training metric that can be used on any report. You can use metrics and attributes within the project as inputs to create stand-alone training metrics.

To create a stand-alone training metric, from the **Tools** menu, select **Training Metric Wizard**.
 - Derived training metric: Derived training metrics are training metrics that are created using the derived metrics feature. You must have MicroStrategy OLAP Services to create derived metrics. Derived training metrics also have the same requirements and restrictions as derived metrics (see the *In-memory Analytics Guide*). This capability is particularly useful during the exploratory phase of the data mining process, during which many different variables are tried in a variety of modeling configurations.

Derived training metrics can be created directly on reports as well as reports that are added as Grid/Graphs on documents:

- For standard reports, navigate to a report, right-click the report, and select **Run**. The Report Editor opens. From the **Insert** menu, select **New Training Metric**.
- For reports that are added as Grid/Graphs on documents, right-click the Grid/Graph within the document and select **Edit Grid**. From the **Insert** menu, select **New Training Metric**.
- MDX cube training metrics: If you integrate MDX cube sources into MicroStrategy as MDX cubes, you can create training metrics based off of the MDX cube data. You can use metrics and attributes within the MDX cube as inputs to create MDX cube training metrics.

Using either the MDX Cube Catalog or the MDX Cube Editor to view an imported MDX cube, from the **Edit** menu, select **Training Metric Wizard**.

The Introduction page of the Training Metric Wizard opens. To skip the Introduction page when creating training metrics in the future, select the **Don't show this message next time** check box.

2 Click **Next** to open the Select Type of Analysis page.

3 Select a type of analysis from the following:

- **Linear regression:** The function attempts to calculate the coefficients of a straight line that best fits the input data. The calculated formula has the following format:

$$y = b_0 + (b_1 \times X_1) + (b_2 \times X_2) + \dots + (b_n \times X_n)$$

where y is the target variable and $X_1 \dots X_n$ are the independent variables.

- **Exponential regression:** The function attempts to calculate the coefficients of an exponential curve that best fits the input data. The natural log (ln) of the input target variables is calculated and then the same calculations used for linear regression are performed. Once the straight-line coefficients are calculated, exponential regression uses the natural exponential of the values, which results in the coefficients for the formula of an exponential curve. The calculated formula has the following format:

$$y = b_0 \times b_1^{X_1} \times b_2^{X_2} \times \dots \times b_n^{X_n}$$

where y is the target variable and $x_1 \dots x_n$ are the independent variables.

- **Logistic regression:** The function is used to determine the most likely outcome of a set of finite values. The technique uses a chi-square test to determine the probability of each possible value and provide a predicted outcome based on the highest likelihood.
- **Cluster:** This function offers a method of grouping data values based on similarities within the data. A specified or determined number of groups, or clusters, is formed allowing each data value to be mathematically categorized into the appropriate cluster.
- **Decision tree:** The function generates a series of conditions based on independent input to determine a predicted outcome. The result is a hierarchical structure with the ability to lead a set of input to the most likely outcome.
- **Time Series:** This function attempts to make forecasts based on a series of time-related input data. It consists of numerous techniques that can be applied to either seasonal or non-seasonal data.
- **Association:** This function looks for relationships between items. The most common example of this is market basket analysis.

4 Set specialized parameters based on the Type of Analysis selected:

Analysis Type	Parameters
Cluster	Do one of the following: • Specify the exact number of clusters to be generated from the set of data. • Allow MicroStrategy Data Mining Services to determine the optimal number of clusters based on the training data. The algorithm creates multiple models, starting with two clusters, and continues to add more clusters one at a time. With each additional cluster, the quality of the model is assessed. The quality of the current model is measured by calculating the total distance of all records to the centers of their assigned clusters ($DCurr$). This result is compared to the same result for the previously generated model ($DPrev$). This process continues until the amount of improvement, $(DPrev - DCurr) / DPrev$, is less than the amount specified by the percent improvement parameter, or the maximum number of clusters is reached. Upon completion of this process, the model with the best quality is used in the predictive metric.

Analysis Type	Parameters
Decision tree	When MicroStrategy trains a decision tree model, the decision tree algorithm splits the training data into two sets; one set is used to develop the tree and the other set is used to validate it. Prior to MicroStrategy 9.0, one fifth of the training data was always reserved for validating the model built on the remaining four fifths of the data. The quality of this model (referred to as the holdout method) can vary depending on how the data is split, especially if there is an unintended bias between the training set and the validation set. Introduced in MicroStrategy 9.0, K-folds cross-validation is an improvement over the holdout method. The training data is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are used to build a model. Then the result across all k trials is computed, typically resulting in a better model. Since every data point is in the validation set only once, and in the training dataset k-1 times, the model is much less sensitive to how the partition is made. A downside is that training time tends to increase proportionally with k, so MicroStrategy allows the user to control the k parameter, limiting it to a maximum value of 10.
Time series	Number of time periods in the seasonal cycle: This value allows for the specification of seasonality inherent to the training data, and may be defined as follows: • Zero (default) or 1: This indicates that no attempt is made during analysis to find seasonality in the training data. In this case, two types of analysis are performed: double (linear) and triple (quadratic) exponential smoothing. Once the analyses are complete, the model which best fits the training data is created. • > 1: This indicates that the training data has seasonality, consisting of the specified number of time periods. In this case, two types of Winter's exponential smoothing (additive and multiplicative) are performed. Once the analyses are complete, the model which best fits the training data is created.

Analysis Type	Parameters
Association	Maximum number of items per antecedent: This setting defines the maximum number of items that can be included in each antecedent itemset (consequent itemsets can only contain a single item). For example, if set to three, items for each antecedent will be grouped into itemsets containing one, two, or three items. In a transaction that includes the four items beef, onions, potatoes, and soda, a maximum of two creates antecedents with no more than two items, while still including each item in the analysis. The default value for this setting is two. Increasing this number may lead to the generation of more rules and, as a consequence, long execution time for training reports. Minimum confidence: The minimum probability that qualifying rules should have. For example, if set to 10%, then an association rule must have a confidence of 10% or more to appear in the model. The default value for this setting is 10%. Increasing this value may lead to the generation of fewer rules. Minimum support: The minimum number of transactions an itemset must occur in to be considered for an association rule. For example, if set to 1%, then itemsets must appear, on average, in one transaction out of 100. The default value for this setting is 10%. Increasing this value may lead to the generation of fewer rules. Maximum consequent support: The maximum support of the consequent allowed for qualifying rules. This can be used to avoid including obvious recommendations in the resulting rules. For example, if set to 99%, then rules that have a consequent support greater than 99% are not included in the resulting model. The default value for this setting is 100%. Decreasing this value may lead to the generation of fewer rules.

- 5 Click **Next** to open the Select Metrics page.
- 6 Select the metrics and attributes from your project or MDX cube to be used in the training metric. To locate and select the metrics and attributes to be used, you can use the following:
 - **Object Browser:** This option is available for all types of training metrics. Use the Object Browser to browse your project for metrics and attributes to be used in the training metric. If you are creating MDX cube training metrics, you can only select metrics and attributes within the MDX cube.
 - **Report Objects:** This option is available if you are creating derived training metrics only. Use the Report Objects to select attributes and metrics from the report to be used in the training metric.

- 7 To add a metric or attribute to use in the training metric, click the right arrow next to the type of metric. For example, to add a dependent metric, select the metric in the Object Browser, then click the right arrow next to Dependent Metric.

If an attribute is selected as an input, the Training Metric Wizard automatically creates a metric expression to be used as input to the training metric. By default, this expression uses the following format:

```
Max (AttributeName) {~}
```

The ID form of the attribute is used as an input to the training metric. If a different attribute form should be used as an input, a metric must be manually created prior to creating the training metric. For steps on how to create this metric, see [Creating metrics to use additional attribute forms as inputs for predictive metrics, page 777](#).

The different metrics that make up the training metric are described below.

- **Dependent Metric** is the metric or attribute representing the field value to be predicted by the model. All types of analysis except cluster analysis require a dependent metric.

You can select only one dependent metric or attribute.

- **Independent Metrics** are the metrics and attributes for each of the independent variables to be considered while generating the predictive model. All types of analysis require independent metrics or attributes.

Select at least one independent metric or attribute.

- **Segmentation Metrics** are optional selections that apply only to linear, exponential, and logistic regression analyses. The Training Metric Wizard can generate either regression models or tree regression models.
 - In a tree regression model, the leaves of the tree contain a regression model trained on the records associated with that leaf of the tree. For example, if a metric or attribute representing a geographic region is used as a Segmentation Metric, each region will have its own unique forecast. Thus, multiple regression models can be included in a single predictive metric.
 - Add one metric or attribute to represent each level within the tree of the model.

- If no Segmentation Metrics are specified, a standard regression model is generated.
- 8 You can use the default options for the algorithm and variable settings by clearing the **Show advanced options** check box. To specify more options for these settings, do the following:
 - a Select the **Show advanced options** check box.
 - b Click **Next** to open the Advanced Options page.
 - c You can define variable reduction for linear and exponential regression. Refer to the MicroStrategy online help for more information on this setting.
 - d To specify options for independent and dependent inputs, click **Advanced Variable Settings**. See the online help for more information on these settings. To save your changes and return to the Advanced Options page, click **OK**.
 - 9 Click **Next** to open the Select Output page.
 - 10 Select the destination of where each predictive metric is saved. Your options depend on the type of training metric being created:
 - **Folder:** This option is available if you are creating stand-alone training metrics or derived training metrics. Click ... (the Browse button) to define the location in your MicroStrategy project to save the predictive metrics created by the training metric. If you select this option, predictive metrics are saved as stand-alone predictive metrics.

If you are using derived training metrics, you should select this option when a predictive model worthy of deployment is found. Selecting this option saves the predictive model as stand-alone predictive metrics.
 - **Report Objects:** This option is available if you are creating derived training metrics only. Select this option to create the predictive metrics as derived predictive metrics. A derived predictive metric exists only within the report used to create the predictive metric. This capability is particularly useful during the exploratory phase of the data mining process, during which you can test many variables and create a large variety of models.

The way in which derived predictive metrics are created depends on whether the derived training metrics are included directly on a report or on a report included in a document as a Grid/Graph:

- When the report is executed to create predictive metrics (see [Create predictive metrics from training metrics, page 820](#)), the following occurs:
 - The derived training metrics are moved off the report's template; the template determines what is displayed for the report. The derived training metrics are still available in the Report Objects of the report, which supplies the definition of the report. If you update the derived training metrics and need to re-create their associated derived predictive metrics, you must add the derived training metrics back to the template of the report. When the derived predictive metrics are re-created, new versions are created to keep a history of the predictive modeling.
 - Any derived training metrics that are moved off the report's template during report execution are replaced with the first predictive metric selected to be generated for the derived training metric. When the derived training metric is initially replaced with the derived predictive metric, the values are not automatically updated to reflect the predictive metric analysis. Re-execute the report to update the values for the derived predictive metric.
 - All other predictive metrics selected to be generated are added to the Report Objects and are not on the report grid by default. This makes the predictive model available for viewing only. To execute the derived predictive metric, you must add the derived predictive metric to the grid of the report.
- When the document including the report as a Grid/Graph is executed to create predictive metrics (see [Create predictive metrics from training metrics, page 820](#)), the following occurs:
 - All predictive metrics are added to the available dataset objects for the report and are not included on the report grid by default. These predictive metrics are not saved in the report itself and are only available with the report as a dataset in the current document. This makes the predictive model available for viewing only. To execute the derived predictive metrics, you must add the derived predictive metrics to the grid of the report or directly to a section within the document.
- Managed Objects folder: If you are creating MDX cube training metrics, the training metrics are created in the Managed Objects folder of the MicroStrategy project.

- 11 To automatically create the predictive metric each time a report that contains the training metric is executed, select the **Automatically create on report execution** check box. This option automatically overwrites any predictive metrics saved with the same name in the specified location. If you do not want to overwrite the predictive metric, clear this check box. For more information, see [Create predictive metrics from training metrics, page 820](#).

If you are creating a derived training metric, the Automatically create on report execution check box is selected and cannot be cleared. This is to ensure that predictive metrics can be created for the derived training metric, since derived training metrics are only available as part of a report or document.

- 12 To include verification records within the model, specify the number of records to include.



If the number of records is set to zero, model verification is not included in the model.

Model verification records are used by the consumer of the model to insure that the results of the consumer are equal to the results intended by the model producer. Each record contains a row of input from the training data along with the expected predicted output from the model.

- 13 If you are performing Association rules analysis, click **Rules**. The Rules to Return dialog box opens.

These criteria determine which rules will be returned as output for a particular input itemset. You can choose from the following options:

- **Rule selection criteria:**
 - **Antecedent only (Recommendation):** Select rules when the input itemset includes the antecedent itemset only. This is the default behavior. Its purpose is to provide a recommendation, based on the input itemset.
 - **Antecedent and consequent (Rule matching):** Select rules when the input itemset includes both the antecedent and consequent itemsets. Its purpose is to locate rules that match the input itemset.
- **Return the top ranked rules:** These options allow for the specification of exactly which rule is to be returned, based on its ranking among all selected rules.
 - **Return the top ranked rules up to this amount:** Defines the maximum number of rules that are to be returned. For instance, if

set to three, the top three rules can be returned as output. In this case, three separate predictive metrics are created. The default value for this setting is one, and its maximum value is 10.

- **Rank selected rules by:** Defines whether the rules are ranked by confidence, support, or lift. Multiple selections may be specified.
- **Select all rules (One rule per row):** This option allows for the creation of a report which displays all rules found within the model. If the corresponding predictive metric is placed on a report, a single rule is returned for each row of the report. The rules are returned in the order they appear within the model, and are not based on an input itemset.

14 Select the Predictive Metric(s) to generate.

- A training metric can produce different output models when it is run on a training report. You must select at least one output model so that the training metric is created, although you can select multiple outputs. A predictive metric will be generated for each output.

For example, if Predicted Value and Probability are selected as output models, then two predictive metrics will be created during the training process. One will output the predicted outcome when used on a report while the other will output the probability of the predicted outcome.



If no predictor type is selected, Predicted Value is selected automatically.

- A default aggregation function is displayed for all types of predictive metrics. The default is the recommended function based on the type of values produced by each model. The aggregation function ensures proper results when drilling and aggregating from a MicroStrategy report.

If desired, you can change the aggregation functions by using the drop-down lists of functions.



If non-leaf level metrics from different hierarchies are used as independent metrics and tree level metrics, you may want to set the aggregation function to None. This can avoid unnecessary calculations and performance degradation. For example, the Quarter attribute from the Time hierarchy is used as an independent variable and Region from the Geography hierarchy is used as a tree level metric. Multiple calculations for each quarter's region can result.

- 15 Click **Next**. The Summary page opens.
- 16 Review the summary of the training metric before creating the metric. Then click **Finish** to create the metric. If it is a derived training metric, the derived training metric is saved in the report with a default name. Otherwise, the Save As dialog box opens.
- 17 Select the MicroStrategy project folder in which to save the new metric. MDX cube training metrics must be saved in the Managed Objects folder of the MicroStrategy project.
- 18 Enter a name for the new metric.
- 19 Click **Save**.

Create predictive metrics from training metrics

You can quickly and easily create predictive metrics from training metrics in either of the following ways:

- If the option to automatically create predictive metrics is disabled on the training metric, execute the training report and then use the Create Predictive Metrics option. For steps, see [To use the Create Predictive Metrics option in the Report Editor, page 822](#).



The Create Predictive Metrics option is available only when you are running reports in MicroStrategy Developer.

- If the option to automatically create predictive metrics is enabled on the training metric, execute the training report to create the predictive metric. This is the default behavior, and is controlled by the **Automatically create on report execution** check box described in [Creating a training metric with the Training Metric Wizard, page 808](#).

If you are using derived predictive metrics and have selected to save the derived predictive metrics to the Report Objects, these derived predictive metrics are created in the following ways:

- When a report is executed to create predictive metrics, the following occurs:
 - The derived training metrics are moved off the report's template; the template determines what is displayed for the report. The derived training metrics are still available in the Report Objects of the report, which supplies the definition of the report. If you update the derived training metrics and need to re-create their

associated derived predictive metrics, you must add the derived training metrics back to the template of the report. When the derived predictive metrics are re-created, new versions are created to keep a history of the predictive modeling.

- Any derived training metrics that are moved off the report's template during report execution are replaced with the first predictive metric selected to be generated for the derived training metric. When the derived training metric is initially replaced with the derived predictive metric, the values are not automatically updated to reflect the predictive metric analysis. Re-execute the report to update the values for the derived predictive metric.
 - All other predictive metrics selected to be generated are added to the Report Objects and are not on the report grid by default. This makes the predictive model available for viewing only. To execute the derived predictive metric, you must add the derived predictive metric to the grid of the report.
- When a document including the report as a Grid/Graph is executed to create predictive metrics, the following occurs:
- All predictive metrics are added to the available dataset objects for the report and are not included on the report grid by default. These predictive metrics are not saved in the report itself and are only available with the report as a dataset in the current document. This makes the predictive model available for viewing only. To execute the derived predictive metrics, you must add the derived predictive metrics to the grid of the report or directly to a section within the document.

After the predictive metric is created in any of these ways, you can use it in reports, documents, and other objects, as discussed in [Using the predictive metric, page 829](#).

Prerequisites

- You must have created a training metric in the Training Metric Wizard. For steps, see [To create a training metric, page 810](#). However, if you have an OLAP Services license, you can create derived training metrics directly on a report. For information on creating derived training metrics, see the *MicroStrategy Developer help*.
- If the training metric is based off an MDX cube, the training metric must be included in an MDX cube report. For steps to create an MDX cube report, see the *MDX Cube Reporting Guide*.

To use the Create Predictive Metrics option in the Report Editor

- 1 Using MicroStrategy Developer, add the training metric to a report.

Ensure that the training metric is on the grid of the report. If the training metric is only in the Report Objects but not on the report grid, no predictive metrics are created for the training metric when the report is executed.

- 2 Execute the training report in MicroStrategy Developer.



The Create Predictive Metrics option is available only when you are running reports in MicroStrategy Developer.

- 3 From the **Data** menu, select **Create Predictive Metric(s)**. The Create Predictive Metric(s) dialog box opens.

- 4 To view information about a training metric, select it from the **Metrics** list, which contains all the training metrics on the report.

- 5 To temporarily rename the predictive metric created by the selected training metric, enter the new name in the **Name** box.



The new name is generated when you click OK, but the changes are not saved when the report is closed.

- 6 You can change the options and associated aggregation function for producing the predictive metric in the Aggregation function area.



The new options are applied when you click OK, but the changes are not saved when the report is closed.

- 7 You can save the PMML from the generated model. To do so, click **Generate PMML File** and then select a location in which to save the PMML file.

- 8 Click **OK** to return to the report.

The predictive metric is created in the location specified by the training metric. It is generated according to the parameters of the training metric.

Using third-party data mining applications

Many companies offer data mining applications and workbenches. These tools are specialized for the data mining workflow and are usually rich in data mining features and options. MicroStrategy Data Mining Services integrates seamlessly with these applications in the following ways:

- MicroStrategy is used as the source of the data mining dataset.
- After the third-party application develops the PMML-based predictive model based on the dataset, the model is imported into MicroStrategy and used to generate reports.

You can use any of the following options to provide the MicroStrategy dataset report to third-party applications:

- Save the dataset report as a data mart in MicroStrategy. For more information on data marts, see [Chapter 12, Accessing Subsets of Data: Data Marts](#).
- Export to a particular file format. See the MicroStrategy online help for more information on exporting reports from MicroStrategy.

Importing the predictive model

After you have created a predictive model and generated PMML to represent that model, the next step is to import the PMML file into your MicroStrategy project. The import function creates predictive metrics from the PMML file.

In addition to importing predictive models directly into your MicroStrategy project, you can also import predictive models into the MDX cubes you have integrated into MicroStrategy, as described in [Importing predictive models for MDX cubes, page 825](#).

Prerequisites

- You have created a dataset report, which is the first step in creating a predictive model. For information on this process, see [Creating a dataset report, page 771](#).
- You have created and trained a predictive model from the dataset report, using MicroStrategy or a third-party application. For information on this process, see [Creating a predictive model, page 785](#).

- You must have the Create Application Objects and Use Metric Editor privileges to access the Import Data Mining Model option. For information on viewing or changing your privileges, see the *MicroStrategy Supplemental Reference for System Administration*.

To import a predictive model into MicroStrategy

- 1 In MicroStrategy Developer, from the **Tools** menu, select **Import Data Mining Model**. The Import Data Mining Model dialog box opens.
- 2 Select the PMML file to import by clicking ... (the Browse button), and then click **Open**.
- 3 You can make the following changes for the predictive metrics:
 - **Select the Predictive Metric Type:** The columns in this table are automatically populated when a predictive model is selected. You can change the information in this table as follows:
 - **Predictor Type:** By default, predictive metrics for all outputs defined within the model selected above will be created and displayed in this column. Clear any outputs for which you do not want to create predictive metrics.
 - **Name:** Displays the default names provided for each predictive metric, which can be changed.
 - **Aggregation function:** Displays the default aggregation functions for the selected predictive metric types. You can change these functions to determine how the output of the model should be aggregated. For more details, see [Aggregating predictive metrics, page 828](#).
- 4 Click **OK** to import the model or models.
- 5 Select the folder in which to save all the predictive metrics.

The import feature reads the PMML automatically and creates the predictive metrics in the folder specified. During the importing process, several key model characteristics are determined from the PMML, including the following:

- **The type of data mining algorithm:** Based on this, the appropriate MicroStrategy data mining function is used in the expression of the metric.

- **The inputs to the data mining function:** Matching the column alias of the predictive input metric is important. Since Data Mining applications use this name to identify each variable, you can use the name of each variable in the PMML to identify which MicroStrategy metric should be used. The import feature robustly handles all potential circumstances in this area as follows:
 - Only one MicroStrategy metric with that name exists: That metric is used automatically as an input to the function.
 - Multiple MicroStrategy metrics with that name exist: The user is prompted to select the right metric.
 - No MicroStrategy metrics with that name exist: The user is prompted to select a metric from within the project, or the user can cancel the import and try again later.
- **The version of PMML:** Since different versions of PMML have different specifications, the version is identified so models are processed properly.
- **PMML validation:** The PMML is validated for consistency with the PMML standard and known vendor-specific extensions.
- **Model verification:** The Import dialog box detects whether the PMML contains verification data and reports this information at the bottom of the dialog box. Verification data can be viewed and tested for expected results by accessing the Model Viewer from the predictive metric.

Once the predictive metric is created, you can use it in reports and other objects. See [Using the predictive metric, page 829](#) for more information.

Importing predictive models for MDX cubes

MicroStrategy allows you to integrate data from MDX cube sources such as SAP BW and Microsoft Analysis Services. Once integrated into MicroStrategy as MDX cubes, you can use the MicroStrategy reporting and analysis features to further analyze your MDX cube source data. For information on integrating MDX cube sources into MicroStrategy, see the *MDX Cube Reporting Guide*.

This also includes performing predictive analysis on your MDX cube data. The steps below show you how to import a predictive model into your MDX cube.

Prerequisites

- You have imported an MDX cube. For steps to import an MDX cube as well as additional requirements to import MDX cubes into MicroStrategy, see the *MDX Cube Reporting Guide*.
- You have created and trained a predictive model from the dataset report, using MicroStrategy or a third-party application. For information on this process, see [Creating a predictive model, page 785](#).
- You must have the Create Application Objects and Use Metric Editor privileges to access the Import Data Mining Model option. For information on viewing or changing your privileges, see the *MicroStrategy Supplemental Reference for System Administration*.

To import a predictive model into MicroStrategy for an MDX cube

- 1 In MicroStrategy Developer, log in to a project that is connected to an MDX cube source.
- 2 If you are using read only mode for the project, from the **Schema** menu, clear the **Read Only Mode** option to switch to edit mode.

Only one user can be editing a project at a given time. Therefore, if someone else is modifying the project, you cannot use the MDX Cube Catalog.
- 3 From the **Schema** menu, select **MDX Cube Catalog**.
 - If you have a single MDX cube source database instance created for the project, the MDX Cube Catalog opens.
 - If you have multiple MDX cube source database instances created for the project, a Database Instance dialog box opens. Select a valid MDX cube source database instance, click **OK**, and the MDX Cube Catalog opens.
- 4 Click the **Cube Mapping** tab.
- 5 From the **Catalog\Cube** drop-down list, select the MDX cube to import a predictive model to.
- 6 From the **Edit** menu, select **Import Data Mining Model**. The Import Data Mining Model dialog box opens.

- 7 Select the PMML file to import by clicking ... (the Browse button), and then click **Open**.
- 8 You can make the following changes for the predictive metrics:
 - **Select the Predictive Metric Type:** The columns in this table are automatically populated when a predictive model is selected. You can change the information in this table as follows:
 - **Predictor Type:** By default, predictive metrics for all outputs defined within the model selected above will be created and displayed in this column. Clear any outputs for which you do not want to create predictive metrics.
 - **Name:** Displays the default names provided for each predictive metric, which can be changed.
 - **Aggregation function:** Displays the default aggregation functions for the selected predictive metric types. You can change these functions to determine how the output of the model should be aggregated. For more details, see [Aggregating predictive metrics, page 828](#).
- 9 Click **OK** to import the model or models.

The import feature reads the PMML automatically and creates the predictive metrics within the Compound Metrics of the MDX cube. During the importing process, several key model characteristics are determined from the PMML, including the following:

- **The type of data mining algorithm:** Based on this, the appropriate MicroStrategy data mining function is used in the expression of the metric.
- **The inputs to the data mining function:** Matching the column alias of the predictive input metric is important. Since Data Mining applications use this name to identify each variable, you can use the name of each variable in the PMML to identify which metric within the MDX cube should be used. The import feature robustly handles all potential circumstances in this area as follows:
 - Only one metric with that name exists in the MDX cube: That metric is used automatically as an input to the function.
 - Multiple metrics with that name exist in the MDX cube: The user is prompted to select the right metric.

- No metrics with that name exist in the MDX cube: The user is prompted to select a metric from within the MDX cube, or the user can cancel the import and try again later.
- **The version of PMML:** Since different versions of PMML have different specifications, the version is identified so models are processed properly.
- **PMML validation:** The PMML is validated for consistency with the PMML standard and known vendor-specific extensions.
- **Model verification:** The Import Data Mining Model dialog box detects whether the PMML contains verification data and reports this information at the bottom of the dialog box. Verification data can be viewed and tested for expected results by accessing the Model Viewer from the predictive metric.

Once the predictive metric is created, you can use it in reports and other objects. See [Using the predictive metric, page 829](#) for more information.

Aggregating predictive metrics

One of the most powerful features of integrating data mining models with a business intelligence platform is the ability to drill up and down through layers of data. For predictive results to make sense, the aggregation function to be used must be specified.

The steps that need to be taken to ensure the predictive metric aggregates properly under all circumstances are described below:

- Place the predictive function within an aggregation function.
- Specify the dynamic aggregation function on the predictive metric. This is required only when the predictive metric is used with MicroStrategy OLAP Services.

Choosing proper aggregation function requires some knowledge about how the model behaves. For example:

- If the predictive metric generates a score that is a zero or a one, use Sum to calculate the number of “one” scores.
- If the predictive metric generates a “linear” output, like “Forecasted Revenue,” usually from a regression predictor, use Sum to roll up the predictive results.

- If the predictive metric generates a confidence or percentage, use Average to calculate the mean confidence.
- If the predictive metric generates a numeric classifier, like a cluster/segment number, use Mode to calculate the most common classifier.
- For models with outputs that cannot be aggregated, select “None” as the aggregation function.

While an explicit aggregation function is useful for normal reports, deployments that take advantage of MicroStrategy OLAP Services should also set the predictive metric's dynamic aggregation function. The Import Data Mining Model feature in MicroStrategy Developer does this automatically, but you can also set the dynamic aggregation function using the Metric Editor.

To set the dynamic aggregation function

- 1 Open the predictive metric in the Metric Editor.
- 2 Select the **Subtotals/Aggregation** tab.
- 3 Select the appropriate Dynamic Aggregation function.

Using the predictive metric

You can use the predictive metric in reports or documents, as well as in other objects. You can also view the PMML information of a predictive metric, using the Predictive Model Viewer.

Using the predictive metric in reports and documents

With the predictive model implemented as a metric, it can be used in reports or documents to determine possible trends and outcomes. Creating a report or document for data mining is similar to creating a regular report or document. There are no special requirements, other than including the predictive metric. For steps to create a report, see the *Basic Reporting Guide*. For steps to create a document, see the *Report Services Document Creation Guide*.

Predictive models are generated based on a certain level of input data. Therefore, in order for a model to produce valid results when scored, the model must be provided with the same level of input data. If the user of the model does not have access to this level of data, then the model cannot be expected to return valid scores. For example, a security filter on the user does not allow access to the appropriate level of data.

Since Exponential Smoothing gives greater weight to more recent data, time series models (see [Time series analysis, page 798](#)) should be updated whenever new data is available. While this can be said for other types of model as well, this is particularly true for time series models since recent results can significantly influence the model. Therefore, for time series models, it is common to deploy the training metrics (see [Creating a training metric with the Training Metric Wizard, page 808](#)) instead of the predictive metrics (see [Create predictive metrics from training metrics, page 820](#)). This guarantees the time series forecast results reflect the most recent data.

Using the predictive metric in other objects

In addition to reports and documents, predictive metrics can also be used in other objects. Generally, wherever a regular MicroStrategy metric is used, a predictive metric can be used. The objects where predictive metrics can be used include:

- Filters
- Custom Groups
- Consolidations
- Derived metrics
- Shortcut metrics

For more information on using or creating any of these objects, see the online help or the *Basic Reporting Guide*.

Predictive Model Viewer

Predictive metrics are different from other MicroStrategy metrics since they implement a data mining model, which is more than a mathematical expression. These predictive models, and the PMML that describes them, contain information about variables, variable transformations, and details

about the data mining techniques they implement. All this information can be viewed on MicroStrategy Developer using the Predictive Model Viewer.

You can access the Predictive Model Viewer in one of the following ways:

- In MicroStrategy Developer, right-click a predictive metric in the right pane and select **View Predictive Model**.
- In the Metric Editor, when you are editing a predictive metric, select **View Predictive Model** from the **Tools** menu.
- In a MicroStrategy Developer report or graph, right-click a predictive metric in the right-hand pane and select **View Predictive Model**.

For information on how to use this viewer, see the online help.

Data mining examples

- *Revenue forecasting example (using linear and seasonal regression), page 832*: To aid in setting goals for next year, you want to establish a forecast of your company's revenue based on existing trends.
- *Campaign management example (using logistic regression), page 836*: Your company wants to improve the effectiveness of its marketing campaigns, with the goals of reducing costs and increasing the percentage of positive responses.
- *Segmentation example (using cluster analysis), page 838*: You use cluster analysis to group your customers into exactly five segments based on demographics and psychographics.
- *Telco churn example (using decision tree analysis), page 841*: You are reviewing your telecommunication trends and are interested in the likelihood of customers to churn based on psychographics and communication habits.
- *Campaign management example: Importing a third-party neural network model, page 844*: You want to execute a scenario similar to the campaign management example created using logistic regression, but instead use a third-party data mining application.
- *Market basket analysis (using association rules analysis), page 848*: Market basket analysis studies retail purchases to determine which items tend to appear together in individual transactions. Retailers use market basket analysis for their commercial websites to suggest additional items to purchase before a customer completes his order.

Revenue forecasting example (using linear and seasonal regression)

To aid in setting goals for next year, you would like to establish a forecast of your company's revenue based on existing trends.

You will use the revenue values already available in your MicroStrategy project. Predictions of future revenue will be determined at quarterly intervals. The quarters for the future dates are already defined in your MicroStrategy project, but revenue figures for each are not yet available.

Your model will require a continuous input of quarters to recognize a regression pattern. A Quarter attribute is commonly formatted to represent a year followed by a quarter, and therefore cannot be used as a continuous index into the quarters. To address this requirement, create a Quarter Index metric using the Quarter attribute. Since a year followed by a quarter is still a sequential list of values, you can often create a Quarter Index metric with the following simple expression:

```
Rank (Max (Quarter@ID) {Quarter} )
```

This uses the ID values for a Quarter attribute, and creates a sequential list of values by ranking them. For example, assume you are using values dating back to the beginning of 2012. Quarter 201202 (Q2 in year 2012) has an index of 2 since it is the second quarter overall. Quarter 201403 (Q3 in year 2014) has an index of 11.

If the ID values for your attribute representing quarter data are not sequential, you must determine a way to convert the ID values into a sequential list of values.



The example Tutorial project includes a Quarter Index metric, along with other reports and metrics created for this forecasting example. Some of the definitions of metrics and reports within the Tutorial project are different than the simplified descriptions in this example. You can use the definitions provided in the steps here, or the definitions in the Tutorial project, depending on what works best for your reporting environment.

To begin your data mining analysis, use the Training Metric Wizard to design a training metric, following the steps described below.

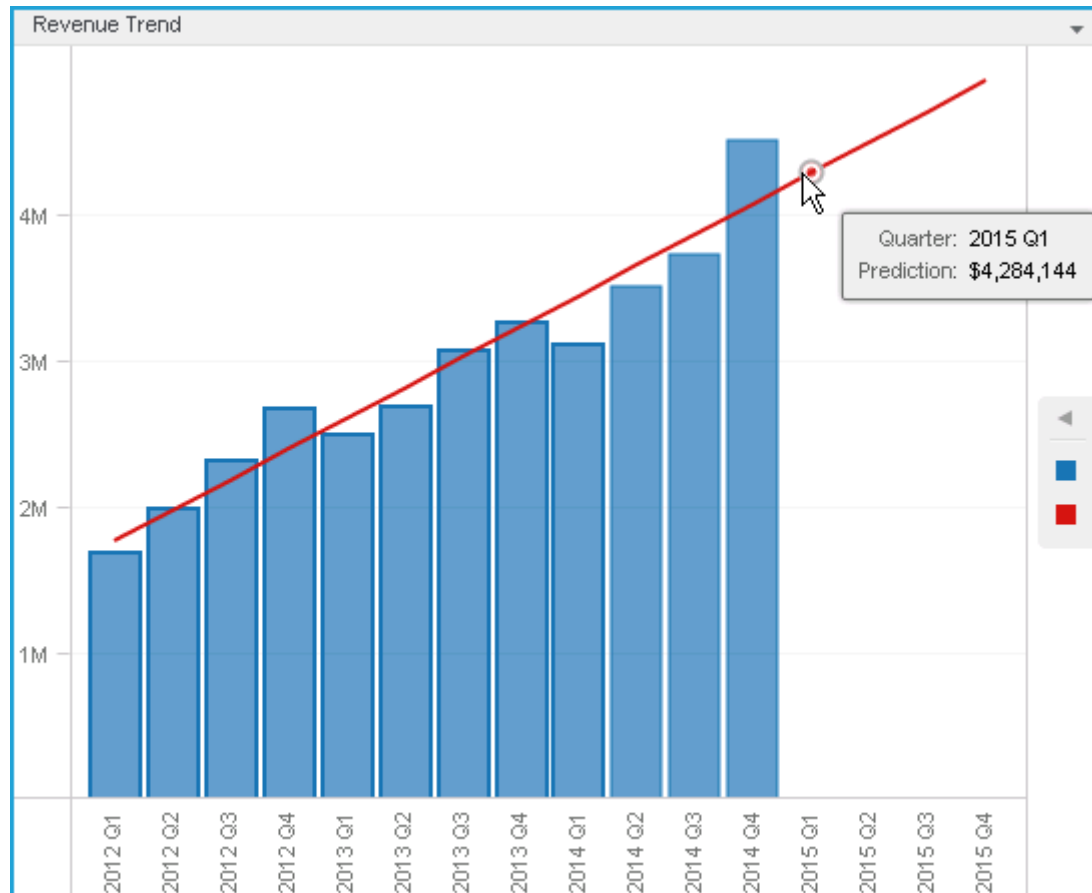
To create a training metric for linear regression analysis

- 1 In MicroStrategy Developer, select **Training Metric Wizard** from the **Tools** menu. The Training Metric Wizard opens on the Introduction page.
- 2 Click **Next**. The Select Type of Analysis page opens.
- 3 Select **Linear regression** as the type of analysis.
- 4 Click **Next**. The Select Metrics page opens.
- 5 Add Revenue as the **Dependent Metric**.
- 6 Add the Quarter Index metric to the list of **Independent Metrics**.
- 7 Clear the **Show advanced options** check box to use the default settings for variable reduction and other variable settings.
- 8 Click **Next**. The Select Output page opens.
- 9 Select the **Automatically create on report execution** check box.
- 10 Within the Predictive metrics to generate area, select **Predicted Value**.
- 11 Click **Finish** to save and create the metric.

For more information on the Training Metric Wizard, see [Creating a training metric with the Training Metric Wizard, page 808](#) or refer to the MicroStrategy online help.

Next, create a report that includes the new training metric and the Quarter attribute. Include the Revenue metric to compare the values calculated by the training metric with the original values. Review the Report Data Options and VLDB properties for your report to ensure that outer join results are displayed for the metrics of your report. Execute the report.

The training metric generates a straight line that best fits the Revenue data. The report, converted into a dashboard to display this trend, is shown below.



A predictive metric is created in the folder you specified in the Training Metric Wizard. The default location is the My Objects folder.

The predictive metric accurately depicts a linear line through the Revenue data, but for this example, assume that you are not satisfied with the predictions. Your data is seasonal and you need to forecast the fluctuations that will occur throughout the year.

Seasonality is recognized by adding another independent metric to the training metric. This additional metric specifies the quarter within the year associated with each Quarter Index value. For example, the Quarter Index values of 1, 5, and 9 are all from the first quarter. The Quarter of Year metric uses the same basic formula as Quarter Index. The BreakBy parameter is defined as year so that the ranking is restarted for each year, allowing each quarter to be numbered 1 through 4 for a given year. The formula is shown below:

```
Rank<BreakBy={Year}> (Max (Quarter@ID) {Quarter} )
```


To include seasonality in your data mining model, complete the following steps.

To add seasonality to the data mining model

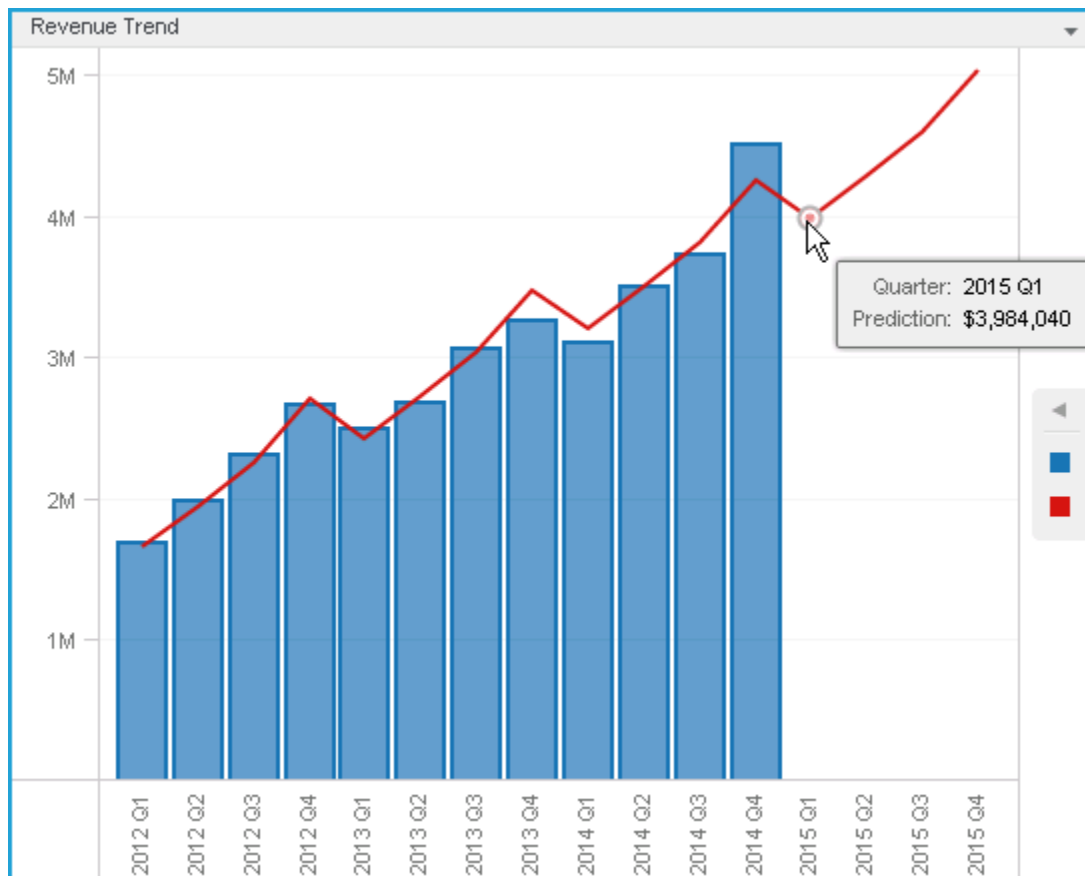
- 1 In MicroStrategy Developer, double-click the training metric you created in *To create a training metric for linear regression analysis, page 833* to open the Training Metric Wizard to the Introduction page.
- 2 Click **Next**. The Select Type of Analysis page opens.
- 3 Do not change any of the values on this page. Click **Next**. The Select Metrics page opens.
- 4 Add the Quarter of Year metric to the list of **Independent Metrics**, which already includes the Quarter Index metric.
- 5 Click **Next**. The Select Output page opens.
- 6 Rename the predictive metric so that the existing linear predictive metric is not overwritten.
- 7 Save the training metric with a new name to distinguish it as a seasonal prediction.



For more information on the Training Metric Wizard, see *Creating a training metric with the Training Metric Wizard, page 808* or refer to the MicroStrategy online help.

You can now re-execute the report you created earlier that included the training metric, Quarter, and Revenue. The results of the training metric now recognize the fluctuations in revenue throughout each year and predict

values accordingly. Notice that the data accounts for seasonality and is no longer a straight line, as shown in the report below.



A predictive metric is created in the folder you specified in the Training Metric Wizard. The default location is the My Objects folder.

Campaign management example (using logistic regression)

Recall the campaign management scenario described in [Data Mining Services, page 764](#). Your company wants to improve the effectiveness of its marketing campaigns, with the goals of reducing costs and increasing the percent of positive responses. The results of a previous campaign will be analyzed to determine what factors, if any, can be used to predict the performance of a similar future campaign. Use logistic regression analysis to generate a predictive model. Logistic regression selects the most likely outcome from a set of distinct possibilities.

A recent back-to-school sale campaign produced hundreds of respondents from a pool of thousands of customers. The campaign was based on the following:

- Age
- Gender
- Household count
- Income range

To predict future campaigns based on the back-to-school sale campaign, you want to use all of these attributes as predictors in the predictive model. Therefore, you must create metrics for each attribute form. Some example metrics for this report are as follows:

```
Max([Customer Age Range]@ID) {Customer}  
Max([Customer Gender]@DESC) {Customer}  
Max([Customer Household Count]@DESC)  
{Customer}
```

 The example Tutorial project includes reports, metrics, and other objects created for this campaign management example (search the project for “Campaign Management”). You can use the objects in the Tutorial project to step through the example and determine how it can be applied to your reporting environment.

Use the Training Metric Wizard to design a training metric, following the procedure below.

To create a training metric for logistic regression analysis

 This procedure assumes you have already created a Back-to-School Sale Responder metric to use as the dependent metric.

- 1 In MicroStrategy Developer, select **Training Metric Wizard** from the **Tools** menu. The Training Metric Wizard opens on the Introduction page.

To skip the Introduction page when creating training metrics in the future, select the **Don't show this message next time** check box.

- 2 Click **Next** to open the Select Type of Analysis page.
- 3 Select **Logistic regression** as the type of analysis.

- 4 Click **Next** to open the Select Metrics page.
- 5 Select **Back-to-School Sale Responder** as the **Dependent Metric**.
- 6 Add the **Age Range**, **Gender**, and **Household Count** metrics to the list of **Independent Metrics**.
- 7 Click **Next** to open the Select Output page.
- 8 Select the **Automatically create on report execution** check box.
- 9 Select **Predicted Value**.
- 10 Click **Finish** to save and create the metric. You can now include the metric in a training metric to create a predictive metric, as described in [Creating a training metric with the Training Metric Wizard, page 808](#).
- 11 Create a new report with the training metric, Back-to-School Sale Responder metric, and the Customer and Order attributes.
- 12 Filter the report to include only orders dated during the back-to-school promotional period. For example, you can create a filter that only includes the months of August and September.
- 13 Execute the report to generate a logistic regression model.

A predictive metric is created in the folder you specified in the Training Metric Wizard. The default location is the My Objects folder.

By adding the predictive metric to a new report along with the Customer attribute and the Back-to-School Sale Responder metric, the accuracy of the prediction is shown to be correct almost 100% of the time; there were only a few incorrect predictions out of thousands of customers.

The predictive metric is ready to be used to target customers who are likely to respond to a future campaign.

Segmentation example (using cluster analysis)

Clustering attempts to segment items so that members of one group are more similar to each other than to members of other groups. These items are most often customers, but can also be products, patients, prescriptions, phone calls, emails, or any other item relevant to the enterprise. Clustering

algorithms do the segmentation by analyzing the characteristics of the items and finding the best ways to group them by similarities.

You will use cluster analysis in the following example to group your customers into exactly five segments based on demographics and psychographics.

Each of the following will be used as inputs into the training metric:

- Age Range
- Education
- Gender
- Housing Type
- Marital Status

As before, you must create a metric for each attribute form, which can then be used to create a predictive metric. Examples are:

```
Max([Customer Age Range]@ID) {Customer}  
Max([Customer Gender]@DESC) {Customer}  
Max([Housing Type]@DESC) {Customer}
```



The example Tutorial project includes reports, metrics, and other objects created for this segmentation example (search the project for “Cluster Analysis”). You can use the objects in the Tutorial project to step through the example and determine how it can be applied to your reporting environment.

Use the Training Metric Wizard to design a training metric, following the procedure below.

To create a training metric for cluster analysis

- 1 In MicroStrategy Developer, select **Training Metric Wizard** from the **Tools** menu. The Training Metric Wizard opens on the Introduction page.

To skip the Introduction page when creating training metrics in the future, select the **Don't show this message next time** check box.

- 2 Click **Next** to open the Select Type of Analysis page.
- 3 Select **Cluster** as the type of analysis.

- 4 Specify exactly **5** clusters for **Model Specifications**.
- 5 Click **Next** to open the Select Metrics page.
- 6 Add the **Age Range, Education, Gender, Marital Status, and Housing Type** metrics to the list of **Independent Metrics**.
- 7 Click **Next** to open the Select Output page.
- 8 Select the **Automatically create on report execution** check box.
- 9 Select **Predicted Value**.
- 10 Click **Finish** to save and create the metric. You can now include the metric in a training metric to create a predictive metric, as described in [Creating a training metric with the Training Metric Wizard, page 808](#).
- 11 Often, training reports do not require a large number of rows of data to formulate an acceptable result. You will need to reduce the number of rows in your training report by sampling the data. For this example, your sample will include a random set of 20% of the customers. Create a filter to define this random set. Use the filter on a new report with the Customer attribute and the training metric created above.
- 12 Execute the report.

A predictive metric is created in the folder you specified in the Training Metric Wizard. The default location is the My Objects folder.

Add the predictive metric to a new report with Customer, Age Range, Gender, Education, Housing Type, and Marital Status to see the relationships between segments.

A custom group can be created based on each segment to further segregate the groupings.

In some cases, the ideal number of clusters is not known before performing an analysis. To overcome this obstacle, Data Mining Services offers a feature to determine the optimal number of clusters while training the cluster model. Retrain the model above, allowing Data Mining Services to determine how many clusters to create, following the procedure below.

To retrain the model to automatically determine the number of clusters

- 1 Double-click the existing training metric to open the Training Metric Wizard.
- 2 Click **Next** to open the Select Type of Analysis page.
- 3 Specify the maximum number of clusters as **10** in **Model Specifications**.
- 4 Click **Next** to open the Select Metrics page.
- 5 Click **Next** to open the Select Output page.
- 6 Rename the predictive metric to prevent overwriting the existing cluster predictive metric.
- 7 Save the training metric with a new name, specifying that it is optimal.
- 8 Add the optimal training metric to a new report that contains the Customer attribute. Filter the report on a random 20% sample of customers.
- 9 Execute the report.

Notice that Data Mining Services creates a predictive model with only two clusters, the optimal number of clusters less than the maximum allowed. Recall that the maximum specified in the Training Metric Wizard was 10.

A predictive metric is created in the folder you specified in the Training Metric Wizard. The default location is the My Objects folder.

Telco churn example (using decision tree analysis)

You are reviewing your telecommunication trends and are interested in the likelihood of customers to churn based on psychographics and communication habits.

Use a decision tree to analyze the following inputs:

- Average Minutes during Off-Peak times
- Average Minutes during Peak times

- Dropped Calls
- Helpdesk Calls
- Renewals
- Age Range
- Gender
- Household Count
- Marital Status
- Income Bracket

You will need to create metrics for each attribute form. These metrics will be used as inputs into the training metric. Examples are:

```
Max([Customer Age Range]@ID) {Customer}  
Max([Customer Gender]@DESC) {Customer}  
Max([Household Count]@DESC) {Customer}
```



The example Tutorial project includes reports, metrics, and other objects created for this telco churn example (search the project for “Telco Churn”). You can use the objects in the Tutorial project to step through the example and determine how it can be applied to your reporting environment.

Use the Training Metric Wizard to design a training metric, following the procedure below.

To create a training metric for decision tree analysis



This procedure assumes you have already created a TelcoChurn metric to use as the dependent metric.

- 1 In MicroStrategy Developer, select **Training Metric Wizard** from the **Tools** menu. The Training Metric Wizard opens on the Introduction page.

To skip the Introduction page when creating training metrics in the future, select the **Don't show this message next time** check box.

- 2 Click **Next** to open the Select Type of Analysis page.
- 3 Select **Decision tree** as the type of analysis.

- 4 Specify the k value for k-Fold Cross Validation.
- 5 Click **Next** to open the Select Metrics page.
- 6 Select **TelcoChurn** metric as the **Dependent Metric**.
- 7 Add the following metrics to the list of **Independent Metrics**:
 - AvgMinOffPeak
 - AvgMinPeak
 - DroppedCalls
 - HelpdeskCalls
 - Renewals
 - Age Range
 - Gender
 - Household Count
 - Marital Status
 - Income Bracket
- 8 Click **Next** to open the Select Output page.
- 9 Select the **Automatically create on report execution** check box.
- 10 Select **Predicted Value**.
- 11 Click **Finish** to save and create the metric. You can now include the metric in a training metric to create a predictive metric, as described in [Creating a training metric with the Training Metric Wizard, page 808](#).
- 12 Create a new report containing the training metric and the Customer attribute. Filter the report on every seventh customer. (Which customers to include is arbitrary but must offer a decent sample size of the data.)
- 13 Execute the report to generate a decision tree model.

A predictive metric is created in the folder you specified in the Training Metric Wizard. The default location is the My Objects folder.

When applied to all customers, the predictive metric reveals hundreds of customers out of thousands of customers that are likely to churn. Efforts can now be made to target these customers for extra deals or value analysis.

Campaign management example: Importing a third-party neural network model

The *Campaign management example (using logistic regression)*, page 836 used Data Mining Services to generate a logistic regression model to improve the effectiveness of marketing campaigns, with the goals of reducing costs and increasing the percent of positive responses. You would like to execute a similar scenario using a third-party data mining application.

A recent back-to-school sale campaign was based on the following:

- Age
- Gender
- Education
- Household count
- Revenue

The results of a previous campaign will be analyzed to determine what factors, if any, can be used to predict the performance of a similar future campaign. The previous campaign counted hundreds of favorable responses out of thousands of customer orders. Accurate data on the targeted customers and their responses have been entered into a MicroStrategy project.

The first step in data mining is to design a dataset report. The pertinent information for this report is listed below:

- Customer
- Order
- Age Range
- Education
- Gender
- Household count

- Revenue
- Promotion ID

You want to use all of these attributes, except for Customer and Order, as predictors in the predictive model. Therefore, you must create a metric for each attribute form. These metrics will be used as inputs into the training metric. Some example predictive inputs metrics for this report are as follows:

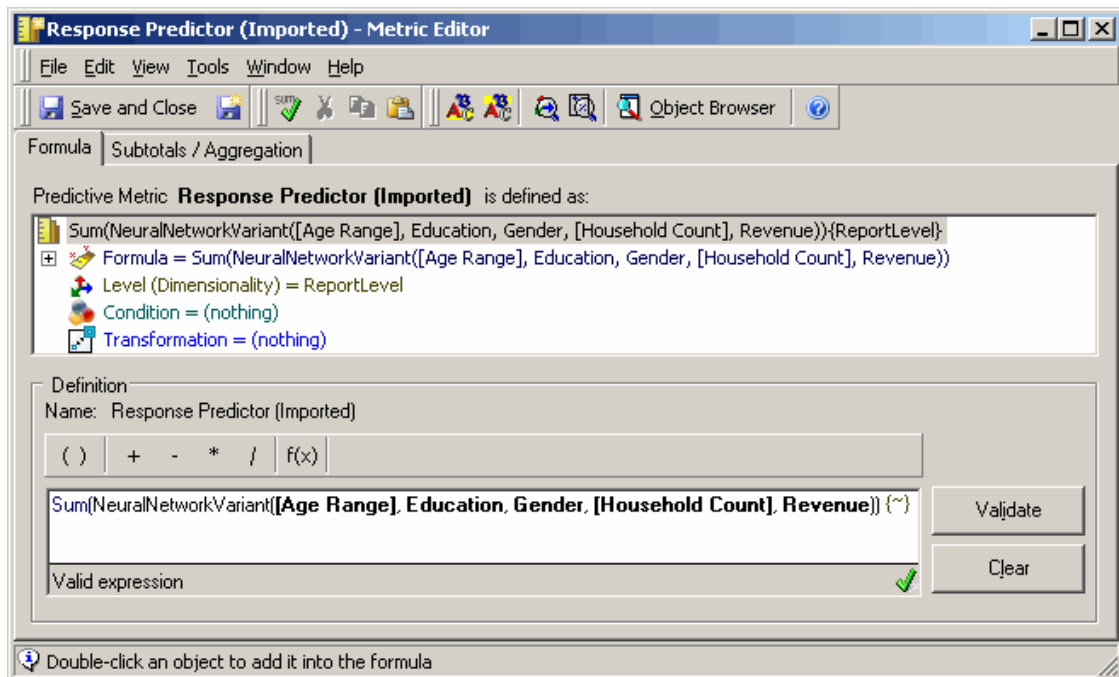
```
Max([Customer Age Range]@ID) {Customer}
Max([Customer Education]@DESC) {Customer}
```

 The example Tutorial project includes reports, metrics, and other objects created for this campaign management example (search the project for “Neural Network”). You can use the objects in the Tutorial project to step through the example and determine how it can be applied to your reporting environment.

Filter the dataset report to include only the customer order dated during the period of the campaign, in this case, between the start of August and the end of September.

Once the dataset report is complete, use it to generate the dataset.

After analyzing the dataset with a third-party data mining application, a predictive model is developed and the PMML representation of that model is imported into MicroStrategy. The predictive metric is shown below:



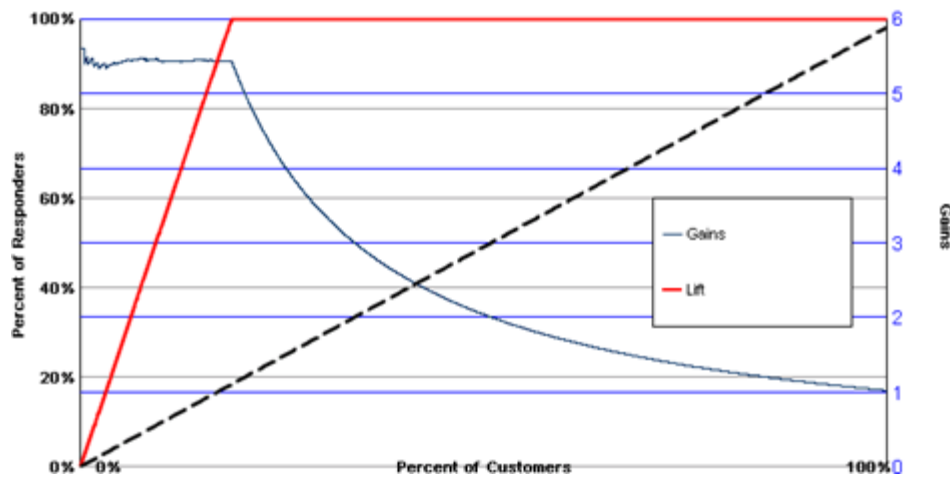
The predictive metric actually uses a neural network algorithm to score each record and determine if that customer is likely to respond or not. Validate the predictive metric against the original data in a report like the one below, which compares the actual response with the response calculated by the predictive metric.

Customer		Metrics	Back-to-School Sale Responder	Response Predictor (Imported)	Response Propensity (Imported)	Correct?
Total			1,002	1,034	18%	5,558
Count			5612	5612	5612	5612
Aadland	Warner		0	0	0%	1
Aafedt	Wendy		0	0	0%	1
Aagesen	Bink		0	0	0%	1
Aamodt	Stacy		1	1	99%	1
Aaronson	Maxwell		0	0	0%	1
Aba	Blain		0	0	0%	1
Abad	Geoffrey		0	0	0%	1
Abarbanel	Hassam		0	0	0%	1
Abatemarco	May		0	0	1%	1
Abbasi	Dwayne		0	0	0%	1
Abbenhaus	Lonzie		0	0	0%	1
Abbott	Rosella		1	1	99%	1
Abbott	Delores		1	1	99%	1
Abbruscato	Shaun		0	0	0%	1
Abdala	Stacy		0	0	0%	1
Abdallah	Erling		0	0	0%	1
Abdullah	Courtney		1	1	99%	1

This report shows the actual (Back-to-School Sale Responder) and predicted (Response Predictor) results for the thousands of customers who were targeted in the first campaign. It also shows each customer's Response Propensity, which can be thought of as the probability or confidence that a customer will respond favorably to the campaign. Finally, the column labeled Correct? indicates if the Response Predictor matches the actual value.

As you can see from the data, the Response Predictor metric is accurate about 99% of the time. This accuracy is definitely acceptable for marketing purposes.

Another way to look at this model is by using a Lift Chart, as shown below. Lift Charts show the gain, or lift, possible when using the predictive metric to target customers over selecting customers at random.



This chart plots the percent of customers available to be part of the campaign against the percent of customers that responded to the campaign. The dashed line shows that selecting a percent of customers at random results in an equivalent percent of responders. For example, randomly contacting 20% of the customers nets 20% of responders.

On the other hand, the solid red line shows the benefit of using the Response Predictor metric. Using this predictor, the most likely responders can be targeted first, providing a lift to the campaign's results. In other words, using the Response Predictor to identify the customers most likely to respond favorably, contacting 20% of the customers can yield over 80% of the responders, a gain of over four times the random approach. Marketing teams use lift charts like this to define programs that optimize the costs and the returns of their campaigns.

Finally, you can use the metric to predict the responses of the customers who were not used in developing the model, and who were acquired on or after October first.

The following report shows that out of the new customers, less than 20% are likely to respond positively to a campaign similar to that of the fall campaign. Based on these results, the number of customers targeted can be greatly reduced from all new customers to only those predicted to respond positively

to the campaign. Targeting the campaign in this way will decrease costs while likely increasing positive responses significantly.

Customer	Metrics	Response Predictor (Imported)	Response Propensity (Imported)
Total		1,861	19%
Count		9877	9877
Aaby	Alen	0	0%
Aadland	Miko	1	98%
Aadland	Warner	0	0%
Aadland	Constant	0	0%
Aafedt	Wendy	0	0%
Aagesen	Bink	0	0%
Aalgaard	Kenney	0	0%
Aamodt	Stacy	1	99%
Aarestad	Benjamin	0	0%
Aarnink	Marlan	0	0%
Aaron	Ferrell	0	0%
Aaronson	Maxwell	0	0%
Aasen	Beatrice	1	99%
Aba	Blain	0	0%
Aba-Bulqu	Leslie	1	99%
Abad	Geoffrey	0	0%
Abad	Bekir	0	0%

◀ ◀ 1 ▾ 2 3 4 5 of 50 pages ▶ ▶

Market basket analysis (using association rules analysis)

Market basket analysis studies retail purchases to determine which items tend to appear together in individual transactions. Retailers use market basket analysis for their commercial websites to suggest additional items to purchase before a customer completes their order.

An example of using market basket analysis for grocery store transactions is described in [Association rules analysis, page 793](#).

The MicroStrategy Tutorial project includes an example of a market basket analysis scenario. The example analyzes customer orders of movies, and recommends other movies based on their selections. These recommendations are based on the other customer orders and are meant to provide customers with additional movie options that they are most likely to

add to their order. The resulting movie recommendations are provided in the report 3 -- Movie Recommendations shown below.

Order	Item	Metrics	Recommendation (Highest Confidence)	Recommendation (2nd Highest Confidence)	Recommendation (3rd Highest Confidence)
10438	Nutrition 101		Christine	L.A. Confidential	Mulan
	The African Queen		Christine	L.A. Confidential	Mulan
	Dracula		Christine	L.A. Confidential	Mulan
	And Out Come Wolves		Christine	L.A. Confidential	Mulan
	Best of Disney Volume 2		Christine	L.A. Confidential	Mulan
	98 Degrees & Rising		Christine	L.A. Confidential	Mulan
	Never Say Never		Christine	L.A. Confidential	Mulan
11594	L.A. Confidential		Wall Street	Christine	Dr. Dolittle
11696	50 Favorite Rooms		I Know What You Did Last Summer	The Wedding Singer	9 Steps to Financial Freedom
	Ferris Bueller's Day Off		I Know What You Did Last Summer	The Wedding Singer	9 Steps to Financial Freedom
27540	Blade		Massage for Health	Pulp Fiction	The Scarlet Pimpernel
	Pretty Woman		Massage for Health	Pulp Fiction	The Scarlet Pimpernel
	Hope Floats		Massage for Health	Pulp Fiction	The Scarlet Pimpernel
	Le Mans		L.A. Confidential	Pretty Woman	The Lost Boys
30209	Alice in Wonderland		L.A. Confidential	Pretty Woman	The Lost Boys
	Cracked Rearview		L.A. Confidential	Pretty Woman	The Lost Boys



The example Tutorial project includes reports, metrics, and other objects created for this market basket analysis example (search the project for “Association Rules”). You can use the objects in the Tutorial project to step through the example and determine how it can be applied to your reporting environment.

The steps below show you how to create similar market basket analysis of movie purchases, using data mining services techniques, in the Tutorial project.

To create a training metric for market basket analysis

The steps below can be used to create the training metric `Training Metric for 'Movie Recommendation'`, which is provided with the Tutorial project.

- 1 In MicroStrategy Developer, from the **Tools** menu, select **Training Metric Wizard**. The Training Metric Wizard opens on the Introduction page.
- 2 Click **Next**. The Select Type of Analysis page opens.
- 3 Select **Association** as the type of analysis. The following additional options are displayed and should be defined as explained below:

- **Maximum number of items per antecedent:** Type 3 for this option.
- **Minimum confidence:** Type .1 for this option.
- **Minimum support:** Type .1 for this option.

4 Click **Next**. The Select Metrics page opens.

5 Browse to the Order metric, which is stored in Tutorial\Public Objects\Reports\MicroStrategy Platform Capabilities\MicroStrategy Data Mining Services\Support Objects. Select the **Order** metric and click the arrow next to **Transaction metric**.



The Order metric's definition is `Max<FactID=Revenue>(Order){~}`, which is based on the Order attribute. The parameter `<FactID=Revenue>` is included to ensure the attribute information can be joined together and displayed on a report.

6 In the same folder, select the Item metric and click the arrow next to **Item metric**.



The Item metric's definition is `Max<FactID=Revenue>(Item@DESC){~}`, which is based on the Item attribute. The parameter `<FactID=Revenue>` is included to ensure the attribute information can be joined together and displayed on a report.

7 Click **Next**. The Select Output page opens.

8 Click ... (browse button) for the **Folder** option to select a folder to save the training metric to.

9 Clear the **Include extended statistical analysis with the model** check box.

10 Click **Rules**. The Rules to Return dialog box opens.

11 For the **Return the top ranked rules up to this amount** option, change the value to 3, which returns the top three movie recommendations based on the movies in a customer's order.

12 Click **OK** to return to the Select Output page.

13 In the **Predictive metric(s) to generate** area, make the following changes:

- a Keep the **Rule** predictor type selected. For the **Name** field, type `Movie Recommendation Rule`.
- b Select the **Consequent** predictor type. For the **Name** field, type `Recommendation`.

14 Click **Next**. The summary page opens.

15 Click **Finish** to create the training metric. The Save As dialog box opens.

16 In the **Object name** field, type `Training Metric for 'Movie Recommendation'`, and click **Save**.

After creating a training metric, you must create a report that includes the new training metric.

In addition to the training metric, this report must include the Order and Item attributes, which display the items that were included in each order. A simple filter of `Category = Movies` is added to ensure that only movies are included in the report.

Additionally, since there are a large number of orders, another filter is required to reduce the result set. This ensures that the report can be executed in a reasonable amount of time, while also providing enough data to provide relevant market basket analysis of the order information. The report 2 -- `Movie Basket Analysis Training`, available with the MicroStrategy

Tutorial project, includes a filter that prompts the user for which orders to return. In the results shown below, every seventh order is displayed.

Report details			
Report Filter: (Category = Movies) And {n-th Order Sample}			
Order	Item	Metrics	Training Metric for 'Movie Recommendation'
10003	Dr. Dolittle		Manhunter
	Pulp Fiction		Manhunter
	Wall Street		Manhunter
	Let's Play Soccer		Manhunter
10010	Scream 2		Everest
	The Omen		Everest
	The Shining		Everest
	Christine		Everest
10024	The Scarlet Pimpernel		Sesame Street
10031	Scream 2		
	The Omen		
	The Shining		
	Caddyshack		Sense and Sensibility
10038	Pulp Fiction		Sense and Sensibility
	The Muppet Movie		Sense and Sensibility
10045	Pulp Fiction		Christine

The steps below show you how to create a similar training report in the Tutorial project.

To create a training report for market basket analysis

- 1 In MicroStrategy Developer, from the **File** menu, point to **New**, and select **Report**. The New Grid dialog box opens.
- 2 From the **General** tab, select **Blank Report**, and click **OK**. The Report Editor opens.
- 3 Add the Order and Item attributes to the rows of the report.
- 4 Add the training metric you created using the steps [To create a training metric for market basket analysis, page 849](#) to the columns of the report.

- 5 In the **Report Filter** pane, drag and drop the **Category** attribute to begin creating a new filter qualification. Options to create an attribute qualification are displayed.
- 6 Click **Add**. The Select Objects dialog box opens.
- 7 In the **Available objects** pane, select **Movies**, and click the arrow to move it to the Selected objects pane.
- 8 Click **OK**. You are returned to the attribute qualification options.
- 9 Click **OK** to create the filter qualification.

To limit the number of orders included in the market basket analysis

Next you need to create a filter to limit the number of orders returned for the report. The example report 2 -- `Movie Basket Analysis Training` uses the filter `n-th Order Sample`. You can add this filter to limit the number of orders returned, or you can use the steps below to create a simple filter that includes a subset of elements from the `Order` attribute. The rest of this example assumes that you create the simple filter detailed below.

- 10 In the **Report Filter** pane, drag and drop the **Order** attribute to begin creating a new filter qualification. Options to create an attribute qualification are displayed.
- 11 Click **Add**. The Select Objects dialog box opens.
- 12 In the **Available objects** pane, select `Order` elements from ID 10,000 to 12,000, and click the arrow to move the elements to the Selected objects pane. You can only select 1,000 elements at a time, so you need to do multiple selections to include all these elements in the Selected objects pane.
- 13 Click **OK**. You are returned to the attribute qualification options.
- 14 Click **OK** to create the filter qualification.

To save the report and create the predictive analysis metrics

- 15 From the **File** menu, click **Save As**. The Save Report As dialog box opens.
- 16 In the **Object name** field, type `Market Basket Analysis Report`, and click **Save**. You are returned to the report.

17 From the toolbar, click the **Run Report** icon. The report execution begins.

Depending on your system resources, the report execution can take a considerable amount of time, as various data mining calculations are being performed to return the market basket analysis. Once the report execution is complete, a message is displayed that verifies that predictive metrics were created.

The following predictive metrics are created by executing the report:

- Three predictive metrics are created to explain the market basket rules. These show the movie items included in an order, and the associated movie items that are recommended based on order item. These metrics show the top three recommendations from highest confidence to lowest confidence:
 - `Movie Recommendation Rule(confidence,1)`
 - `Movie Recommendation Rule(confidence,2)`
 - `Movie Recommendation Rule(confidence,3)`
- Three predictive metrics are created to list only the recommended movie, based on the movie items included in an order. These metrics show the top three recommendations from highest confidence to lowest confidence:
 - `Recommendation(confidence,1)`
 - `Recommendation(confidence,2)`
 - `Recommendation(confidence,3)`

To view these market basket analysis recommendations, you can include them on a report. You can use the same report you created using the steps [To create a training report for market basket analysis, page 852](#).

The steps below show you how to create a report in the Tutorial project, similar to the example report 3 -- Movie Recommendations.

To view the market basket analysis predictive metrics

- 1 Open the report you created using the steps [To create a training report for market basket analysis, page 852](#).

- 2 Right-click the training metric `Training Metric for 'Movie Recommendation'` and select **Remove From Report**.
- 3 Double-click the filter qualification on `Order` to modify it. Remove the `Order` elements with IDs 10,501 to 12,000. This should leave just the `Order` elements with IDs 10,000 to 10,500 in the filter qualification. Click **OK** to accept the changes and return to the report.



This step is done to reduce the number of results to return and analyze. Returning fewer results allows the report to return results quicker while still providing market basket analysis. You can include other subsets of `Order` elements to evaluate other orders.

- 4 From the toolbar, click the **Run Report** icon. The report is executed.

The report displays both the rules that determine which movies are recommended, as well as just the recommendations themselves. To make the report easier to read, you can remove the `Movie Recommendation Rule(confidence,1)`, `Movie Recommendation Rule(confidence,2)`, and `Movie Recommendation Rule(confidence,3)` metrics from the report. This leaves just the metrics that display the three movies recommended to a customer to add to their order, as shown in the report below.

Order	Item	Metrics	Recommendation(confidence,1)	Recommendation(confidence,2)	Recommendation(confidence,3)
10001	The Horse Whisperer		Pretty Woman	Dracula	The Year Without a Santa Claus
	Mysteries of the Bible		Pretty Woman	Dracula	The Year Without a Santa Claus
	Learn! Microsoft Office 2000		Pretty Woman	Dracula	The Year Without a Santa Claus
	Living Yoga		Pretty Woman	Dracula	The Year Without a Santa Claus
10002	The Princess Bride		Blue's Clues	Lethal Weapon 4	Psycho
10003	Dr. Dolittle		Sesame Street	The Jungle Book	The Omen
	Pulp Fiction		Sesame Street	The Jungle Book	The Omen
	Wall Street		Sesame Street	The Jungle Book	The Omen
	Let's Play Soccer		Sesame Street	The Jungle Book	The Omen
10004	The Object Of My Affection		Apollo 13	Scream 2	Jesus of Nazareth
	Pretty Woman		Apollo 13	Scream 2	Jesus of Nazareth
10006	The Avengers		Blue's Clues	Lethal Weapon 4	Psycho
	The Princess Bride		Blue's Clues	Lethal Weapon 4	Psycho
10009	Blue's Clues		AAA Travel Video Series	AAA Travel Video Series	Mulan
	9 Steps to Financial Freedom		AAA Travel Video Series	AAA Travel Video Series	Mulan

The report shown above shows a simple display of movie recommendations based on the order of other movie items. These market basket techniques can be used to make recommendations to customers about any items they are likely to add to their order at purchase time.

The steps below show you how to include a prompted filter on this report so that an order or a group of orders can be selected for on the fly recommendations.

To include a prompt on the report to provide on the fly recommendations

- 1 Once you have viewed the results of the report created using the steps [To view the market basket analysis predictive metrics, page 854](#), from the **View** menu in the Report Editor, select **Design View**.
- 2 In the **Report Filter** pane, double-click the filter qualification on the Order attribute elements. The details for the attribute qualification are displayed.
- 3 Click **Prompt**. The Prompt Generation Wizard opens.
- 4 In the **Attribute** field, the Order attribute should already be selected. Click **Next**.
- 5 Keep the default option of **List all elements (no restriction)** selected and click **Next**.
- 6 Within the Prompt restrictions area, make the following changes:
 - a Select the **Minimum number of answer** check box. Increase the value to 1.
 - b Select the **Maximum number of answers** check box. Increase the value to the number of orders you want to limit the report to. For example, you can set the maximum to 10, so that the report displays only 10 orders at a time.
 - c Select the **Prompt answer required** check box.Once you have made these changes, click **Next**.
- 7 In the **Select default prompt answers** area, click **Clear**. This will require the person using the report to decide what orders they want to return movie recommendations for, rather than simply returning recommendations for some default orders.
- 8 Click **Finish**. The Prompt Generation Wizard is closed and you are returned to the attribute qualification options.
- 9 Click **OK** to accept the changes to the filter qualification. You are returned to the report, displayed in Design View.
- 10 From the toolbar, click the **Run Report** icon. The prompt is displayed.

- 11** Make a selection of orders to return recommendations for based on the earlier market basket analysis. For example, select orders 13,000 through 13,009 and click the arrow icon to include those orders in the report.
- 12** Click **Finish**. The report is executed, which displays movie recommendations for the selected orders, as shown below.

Order	Item	Metrics	Recommendation(confidence,1)	Recommendation(confidence,2)	Recommendation(confidence,3)
13000	My Cousin Vinny		AAA Travel Video Series	Barney	The Jungle Book
	City Of Angels		AAA Travel Video Series	Barney	The Jungle Book
	The Hustler		AAA Travel Video Series	Barney	The Jungle Book
13001	The Hustler		AAA Travel Video Series	Barney	9 Steps to Financial Freedom
13002	As Good As It Gets		Babe	Sense and Sensibility	Manhunter
13004	Lethal Weapon 4		Dracula	The Jungle Book	Pulp Fiction
	Dr. Dolittle		Dracula	The Jungle Book	Pulp Fiction
	Classic Rock Guitar		Dracula	The Jungle Book	Pulp Fiction
13005	Titanic		The Omen	Alice in Wonderland	Scream 2
	The Parent Trap		The Omen	Alice in Wonderland	Scream 2
	The Secrets of Happiness		The Omen	Alice in Wonderland	Scream 2
13006	Dr. Dolittle		The Omen	The Omen	The Omen
	Great Expectations		The Omen	The Omen	The Omen
	Scream 2		The Omen	The Omen	The Omen
	The Shining		The Omen	The Omen	The Omen
13007	Vanishing Point		The Mask of Zorro	Hope Floats	The Year Without a Santa Claus
	Godzilla		The Mask of Zorro	Hope Floats	The Year Without a Santa Claus
	The Full Monty		The Mask of Zorro	Hope Floats	The Year Without a Santa Claus
	Golf My Way Collection		The Mask of Zorro	Hope Floats	The Year Without a Santa Claus
	Tae-Bo Workout		The Mask of Zorro	Hope Floats	The Year Without a Santa Claus
13008	The Little Mermaid		Swingers	Classic Rock Guitar	Small Soldiers
13009	Great Expectations		AAA Travel Video Series	The Omen	Sense and Sensibility
	The Hustler		AAA Travel Video Series	The Omen	Sense and Sensibility
	Mulan		AAA Travel Video Series	The Omen	Sense and Sensibility
	The Secrets of Happiness		AAA Travel Video Series	The Omen	Sense and Sensibility

This simple prompted report can be the beginning of a recommendation engine that provides recommendations to customers based on market basket analysis techniques.

VLDB PROPERTIES

Introduction

VLDB properties allow you to customize the SQL that MicroStrategy generates. These settings affect how MicroStrategy Intelligence Server manages joins, metric calculations, and query optimizations, among other options. This chapter focuses on the VLDB properties that are set at the metric and report level. This chapter includes the following sections:

- [*Levels of VLDB properties, page 860*](#)
- [*VLDB properties at the metric level, page 861*](#)
- [*VLDB properties at the report level, page 865*](#)
- [*VLDB properties at the database instance level, page 882*](#)

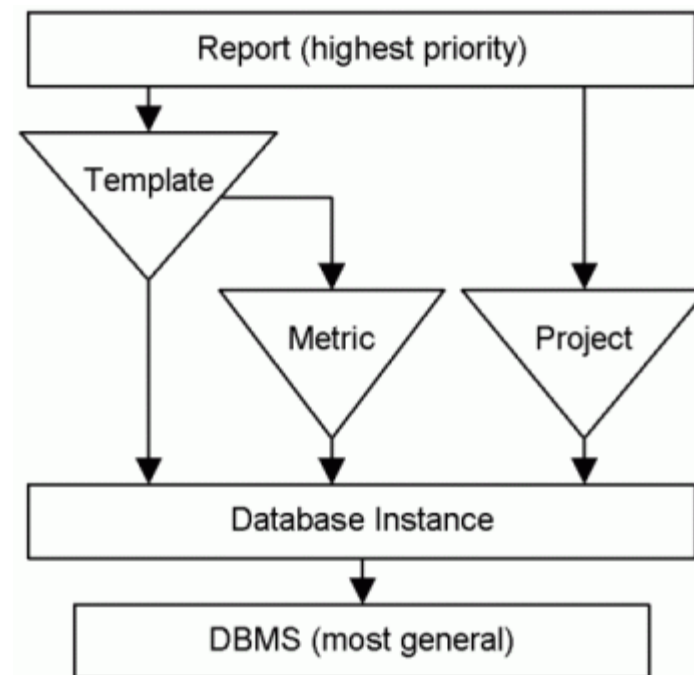


For more information on VLDB properties set at other levels and the VLDB Properties Editor, refer to the *MicroStrategy Supplemental Reference for System Administration*.

Levels of VLDB properties

VLDB properties are available at multiple levels, so that SQL generated for one report can be manipulated separately from the SQL generated for another, similar report. This flexibility is an important benefit of VLDB properties.

The hierarchy, or order of precedence, for VLDB properties is outlined in the following figure.



The arrows depict the override authority of the levels, with the report level having the greatest authority. For example, if a VLDB property is set one way for a report, and the same property is set differently for a metric included on the report, the report property takes precedence.

Conversely, if the VLDB property at the report level is set to use the default inherited value, the project level setting is used. If that level is also set to the default or if the VLDB property is not set at the project level, the setting at the database instance is used. This information is available for each property in the VLDB Properties dialog box at each level. The Use default inherited value option indicates the level that is active, while the SQL preview box displays a description of the selected setting.

The VLDB properties at the different levels work together to affect report results. However, you set the properties of each level in a different interface, so the sections of this chapter are grouped at the different levels.

VLDB properties at the metric level

Although there are other VLDB properties that pertain specifically to metrics, only six metric VLDB properties are available at the individual metric level. They are the following:

- [Integer Constant In Metric, page 861](#)
- [Metric Join Type, page 862](#)
- [Null Check, page 862](#)
- [Zero Check, page 863](#)
- [Null checking for Analytical Engine, page 863](#)
- [Subtotal Dimensionality Aware, page 863](#)



The other metric properties are described in the *MicroStrategy Supplemental Reference for System Administration*.

To access metric VLDB properties

- 1 In the Metric Editor, from the Tools menu select **Advanced Settings**.
- 2 Choose **VLDB Properties** to access the VLDB Properties (Metric) dialog box. The Null Checking for Analytical Engine and Subtotal Dimensionality Aware settings are contained in the Analytical Engine folder, while the others are found in the Metrics folder.

Integer Constant In Metric

This setting determines whether to add a “.o” after an integer (a whole number). This prevents imprecise rounding when division is performed. For example, 2/7 returns the integer 0 if “.o” has not been added. Generally, the “.o” is added, so that the same calculation returns a more accurate result, for example, 2.o/7.o = 0.3. The use of “.o” is usual.

Some databases have trouble with this change because certain database functions only work with integer data types. Therefore this setting is offered so you can disable the addition of the “.o”.

The options for this property are as follows:

- Add “.o” to integer constants in metric expressions
- Do not add “.o” to integer constants in metric expressions
- Use default inherited value

Metric Join Type

This property sets the type of join used in the metric. The options are

- Inner join, which includes only rows that have data present for all metrics across that row in a report
- Outer join, which shows all rows for the metric, regardless of other metrics on the report
- Use default inherited value

For more information on join types, see [Join specifications, page 121](#).

Null Check

The Null Check property indicates how to handle arithmetic operations that contain null values. When the null check is performed, null values are changed to zero in arithmetic calculations (+, -, *, and /).

The options for this property are as follows:

- Do nothing, which means that the database rather than the Analytical Engine handles arithmetic calculations including a null value
- Check in all queries
- Check in temporary table join only
- Use default inherited value



This property can also be set at the report level, which overwrites the setting at the metric level.

Zero Check

The Zero Check property indicates how to handle division by zero and when to check for zeros in the denominator during division. When a zero check is performed, a zero in the denominator of a division calculation is changed to a null value.

The options for this property are as follows:

- Do nothing, which means that the database rather than the Analytical Engine handles division by a zero
- Check in all queries
- Check in temporary table join only
- Use default inherited value



This property can also be set at the report level, which overwrites the setting at the metric level.

Null checking for Analytical Engine

This setting determines whether a null value is interpreted as zero when the Analytical Engine performs calculations. It is found in the Analytical Engine folder of the VLDB Properties (Metric) dialog box.

The options for this property are as follows:

- False, which means that null values are not altered
- True, which means the Analytical Engine converts null values to zeros
- Use default inherited value



You can set replacement text for null values at the report level. For more information, see the *Report Formatting* chapter in the *Basic Reporting Guide*.

Subtotal Dimensionality Aware

The Subtotal Dimensionality Aware setting enables subtotaling based on the dimensionality of a metric, also known as its level. How it works depends on another VLDB property, Query Population Dimensionality Aware, which

handles backwards compatibility with MicroStrategy 7.1. These settings work together as illustrated in the following table:

If Query Population Is:	Then Subtotal Is:
TRUE	TRUE or FALSE
FALSE	Ignored, meaning that subtotals are never aware of dimensionality

The default setting for the Subtotal Dimensionality Aware property is TRUE, so subtotals depend on the metric's level. If you must subtotal without using the level of the metric, set this property to FALSE.

This property is found in the Analytical Engine folder of the VLDB Properties (Metric) dialog box. The options for this property are:

- FALSE, which means that subtotals do not take into account the level/dimensionality of the metric
- TRUE, which means that subtotaling is aware of metric level/dimensionality
- Use default inherited value

For example, the Quarterly Revenue metric is defined as `Sum (Revenue) Dimensionality = Quarter`, and the Yearly Revenue metric is defined as `Sum (Revenue) Dimensionality = Year`.

Year	Quarter	Quarterly Revenue	Yearly Revenue
1	1	\$100	\$600
1	2	\$200	\$600
1	3	\$100	\$600
1	4	\$200	\$600

If Subtotal Dimensionality Aware is set to FALSE, the quarterly subtotal is calculated as \$600, the total of the Quarterly Revenue values. The yearly subtotal is calculated as \$2400, the total of the Yearly Revenue values. This is how MicroStrategy 7.1 calculated the subtotal.

If Subtotal Dimensionality Aware is set to TRUE, the quarterly subtotal is still \$600. MicroStrategy is aware of the level of the Yearly Revenue metric, so rather than simply adding the column values, it calculates the total as \$600.

VLDB properties at the report level

The report level contains numerous VLDB properties. The most commonly used are listed below. The top level of the list indicates the folder in the VLDB Properties dialog box in which the property is found.

- Joins:

- [Cartesian Join Warning, page 866](#)
- [Full Outer Join Support, page 868](#)
- [Preserve All Final Pass Result Elements, page 868](#)
- [Preserve All Lookup Table Elements, page 869](#)



The term “cross join” is used interchangeably with Cartesian join in these sections.

- Metrics:

- Null Check
- Zero Check



The metric properties above can also be set at the metric level, although the setting at the report level overwrites it. These properties are described in [VLDB properties at the metric level, page 861](#).

- Pre/Post Statements:

- [Report Post Statement, page 872](#)
- [Report Pre Statement, page 872](#)
- [Table Post Statement, page 873](#)

- Query Optimizations:

- [Sub Query Type, page 873](#)
- [WHERE Clause Driving Table, page 874](#)

- Select/Insert:

- [Attribute Form Selection Option for Intermediate Pass, page 876](#)
- [Attribute Selection Option for Intermediate Pass, page 876](#)
- [GROUP BY ID Attribute, page 877](#)

- [GROUP BY Non-ID Attribute, page 878](#)
- [SQL Hint, page 878](#)
- [UNION Multiple INSERT, page 878](#)
- Tables:
 - [Drop Temp Table Method, page 878](#)
 - [FallBack Table Type, page 879](#)
 - [Intermediate Table Type, page 879](#)
 - [Table Option, page 881](#)
 - [Table Space, page 881](#)



For information on other report level VLDB properties, refer to the *MicroStrategy Supplemental Reference for System Administration*.

Accessing report VLDB properties

To set these properties, open the report in the Report Editor or Report Viewer. Select **VLDB Properties** from the Data menu to access the VLDB Properties (Report) dialog box. The properties are saved in different folders, as indicated in the previous list.

The following settings are advanced properties which are hidden by default:

- Full Outer Join Support
- Preserve All Final Pass Result Elements
- GROUP BY ID Attribute

To access them, select **Show Advanced Settings** from the Tools menu in the VLDB Properties (Report) dialog box.

Cartesian Join Warning

The Cartesian Join Warning indicates the action that occurs when the Analytical Engine generates a report that contains a Cartesian join. In a Cartesian join, every row in one table is joined to every row in the other table. The query returns every possible row combination of the two tables, so a Cartesian join is usually costly to perform. Since a Cartesian join of two

warehouse tables is much more costly than a Cartesian join of two intermediate tables, additional options are included to govern these cases.

The options for this property are

- **Execute**

When a Cartesian join occurs, the report is executed without any warning.

- **Cancel execution**

When a Cartesian join occurs, report execution is canceled without any warning.

- **Cancel execution only when warehouse table is involved in either side of Cartesian join**

A Cartesian join is allowed only when all tables involved in it are intermediate tables. No warning is sent.

- **If only one side of Cartesian join contains warehouse tables, SQL will be executed without warning**

A Cartesian join is allowed if only one warehouse table is involved, otherwise it is canceled without warning.

- **Use default inherited value**



Some Cartesian joins are not a direct table-to-table join. If one join Cartesian joins to another join, and one of the joins contains a warehouse table (not an intermediate table), then the execution is either canceled or allowed depending on the option selected. For example, when a Cartesian join is performed between the join of two intermediate tables (TT_A join TT_B) and the join of an intermediate table and a warehouse table (TT_C join WH_D), the following happens:

- When **Cancel execution only when warehouse table is involved in either side of Cartesian join** is selected, the report is canceled. In the above example, execution is canceled because a warehouse table is used, even though TT_A, TT_B, and TT_C are all intermediate tables.
- When **If only one side of Cartesian** is selected, the report runs without warning. In the above example, execution continues because a warehouse table (WH_D) is used on only one side of the join.

Full Outer Join Support

The Full Outer Join Support setting specifies if the database platform supports full outer join syntax.

The options for this property are

- No support
- Support, which assumes that the Join Type VLDB setting is Join 92; any other value in Join Type is ignored
- Use default inherited value



Full Outer Join Support is an advanced property that is hidden by default.

Preserve All Final Pass Result Elements

The Preserve All Final Pass Result Elements setting determines how to join on the final result and the lookup and relationship tables.



Preserve All Final Pass Result Elements is an advanced property that is hidden by default.

The options for this property are:

- Preserve common elements of final pass result table and lookup table

The SQL Engine generates an equi-join, which is a join which combines the table rows. These are added to the results only when the joined columns have equal values.

Therefore, the report contains only the elements that are common to both tables.

- Preserve all final result pass elements

The SQL Engine generates an outer join from the fact table to the lookup table, as well as to any relationship tables. This occurs because it is hard to distinguish which table is used as a lookup table and which table is used as a relationship table. For example, LOOKUP_DAY serves as a lookup table for the Day attribute and a relationship table for Day and Month.

The report contains all of the elements that are in the final result set.



This setting should not be used in standard data warehouses, where the lookup tables are properly maintained and all elements in the fact table have entries in the respective lookup table. It should be used only when a certain attribute in the fact table contains more (unique) attribute elements than its corresponding lookup table. The lookup table, by definition, should contain all the attribute elements, but this scenario could occur if the fact tables are updated more often than the lookup tables.

- Preserve all elements of final pass result table with respect to lookup table but not relationship table

The SQL Engine generates an inner join on all passes except the final pass, which uses an outer join.

- Do not listen to per report level setting, preserve elements of final pass according to the setting at attribute level. If this choice is selected at attribute level, it will be treated as preserve common elements (i.e. choice 1) option

If this option is selected at the database instance or template level, the setting for this VLDB property at the attribute level is used. This value should not be selected at the attribute level. However, if you choose this setting at the attribute level, the VLDB property is set to the Preserve common elements of final pass result table and lookup table option.

This setting is useful if you have only a few attributes that require different join types. For example, if among the attributes in a report only one needs to preserve elements from the final pass table, you can set the VLDB property to Preserve all final pass result elements setting for that one attribute. You can then set the report to the Do not listen setting for the VLDB property. When the report is run, only the attribute set differently causes an outer join in SQL. All other attribute lookup tables will be joined using an equal join, which leads to better SQL performance.

- Use default inherited value

For examples of these settings in use, see the *MicroStrategy Supplemental Reference for System Administration*.

Preserve All Lookup Table Elements

The Preserve All Lookup Table Elements setting is used to show all attribute elements that exist in the lookup table, even though there is no

corresponding fact in the result set. For example, your report contains Store and Sum(Sales). You need to show all the stores in the report, even those stores that do not have sales. Instead of relying on the stores in the sales fact table, you must ensure that all the stores from the lookup table are included. The SQL Engine must use a left outer join from the lookup table to the fact table.

The report can include multiple attributes. To keep all of them, the Analytical Engine must use a Cartesian join between the attributes' lookup tables before doing a left outer join to the fact table.

The options for this property are:

- Preserve common elements of lookup and final pass result table

The Analytical Engine does a normal join (equi-join) to the lookup table.

- Preserve lookup table elements joined to final pass result table based on fact keys

If the fact table level is not the same as the report level, this option first uses a left outer join to keep all the attribute elements at the fact table level. Next, it aggregates to the report level. The advantage of this approach is the same pass of SQL performs the left outer join and aggregation. The disadvantage is that the Cartesian join with the lookup tables is at a much lower level and can result in a very large joined table.

- Preserve lookup table elements joined to final pass result table based on template attributes without filter

If the fact table level is not the same as the report level, this option first aggregates to the report level, then uses a left outer join to take in all the attribute elements. This is the reverse of the approach described above. It needs an additional pass, but the cross join table is usually smaller.

- Preserve lookup table elements joined to final pass result table based on template attributes with filter

This option is similar to the previous option, the difference being that the report filter is applied in the final pass.

- Use default inherited value



The various “Preserve lookup table elements” options are described in more detail in [Example: Different levels for fact table and report, page 871](#).

When you use any of the “Preserve lookup table elements” options, it is assumed you want to keep all the elements of the attributes in their lookup

tables. However, you may want the setting to affect only some of the attributes on a template. For example, a report contains Store, Month, and the Sum(Sales) metric. You want to show all the store names, even if they do not have sales, but not necessarily all the months in the LOOKUP_MONTH table. You can individually select attributes on the report that need to preserve elements with the Attribute Join Type setting in Report Data Options. For more information, see [Selecting an attribute join type, page 339](#).

Example: Different levels for fact table and report

The fact table level is not always the same as the report level. For example, a report contains Store, Month, and the Sum(Sales) metric, but the fact table is at the level of Store, Day, and Item. You can choose any of the “Preserve lookup table elements” options to accomplish this, as described below.

Preserve lookup table elements joined to final pass result table based on fact table keys

First, a left outer join keeps all the attribute elements at the Store, Day, and Item level. Then the results are aggregated to the Store and Month level. Two SQL passes are run:

- **Pass 1:** LOOKUP_STORE cross join with LOOKUP_DAY cross join LOOKUP_ITEM produces TempTable1
- **Pass 2:** TempTable1 left outer join with Fact_Table on (Store, Day, Item)

The advantage is that the left outer join and aggregation is completed in the same pass (Pass 2). The disadvantage is that the Cartesian join with the lookup tables is performed at a much lower level (Pass 1) so the result of the Cartesian joined table (TempTable1) can be very large.

Preserve lookup table elements joined to final pass result table based on template attributes without filter

The aggregation is completed first, then a left outer join brings in all the attribute elements. Three SQL passes are run:

- **Pass 1:** Aggregates the Fact_Table to TempTable1 at Store and Month. This is actually the final pass of a normal report without turning on this setting.
- **Pass 2:** LOOKUP_STORE cross join with LOOKUP_MONTH produces TempTable2

- **Pass 3:** TempTable2 left outer join with TempTable1 on (Store, Month)

This approach needs one more pass than the previous option, but the Cartesian join table (TempTable2) is usually smaller.

Preserve lookup table elements joined to final pass result table based on template attributes with filter

This option is similar to the previous option. The only difference is that the report filter is applied in the final pass (Pass 3). For example, a report contains Store, Month, the Sum(Sales) metric, and a filter of Year = 2002. You want to display every store in every month in 2002, regardless of whether sales occurred. However, you do not want to show months from any other years. This setting resolves the issue.

Report Post Statement

The Report Post Statement settings insert custom SQL statements after the final SELECT statement but before the DROP statements. The settings are numbered 1-5. Each text string entered in Report Post Statement 1 through Report Post Statement 4 is executed separately as a single statement. To execute more than five statements, insert multiple statements in Report Post Statement 5, separating each statement with a semicolon (;). The SQL Engine breaks them into individual statements at the semicolons and executes each separately. The custom SQL is applied to every intermediate table or view.



These settings are applicable when the Intermediate Table Type property is set to Permanent table, True temporary table, or Temporary view. For a description of that property, see [Intermediate Table Type, page 879](#).

Report Pre Statement

The Report Pre Statement settings insert custom SQL statements at the beginning of the report SQL. The settings are numbered 1-5. Each text string entered in Report Pre Statement 1 through Report Pre Statement 4 is executed separately as a single statement. To execute more than five statements, insert multiple statements in Report Pre Statement 5, separating each statement with a semicolon (;). The SQL Engine breaks them into individual statements at the semicolons and executes each separately.

Table Post Statement

The Table Post Statement settings insert custom SQL statements after the CREATE TABLE and INSERT INTO statement. The settings are numbered 1-5. Each text string entered in Table Post Statement 1 through Table Post Statement 4 is executed separately as a single statement. To execute more than five statements, insert multiple statements in Table Post Statement 5, separating each statement with a semicolon (;). The SQL Engine breaks them into individual statements at the semicolons and executes each separately. The custom SQL is applied to every intermediate table or view.



These settings are applicable when the Intermediate Table Type property is set to Permanent table, True temporary table, or Temporary view. For a description of that property, see [Intermediate Table Type, page 879](#).

Sub Query Type

Different databases have different syntax support for sub queries. The Analytical Engine checks the Sub Query Type property for the type of syntax to use when generating a sub query.

It is more efficient to select the needed column rather than selecting every column. It is also more efficient to use the IN clause rather than using the Exists function. The most optimal option depends on the capability of your database to support different sub query syntax.

The default setting is **Where col1 in (Select s1.col1...) fall back to Exists (Select *...) for multiple column IN**. However, based on your database type, the default to a more optimal setting is automatically changed. See below for database platform exceptions to the default setting.

Database exceptions to the default setting

Database	Default
Oracle	Where col1, col2 in (Select s1.col1, s1.col2...)
DB2 UDB	Use Temporary Table
DB2 UDB for OS/390	Where Exists (Select *...)
RedBrick	Where col1 in (Select s1.col1...) fall back to Exists (Select col1, col2...) for multiple column in

Notice that most options have a fallback action. In some scenarios, the selected option does not work, so the SQL Engine must fall back to an approach that always works. The typical scenario for falling back is when multiple columns are needed in the IN list, but the database does not support it and the correlated sub queries.

See the *MicroStrategy Supplemental Reference for System Administration* for examples.

WHERE Clause Driving Table

The WHERE Clause Driving Table setting tells the Analytical Engine the type of column that is preferred in the qualification of a WHERE clause. One SQL pass usually joins fact tables and lookup tables on certain ID columns. When a qualification is defined on such a column, the Analytical Engine can use the column in either the fact table or the lookup table. In certain databases like Teradata and RedBrick, a qualification on the lookup table can achieve better performance. The Analytical Engine tries to pick the column from the lookup table when the WHERE Clause Driving Table property is set to use the lookup table.

If this property is set to lookup table, but no lookup table exists in the FROM clause for the column being qualified on, the Analytical Engine does not add the lookup table to the FROM clause. To ensure that the qualification uses a lookup table column, set the DSS Star Join setting to use a partial star join.

For more information on the DSS Star Join setting, see the *MicroStrategy Supplemental Reference for System Administration*.

The options for this property are

- Use lookup table
- Use fact table
- Use default inherited value

Attribute and Attribute Form Selection Option for Intermediate Passes properties



This section describes both the Attribute Selection Option for Intermediate Passes and the Attribute Form Selection Option for Intermediate Passes properties.

Normally, the MicroStrategy SQL Engine selects the minimum number of columns that are needed in each pass. For an intermediate pass, the SQL Engine usually selects only attribute ID forms. The SQL Engine also selects the attributes necessary to make the join, usually key attributes. Then in the final pass, additional attributes or attribute forms that are necessary for report display can be joined.

This algorithm is optimal in most cases, as it minimizes the size of intermediate tables. However, in certain schemas, especially denormalized ones, and schemas that use fact tables as both lookup tables and relationship tables, such an algorithm can cause additional joins in the final pass.

Example

A report contains Region, Store, Metric1, and Metric2. Metric1 uses FactTable1, which contains Store_ID, Store_Desc, Region_ID, Region_Desc, and F1. Metric2 uses FactTable2, which contains Store_ID, Store_Desc, Region_ID, Region_Desc, and F2.

With the normal SQL Engine algorithm, the intermediate pass that calculates Metric1 selects Store_ID and F1. The intermediate pass that calculates Metric2 selects Store_ID and F2. Then the final pass joins these two intermediate tables together. Since Region is in the report, it should join upward to the region level and find the Region_Desc form. This can be done by joining either fact table in the final pass. So either FactTable1 or FactTable2 is accessed twice. If these tables are big, which is usual, the performance can be very slow. On the other hand, if Store_ID, Store_Desc, Region_ID, and Region_Desc are picked up in the intermediate passes, neither FactTable1 or FactTable2 needs to be joined in the final pass, thus boosting performance.

The Attribute Selection Option for Intermediate Pass and Attribute Form Selection Option for Intermediate Pass settings allow the SQL Engine to select additional attributes or attribute forms in the intermediate pass, when they are directly available. When these settings are selected, the SQL Engine does not join additional tables to select more attributes or forms. So for

intermediate passes, the number of tables to be joined is the same as turning the setting off.



Note the following:

- These settings intend to use bigger (wider) intermediate tables to save additional joins in the final pass and exchange space for time.
- These settings work independently. One does not influence the other.
- Each setting has two values. The default behavior is the original algorithm.

Attribute Form Selection Option for Intermediate Pass

This property determines whether the SQL Engine selects the needed attribute forms in the intermediate passes if available. See the previous section, [Attribute and Attribute Form Selection Option for Intermediate Passes properties, page 875](#), for more detailed information and an example.

The options for this property are

- Select ID form only (default algorithm)
- Select ID and other forms if they are on template and available on existing join tree
- Use default inherited value

Attribute Selection Option for Intermediate Pass

This property determines whether the SQL Engine selects additional attributes needed on the report in the intermediate passes. These attributes are in addition to the needed join ID columns and are usually parent attributes. See the previous section, [Attribute and Attribute Form Selection Option for Intermediate Passes properties, page 875](#), for more detailed information and an example.

The options for this property are

- Select only the attributes needed (default algorithm)
- Select other attributes in current join tree if they are on template and their child attributes have already been selected

- Use default inherited value

GROUP BY ID Attribute

This property determines how to group by a selected ID column when an expression is performed on the ID expression. Each of the options is described below. The code fragment following each description replaces the section named *group by ID* in the following sample SQL statement.

```
select a22.STORE_NBR STORE_NBR,  
a22.MARKET_NBR * 10 MARKET_ID,  
sum(a21.REG_SLS_DLR) WJXBFS1  
from STORE_DIVISION a21  
join LOOKUP_STORE a22  
on (a21.STORE_NBR = a22.STORE_NBR)  
where a22.STORE_NBR = 1  
group by a22.STORE_NBR, group by ID
```

The options for this property are

- **Group by expression:** Group by the expression performed in the Select statement on the ID column.

```
a22.MARKET_NBR * 10
```

- **Group by alias:** Group by the expression alias in the Select statement.

```
MARKET_ID
```

- **Group by column:** Group by the column ID, ignoring the expression performed on the ID column.

```
a22.MARKET_NBR
```

- **Group by position:** Group by the physical table position of the ID column.

- **Use default inherited value**



GROUP BY ID Attribute is an advanced property that is hidden by default.

GROUP BY Non-ID Attribute

The GROUP BY Non-ID Attribute property controls whether non-ID attribute forms, such as descriptions, are used in the GROUP BY clause. If you do not want non-ID columns in the GROUP BY, you can use a MAX function when the column is selected so that it is not used in the GROUP BY.

For an example, see the *MicroStrategy Supplemental Reference for System Administration*.

SQL Hint

The SQL Hint property is used for the Oracle SQL Hint pattern. This string is placed after the SELECT in the Select statement. This setting can insert any SQL string that makes sense after the SELECT in a Select statement, but it is added specifically for Oracle SQL hints.

UNION Multiple INSERT

The Union Multiple Insert property allows the Analytical Engine to UNION multiple select statements into a single table before performing an insert. This allows for a single insert statement to be performed rather than separate inserts for each select statement. This is a database-specific setting, and some databases do not support unions.

Drop Temp Table Method

The Drop Temp Table Method property specifies whether the intermediate tables, permanent tables, temporary tables, and views are to be deleted, or dropped, after report execution. Dropping the tables can lock catalog tables and affect performance, so dropping the tables manually in a batch process when the database is less active can result in a performance gain. The trade-off is space on the database server. The tables remain on the database server using space until the database administrator drops them.

Fallback Table Type

All reports can be resolved using permanent or temporary intermediate tables. Reports can also be resolved by generating derived tables, common table expressions, and views, but these means cannot cover all possible scenarios. For example, they cannot be used when the report contains Analytical Engine SQL, partitioning, or certain outer joins. In such a situation, the MicroStrategy SQL Engine needs another solution, which is provided by the Fallback Table Type property. If the Intermediate Table Type property is set to Derived table, Common table expression, or Temporary view and the SQL Engine concludes that the report cannot be resolved using the setting, it uses the Fallback Table Type property to resolve the report.

The options for this property are

- Permanent table
- True temporary table
- Use default inherited value

Intermediate Table Type

The Intermediate Table Type property determines the type of intermediate tables, also referred to as temporary tables, are used to execute the report. All reports can be executed using permanent and temporary tables. Reports can also be resolved by generating derived tables, common table expressions, and views. However, certain scenarios involving partitioning, outer joins, or analytical functions cannot use these alternatives. In this case, the Fallback Table Type property is used to execute the report.



Only UDB DB2 and Oracle 9i support common tables.

This setting can have a major impact on the performance of the report. Permanent tables are usually less optimal. Derived tables and common table expressions usually perform well, but they do not work in all cases and for all databases. True temporary tables also usually perform well, but not all databases support them. The default setting is permanent tables, because it works for all databases in all situations. However, based on your database type, this setting is automatically changed to what is generally the most optimal option for that platform, although other options could prove to be

more optimal on a report-by-report basis. The database-specific settings are noted below.

Database	Default
DB2 UDB OX/390	True Temporary Tables
DB2 UDB	Common Table Expression
Informix IDS and XPs	True Temporary Tables
SQL Server	True Temporary Tables
Oracle	True Temporary Tables
RedBrick	True Temporary Tables
Sybase ASE and IQ	True Temporary Tables
Teradata	Derived Table

The options for this property are:

- Permanent table
- Derived table
- Common table expression
- True temporary table
- Temporary view
- Use default inherited value

For examples of this property in use, see the *MicroStrategy Supplemental Reference for System Administration*.

Table Option and Table Space properties

The Table Option and Table Space properties can be used to customize the CREATE TABLE SQL syntax for any platform. All of these settings are reflected in the SQL statement only if the Intermediate Table Type property is set to Permanent table. For a description of that property, see [Intermediate Table Type, page 879](#).

Customizing a CREATE TABLE statement is only possible for a permanent table. For all other valid Intermediate Table Type options, the SQL does not

reflect the values set for these settings. The location of each table setting in the CREATE TABLE statement is provided below.

```
create /* Table Qualifier */ table
/*Table Descriptor*//* Table Prefix
*/ZZTIS003RB6MD000 /*Table Option*/
  (STORE_NBR NUMBER,
   CLEARANCESAL DOUBLE)
/* Table Space */
/* Create PostString */
```



For more information on the Table Qualifier, Table Descriptor, Table Prefix, and Create Post String properties, see the *MicroStrategy Supplemental Reference for System Administration*.

For platforms like Teradata and DB2 UDB 6.x and 7.x, the Primary Index or the Partition Key SQL syntax is placed between the Table Space and Create Post String VLDB setting.

Table Option

The Table Option property defines the string to be placed after the table name in the CREATE TABLE statement.

Table Space

The Table Space property contains the string appended after the CREATE TABLE Statement but before any Primary Index/Partition key definitions.



To populate dynamic information by the Analytical Engine, insert the following syntax into the string:

!d inserts the date.

!o inserts the report name.

!u inserts the user name.

VLDB properties at the database instance level

The database instance level contains numerous VLDB properties. Those that are most commonly used are listed below:

- [Maximum SQL/MDX Size, page 882](#)
- [SQL Date Format, page 883](#)



For information on other database instance-level VLDB properties, refer to the *MicroStrategy Supplemental Reference for System Administration*.

Accessing database instance VLDB properties

To set these properties, right-click a project within the database instance to be updated. Select **Project Configuration**. The Project Configuration Editor opens. Expand the **Database instances** folder. The Database instances - SQL Data warehouses pane displays. Click **VLDB Properties**. The VLDB Properties (Database Instance) dialog box opens.

The Maximum SQL/MDX Size property is contained in the Governing folder, while the SQL Date Format property is found in the Select/Insert folder.

Maximum SQL/MDX Size

The Maximum SQL Size property specifies the SQL size on a pass-by-pass basis. If the limit is exceeded, report execution is terminated.

The options are:

- 0, for no limit
- Any other number, which is the maximum SQL pass size in bytes
- Use default inherited value

SQL Date Format

The SQL Date Format property specifies the format of the date string literal in the SQL statements. This setting is used when a report contains date-related qualifications.

Enter the date format using the following characters:

- mm or mmm for month
- dd for date
- yy or yyyy for year
- separators such as / and -

For example, the default format is yyyy-mm-dd. Oracle uses dd-mmm-yy and Teradata uses yyyy/mm/dd.

LOGICAL AND MATHEMATICAL OPERATORS FOR FILTERING

Introduction

As discussed in [Types of qualifications, page 127](#), you can use logical operators, such as **AND**, **OR**, **NOT**, and so on, to add additional qualifications to filters and report limits. These logical operators set conditions for the report query to retrieve data from the data warehouse and display the data in the report.

This appendix discusses operators, specifically what logical operators are and how to use them.

What is an operator?

Operators are used to manipulate individual data items and data sets. These data items are called operands or arguments. Operators are represented by special characters or by keywords. For example, multiplication is represented by an asterisk (*) and division is represented by a slash (/). Filtering conditions are expressions built from attribute forms, metrics,

constants, expressions, and operators. For example, consider the following filtering definition:

```
Store_ID = 1
```

The definition above contains an attribute (Store), an attribute form (Store_ID), a comparison operator (=), and a numeric constant (1).

The following types of operators are used when specifying filtering conditions:

- logical
- comparison
- rank and percent
- pattern

Logical operators

Logical operators allow the application of certain conditions to two sets of filter expressions simultaneously. There are three basic logical operators:

- Union behaves as the inclusive term **OR** does in grammar. The union of two sets yields a TRUE value any time that either or both of the sets of filtering criteria are met.
- Intersection behaves as the term **AND** does in grammar. The intersection of two sets yields a TRUE value only when both sets of filtering criteria are met.
- Exclusion behaves as the term **AND NOT** does in grammar. When two sets of filtering criteria are linked in this manner, their combination yields a TRUE value only when the first set is met, and the other set is not satisfied.

The following tables show the combinations possible with each logical operator, and the value that each combination yields, using the following filtering criteria as an example:

```
A =(customers located in the) Northeast region
B =(customers that purchased) blankets
```

Logical union filter: A OR B

Possible filter combinations resulting from the union of attributes A and B (customers that either are located in the Northeast region **OR** have purchased blankets) are as follows.

	A	B	Result Displayed
1	TRUE	TRUE	customers located in the Northeast region OR customers that purchased blankets
2	FALSE	TRUE	customers that purchased blankets (but are not located in the Northeast region)
3	TRUE	FALSE	customers located in the Northeast region (but have not purchased blankets)
4	FALSE	FALSE	no display (customers that are neither located in the Northeast region nor purchased blankets)

Because a union of two sets yields a valid result if data corresponding to either set is found, this filter causes a display as shown in rows 1, 2, and 3.

Logical intersection filter: A AND B

Possible filter combinations resulting from the intersection of attributes A and B (customers that are located in the Northeast region **AND** have purchased blankets) are as follows.

	A	B	Result Displayed
1	TRUE	TRUE	Customers that are located in the Northeast region AND have purchased blankets
2	FALSE	TRUE	No display (customers that purchased blankets but are not located in the Northeast region)
3	TRUE	FALSE	No display (customers that are located in the Northeast region but have not purchased blankets)
4	FALSE	FALSE	No display (customers that neither are located in the Northeast region nor have purchased blankets)

Because an intersection of two sets yields a valid result only if data corresponding to **both** sets is found, this filter causes display as shown in row 1, and in no other combination.

Logical exclusion filter: A AND NOT B

Possible filter combinations resulting from the **not (and not) exclusion** of an attribute (for example, B) (customers that are located in the Northeast region AND have not purchased blankets) are as follows.

	A	NOT B	Result Displayed
1	TRUE	TRUE	Customers who are located in the Northeast region AND have not purchased blankets
2	FALSE	TRUE	No display (customers who have not purchased blankets AND are not located in the Northeast region)
3	TRUE	FALSE	No display (customers that are located in the Northeast region AND have purchased blankets)
4	FALSE	FALSE	No display (customers not located in the Northeast region who have purchased blankets)

The behavior of exclusive * **not (and not)** statements is the same as that of intersections—the combination yields a valid result only when data corresponding to the “included” set is found and data corresponding to the “excluded” set is not. This filter causes display as shown in row 1 and in no other combination.

Logical exclusion filter: A OR NOT B

Possible filter combinations resulting from the **+ not (or not) exclusion** of an attribute (for example, B) (customers that are located in the Northeast region OR have not purchased blankets) are as follows.

	A	B	Result Displayed
1	TRUE	TRUE	Customers who are located in the Northeast region OR have not purchased blankets
2	FALSE	TRUE	Customers who have not purchased blankets OR are not located in the Northeast region
3	TRUE	FALSE	Customers that are located in the Northeast region OR have purchased blankets
4	FALSE	FALSE	No display (customers not located in the Northeast region who have purchased blankets)

The behavior of exclusive **or not** statements is the same as that of unions—the combination yields a valid result when either data corresponding to the “included” set is found **or** data corresponding to the “excluded” set is not. This filter causes display as shown in rows 1, 2, and 3.

Comparison operators

Comparison operators compare values. The values can be numbers, text strings, or expressions. The comparison operators are as follows:

- **Between:** Identifies values in a range that includes both a lower and an upper limit. For example, “between 10 and 20” returns all values that are greater than or equal to 10 and less than or equal to 20.
- **Different from:** Identifies values that are other than the specific value indicated. For example, “different from 10” returns all values that are not 10.
- **Exactly:** Identifies an specific value. For example, “exactly 1” returns all items with a value of 1.
- **Greater than:** Identifies values that are greater than an indicated lower limit. For example, “greater than 10” returns values that are greater than 10.
- **Greater than or equal to:** Identifies values that are greater than or equal to the limit indicated. For example, “greater than or equal to 10” returns all values that are 10 or greater.
- **Less than:** Identifies values that are less than an indicated upper limit. For example, “less than 10” returns values that are less than 10.
- **Less than or equal to:** Identifies values that are less than or equal to the limit indicated. For example, “less than or equal to 10” returns all values that are 10 or less.
- **Not between:** Identifies values that are outside a specified range. For example, “not between 10 and 20” returns all values that are less than 10 and greater than 20.
- **Is null:** Identifies values that are null.
- **Is Not null:** Identifies values that are not null.



When you use these operators on a description for a date, the results can appear to be incorrect. For example, assume the date description is formatted as Jan 2003. Create a filter on the description attribute

using the Between operator to return the months between January 2003 and June 2003. The results are Jan 2003, Jul 2003, and Jul 2003, not the first six months of the year as expected. This occurs because the description attribute is formatted as text, not as numbers or dates, and therefore the SQL sorts the data alphabetically. To obtain results of January 2003, February 2003, March 2003, April 2003, May 2003, and June 2003, filter on the ID rather than the description or use the in list operator.

Rank and percent operators

Rank and percent operators are only applicable to metrics. The following operators are visible when you qualify on the Rank or Percent function:

- **Between:** Identifies values in a range that has both a lower and an upper limit. For example, “between 10 and 20” returns all values that are greater than or equal to 10 and less than or equal to 20.
- **Bottom:** Identifies the lowest set of values in a range. For example, “bottom 40” returns the 40 lowest values within a given range selected.



In MicroStrategy Web, this is called Lowest.

- **Different from:** Identifies values that are other than the specific value indicated. For example, “different from 10” returns all values that are not 10.
- **Exactly:** Identifies a specific value. For example, “exactly 1” returns all items with a value of 1.
- **Exclude top:** Is used in rank and percent calculations, discards a range of top values from a given set. For example, “exclude top 10” returns all values in the set but the top 10 percent.
- **Exclude bottom:** Is used in rank and percent calculations, discards a range of bottom values from a given set. For example, “exclude bottom 10” returns all values in the set but the bottom 10 percent.
- **Is null:** Identifies values that are null
- **Is not null:** Identifies values that are not null
- **Not between:** identifies values that are outside a specified range. For example, “not between 10 and 20” returns all values that are less than 10 and greater than 20.

- **Top:** Indicates the topmost value range in a given set. For example, “top 40” returns the 40 highest values in a set.



In Web, this is called Highest.

Pattern operators

Pattern operators allow text strings to be compared. Pattern operators are case-sensitive. The following pattern operators are available in MicroStrategy:

- **Begins with:** Returns a result set that starts with a specified value. For example, “begins with J” returns all strings beginning with J, including January, June, and July.
- **Ends with:** Returns a result set that ends with a specified value. For example, “end with r” returns all strings ending with r, including September, October, and all the other months meeting the criterion.
- **Contains:** Returns a result set that contains a specified value. For example, “contains ua” returns all strings that contain ua, such as January and February.
- **Does not begin with:** Returns a result set that does not start with a specified value. For example, “does not begin with J” returns only those values that do not begin with J, such as May, February, October, and so on.
- **Does not end with:** Returns a result set that does not end with a specified value. For example, “does not end with r” returns only those strings that do not end with r, such as March, April, and all the other months meeting the criterion.
- **Does not contain:** Returns a result set that does not contain a specified value. For example, “does not contain ua” returns only those values that do not contain ua, such as March, May, and so on.

FORMATTING DEFAULT VALUES

Introduction

An autostyle is a collection of all the formatting layers, allowing you to format different grid units on different report sections. Not every grid unit must be configured to create an autostyle, so any grid unit can have formatting options set to default. Recall that formats are applied in a particular order, as described in [Order of layers, page 322](#). When the lowest layer is set to default, the options are supplied by the `guiprop.pds` file. This file is saved in the `\Common Files\MicroStrategy` directory.

This appendix provides the default values for all the formatting options. Each tab of the Format dialog box is listed below, with the default values for each option on that tab.

Default values

Number

The default values for the Number tab are:

- Category—general
- Decimal places—zero
- Use thousand separator—yes
- Negative numbers—no special consideration, meaning that neither a red font nor parentheses are used
- Currency symbol—value determined by locale settings
- Currency position—value determined by locale settings
- Format—the number's data type determines how the value is formatted; for example, a date is formatted differently than a time

Alignment

For report rows, the default values for the alignment settings are:

- Horizontal alignment—left
- Vertical alignment—center
- Wrap text—yes

For report columns, the default values for the alignment settings are:

- Horizontal alignment—center
- Vertical alignment—bottom
- Wrap text—yes

Font

The default font values are:

- Name—value determined by localization settings
- Script—western
- Bold—no
- Italic—no
- Size—10
- Strikeout—no
- Underline—no
- Color—black

Border

The values for borders are:

- Top border—yes
- Bottom border—no
- Left border—yes
- Right border—no
- Top border color—black
- Bottom border color—black
- Left border color—black
- Right border color—black

Patterns

The pattern defaults are:

- Fill color—white

- Pattern color—blue
- Pattern style—blank

Banding

While this is not a tab on the Format dialog box, banding is enabled by default for the following autostyles:

- Accounting
- Finance
- Colorful
- Greybands

Banding is accessed by selecting Options from the Grid menu.

The merge header cells option, on the Grid menu, is set to false on the autostyles listed above. This option allows element names to be repeated for a unit displayed on a row of a report. For example, Country and Region are displayed on the row axis of a report. If merge header cells is set to true, the report displays as:

Country	Region
USA	Northwest
	Northeast
	South
Germany	Saxony
	Bavaria

If merge header cells is set to false, then the report looks like the following:

Country	Region
USA	Northwest
USA	Northeast
USA	South

Country	Region
Germany	Saxony
Germany	Bavaria

MICROSTRATEGY DEVELOPER COMMANDS

Introduction

MicroStrategy Developer commands are a collection of methods that MicroStrategy Developer supports. These commands provide functionalities such as executing a report, loading an editor, and so on. With these commands you can create your own project homepage and customize it according to your user community's needs.



Even though MicroStrategy Developer commands can be invoked from the command line, this appendix focuses on the MicroStrategy Developer Homepage usage and describes the commands from an HTML perspective.

This appendix covers the following topics on how MicroStrategy Developer commands can be used in MicroStrategy products:

- *[Basics, page 900](#)*
- *[Enabling the MicroStrategy Developer homepage, page 900](#)*
- *[Viewing the MicroStrategy Developer commands, page 902](#)*
- *[Commands, page 902](#)*

Basics

MicroStrategy Developer commands are created using HTML anchor elements. Anchor elements typically contain a reference to a uniform resource locator (URL). When an anchor is clicked, it performs an operation depending on the URL scheme.

For example, a file transfer protocol (ftp) anchor starts a file transfer operation; other schemes are http, gopher, and so on. To differentiate a MicroStrategy Developer command from other anchors, MicroStrategy Developer commands use a unique syntax. An anchor with this special syntax is referred to as a Developer anchor. The following is the HTML syntax of a Developer anchor:

```
<a href="dss://Command Parameters"></a>
```

Enabling the MicroStrategy Developer homepage

The MicroStrategy Developer homepage is the first window displayed when you log in to a MicroStrategy project in the MicroStrategy Developer environment. It serves as the primary point of access to the editors, dialogs, and wizards that enable the use of MicroStrategy Developer interface functionality.

When you select a project in MicroStrategy Developer, you see the HTML homepage. This homepage can display links to the reports, folders, documents, description of the project, and so on. To see an HTML page as shown in step 6 of the following procedure, log in to MicroStrategy Tutorial as **User**.

The MicroStrategy Developer homepage is displayed when you select the project within a project source. The objects within the project are always displayed as folders.

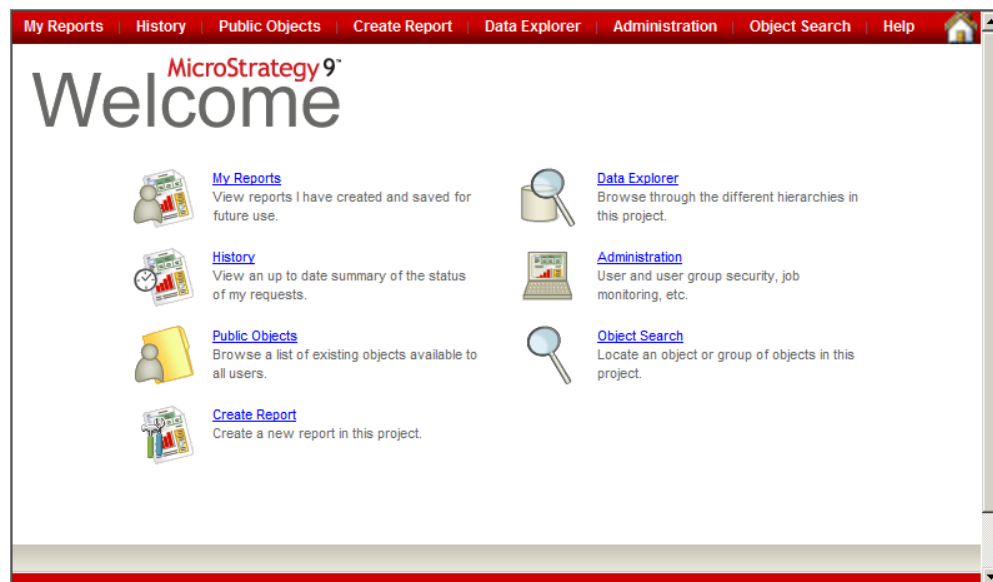
You can choose to display the project homepage or the Folder List. The procedure below describes how to enable and select a project homepage.

To enable and select a project homepage

- 1 Log in to the MicroStrategy project source containing the project to enable a project homepage for.
- 2 From the **Tools** menu, select **My Preferences**. The My Preferences dialog box opens.
- 3 Select the **Enable project home page functionality** check box.
- 4 Browse for and locate the HTML file for the project homepage in the **HTML file path** box or use the default homepage location displayed in this box.

 If you have created your own customized HTML page to use in the project, locate the file using **Browse** button. Otherwise, the project uses the default homepage designed by MicroStrategy.

- 5 Click **OK**.
- 6 Select the project within the project source. A project homepage is displayed. For example, in the MicroStrategy Tutorial you will see the HTML homepage which looks like the following:



If you cannot see the HTML project homepage even after you have enabled the homepage option from the My Preferences dialog box, this is because the project homepage option is not enabled in the Developer Preferences dialog box. To do this, complete the following procedure.

To enable a project homepage in Developer preferences

- 7 From the **Tools** menu, select **Preferences**. The Developer Preferences dialog box opens.
- 8 In the dialog box, select the **Enable project home page functionality** option.
- 9 Click **OK**.

To work with the homepage functionality, the **Enable project home page functionality** option should always be enabled in both My Preferences and Developer Preferences dialog boxes.

Viewing the MicroStrategy Developer commands

In the homepage, MicroStrategy Developer commands exist embedded in the HTML document. When you view the source code of the HTML page, you can view the MicroStrategy Developer commands.

To see where in the HTML the MicroStrategy Developer commands are embedded, right-click the HTML page and from the shortcut menu, select **View Source**. The HTML code is displayed in your default text editor. Search the code for anchor elements that have syntax that matches the following statement:

```
<A hRef="dss://Command Parameters"></A>
```

Many MicroStrategy Developer commands use a unique object ID to refer to objects in a project. To view an object's ID, in MicroStrategy Developer right-click the object and select **Properties**. The ID is displayed in the dialog box.

Commands

MicroStrategy Developer supports the following commands:

- [ChangeView, page 903](#): updates the MicroStrategy Developer view style
- [Editor, page 904](#): loads a MicroStrategy Developer editor

- [Execute, page 905](#): executes a report or document definition
- [ExecuteDocument, page 906](#): executes a document definition
- [ExecuteReport, page 906](#): executes a report definition
- [Open, page 907](#): opens a connection to a project source or a session to a project
- [Reset, page 908](#): closes a connection to a project source or a session to a project
- [Shortcut, page 909](#): finds and selects a node in the folder list pane of the object browser

The description of each of these commands is in the following sections.

ChangeView

The ChangeView command shows or hides the shortcut and folder list panes in the object browser.

```
dss://ChangeView View1\View2\...\Viewn
```

Parameter	Description
View	The view parameter indicates what operation should be executed. The following operations are supported: • ShowShortcutView • ShowFolderView • HideShortcutView • HideFolderView

Remarks

The ChangeView command can take a list of operations in a single command. For example, to show shortcuts and hide the folder list use the following command:

```
ChangeView ShowShortcutView\HideFolderView
```

Examples

```
<A hRef="dss://changeview showshortcutview">  
Show Shortcuts</A>
```

```
<A href="dss://changeview showfolderview">  
Show Folders</A>
```

```
<A href="dss://changeview hideshowshortcutview">  
Hide Shortcuts</A>
```

```
<A href="dss://changeview hidefolderview">  
Hide Folders</A>
```

```
<A href="dss://changeview  
hidefolderview\showshortcutview"> Hide Folders  
and Show Shortcuts</A>
```

```
<A href="dss://changeview  
showfolderview\hideshowshortcutview"> Show Folders  
and Hide Shortcuts</A>
```

Editor

The Editor command loads a new instance of a specified MicroStrategy Developer editor in the currently selected project.

```
dss://editor EditorName
```

Parameter	Description
EditorName	Indicates the name of the editor to load. The command supports the following editors: • Search • ReportDefinition • DocumentDefinition • Prompt • Filter • Template • Metric • CustomGroup • Consolidation • Attribute • Fact • Hierarchy • Transformation • Partition

Example

```
<A href="dss://editor search">Open Search
Editor</A>
```

Execute

The Execute command executes one or more reports or documents. The reports or documents are identified by their object type and object ID.

```
dss://Execute ObjID1.ObjType1\ObjID2.ObjType2
\...\ObjIDn.ObjTypen
```

Parameter	Description
ObjID	The ID of the object to be executed. To view an object's ID, in MicroStrategy Developer right-click the object and select Properties . The ID is displayed in the dialog box.
ObjType	The object type of the object to be executed. The type is used from the EnumDSSObjectType enumeration in the MicroStrategy Objects type library. The command supports the following two types: • 3—DSSTypeReportDefinition • 55—DSSTypeDocumentDefinition

Example

```
<A href="dss://execute B5C67DFC11D60B5610008C
B3D1CEE6A4.3\48CAD4644AB189F763E0EAA22BC0E6D
C.55">Execute: {Profit Forecast 2003, Document
(Customer Hierarchy)}</A>
```

ExecuteDocument

The ExecuteDocument command loads the Document Editor. The command can execute one or more documents.



The ExecuteDocument command executes the document only if the current project source uses a server connection (three-tier).

```
dss://ExecuteDocument DocumentID1\DocumentID2\...\DocumentIDn
```

Parameter	Description
DocumentID	The ID of the document definition object. The command takes any number of documents to execute. To view an object's ID, in MicroStrategy Developer right-click the object and select Properties . The ID is displayed in the dialog box.

Examples

```
<A href="dss://executedocument 3D4DA91C4D20DA
7532D4AB848C428031">Execute Document:
{Document (My Electronics Dashboard)}</A>
```

```
<A href="dss://executedocument 3D4DA91C4D20D
A7532D4AB848C428031\0BD252404BB97A2167B08584
8A40A60B">Execute Document:
{Document (My Electronics Dashboard),
Document (Product Hierarchy)}</A>
```

ExecuteReport

The ExecuteReport command runs a report and displays it in the view (Grid, Graph, or Grid Graph) that it was saved in. The command can execute one or more reports.


```
dss://ExecuteReport ReportID1\ReportID2\...\ReportIDn
```

Parameter	Description
ReportID	The ID of the report. The command takes any number of reports to execute. To view an object's ID, in MicroStrategy Developer right-click the object and select Properties . The ID is displayed in the dialog box.

Examples

```
<A href="dss://executereport BF294AA247895DD
9354CA9B296D91D33">Execute Report: {Electronics
Revenue vs. Forecast 2003}</A>
```

```
<A href="dss://executereport BF294AA247895DD
9354CA9B296D91D33\2C3DFFB411D6044FC0008C916B
98494F">Execute Report: {Electronics Revenue vs.
Forecast 2003, Electronics Revenue By Region}</A>
```

Open

The Open command connects to a project source in the object browser. The Open command searches for the project source by name. The search is case sensitive. After the project source is found, it is expanded. If the user is currently not connected to the project source the command opens the login window.

You can also include a project ID to open a project.

```
dss://Open ProjectSourceName\ProjectID\UserLogin\UserPassword
```

Parameter	Description
ProjectSourceName	The name of the project source node in the object browser control.
ProjectID	Optional. If the ProjectID parameter is included, the command finds and connects to the project. To obtain a project ID, right-click the project and select Project Configuration . The ID is displayed in the dialog box.

Parameter	Description
UserLogin	Optional. The UserLogin parameter defines the default value of the user login when the project source is expanded. The login window is only displayed if the user is not currently connected to the project source.
UserPassword	Optional. The UserPassword parameter defines the default value of the user password when the project source is expanded. The login window is only displayed if the user is not currently connected to the project source.

Examples

```
<A href="dss://open MicroStrategy Tutorial">
Open the MicroStrategy Tutorial project source
</A>
```

```
<A href="dss://open MicroStrategy Tutorial
\B19DEDCC11D4E0EFC000EB9495D0F44F">Open Tutorial
Project</A>
```

```
<A href="dss://open MicroStrategy Tutorial
\B19DEDCC11D4E0EFC000EB9495D0F44F\
Administrator">Open Tutorial Project using
Administrator login </A>
```

Reset

The Reset command closes a session to a project or a connection to a project source.

```
dss://Reset ProjectSourceName\ProjectID
```

Parameter	Description
ProjectSourceName	The name of the project source in the folder list.
ProjectID	Optional. If the ProjectID parameter is sent, the command closes the session to the project. Otherwise, the command closes the connection to the project source. To obtain a project ID, right-click the project and select Project Configuration . The ID is displayed in the dialog box.

Examples

```
<A hRef="dss://reset MicroStrategy Analytics Modules"> Close
connection to the MicroStrategy
Analytics Modules project source</A>
```

```
<A hRef="dss://reset MicroStrategy Tutorial
\B19DEDCC11D4E0EFC000EB9495D0F44F"> Close session
to Tutorial Project</A>
```

Shortcut

The Shortcut command opens a folder in the object browser and displays the contents of the folder. This command searches for the folder ID in the project that the user is currently browsing. To select a project in the object browser use the Open command (see [Open, page 907](#)).

```
dss://Shortcut FolderID
```

Parameter	Description
FolderID	The ID of the target folder. To obtain a folder ID, right-click the folder and select Properties . The Properties dialog box displays the ID of the folder.

The folder ID parameter can specify one of the following special folder names instead of a folder ID.

Profile_MyAnswers	Public_Reports	Public_Templates
Profile_MyFavorites	Public_Prompts	Schema_Partition_Mappings
Profile_MyObjects	Public_Searches	Schema_Tables
Profile_MyReports	Public_Metrics	Schema_Hierarchies
Public_Autostyles	Public_Filters	Schema_Functions
Public_Consolidations	Schema_Objects	Data_Explorer
Public_Custom_Groups	Schema_Attributes	Server_Admin
Public_Objects	Schema_Facts	Schema_Transformations

Examples

```
<A hRef="dss://shortcut A20C898211D60AE31000
```

```
8BB3D1CEE6A4">Financial Reports</A>
```

```
<A hRef="dss://shortcut profile_myreports">  
My Reports</A>
```

SORTING AND BROWSING STANDARDS

Introduction

MicroStrategy employs its own sorting and browsing standards for characters and MicroStrategy projects.

The sorting is also dependent on your machine's locale. International Components for Unicode (ICU) libraries are used to compare strings. ICU uses the Unicode Collation Algorithm, which sorts on multiple levels, as described below:

- 1 Letters are sorted first.
- 2 Differences in accents resolve any ties resulting from the first sort level.
- 3 Differences in case (lower case vs. upper case) resolve any ties.
- 4 Differences in punctuation resolve any ties.

Sort order of characters in MicroStrategy

The sort order of characters in MicroStrategy is different from ASCII standards. The MicroStrategy sort order of characters affects:

- The order of files and folders in MicroStrategy Developer and MicroStrategy Web
- The order of ascending and descending sorts of alphanumeric values in MicroStrategy reports

The sort order described below is for English locales only. The sort order may be different in different locales.

The table below lists the non-printable characters which are first in the sort order. All these characters are equal to each other, so the final sorting order is determined by which character is retrieved from the data warehouse first.



ASCII characters that are non-printable characters are listed in the tables below with their character representation. For example, a backspace is represented as BS.

NULL	SOH	STX	ETX	EOT	ENQ	ACK
BEL	BS	SO	SI	DLE	DCL	DC2
DC3	DC4	NAK	SYN	ETB	CANC	EM
SUB	ESC	FS	GS	RS	US	DEL

The table below lists the sort order of characters in MicroStrategy. The order goes from left to right and top to bottom. For example, the top left table cell (TAB) is the beginning of the sort order, the cell to the right of that (LF) is next in order, and the bottom right cell (Z) is the end of the sort order.

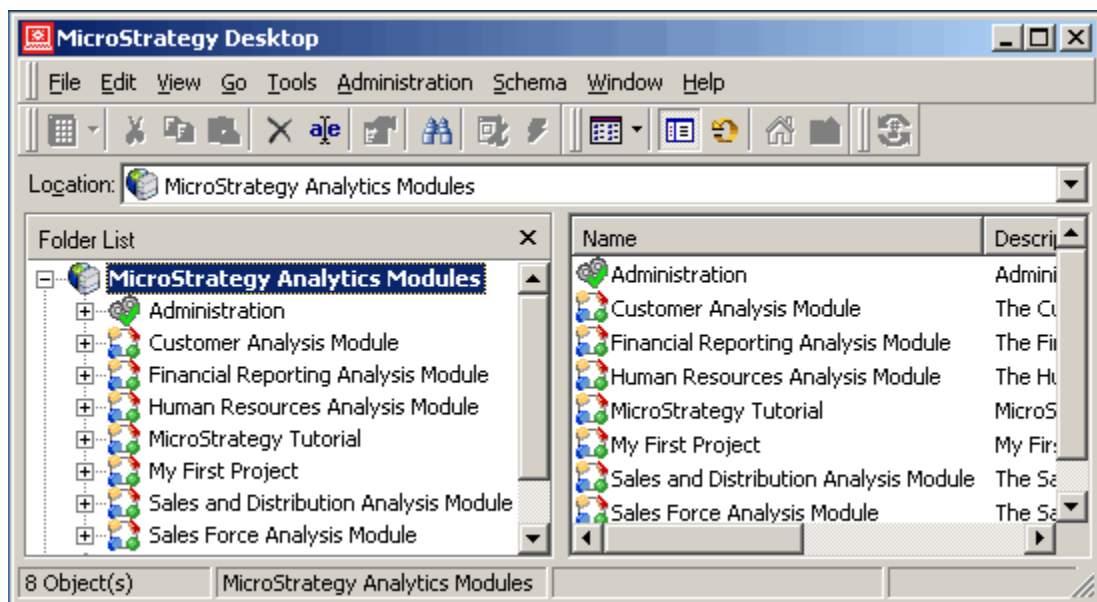
TAB	LF	VT	FF	CR		'	^	_	-
,	;	:	!	?	.	'	"	(	)
[	]	{	}	@	*	/	\	&	#
%	+	<	=	>		~	\$	0	1
2	3	4	5	6	7	8	9	a	A
b	B	c	C	d	D	e	E	f	F
g	G	h	H	i	I	j	J	k	K
l	L	m	M	n	N	o	O	p	P

q	Q	r	R	s	S	t	T	u	U
v	V	w	W	x	x	y	Y	z	Z

Project order in MicroStrategy Developer and Web

Projects in MicroStrategy can only be created in MicroStrategy Developer, but they can be accessed in both MicroStrategy Developer and Web. After a project is created and its associated project source is refreshed, the projects can be sorted and browsed in the following ways:

- **MicroStrategy Developer:** The projects are sorted in alphabetical order within the folder list, as shown in the image below.



- **Web:** By default, projects are sorted by project name, without regard to which Intelligence Server that they are on, but you can use MicroStrategy Web to change how the projects are sorted.
 - The **Server Sorting** setting indicates whether the projects are grouped together by Intelligence Server before sorting. The options are:
 - **None:** (Default) Projects are sorted independently of which Intelligence Server they are on.

- **By Name:** Intelligence Servers are sorted alphabetically by name, and then projects are sorted within each Intelligence Server.
- **Default:** Intelligence Servers are sorted in the order in which MicroStrategy Web connects to them, and then projects are sorted within each Intelligence Server.
- The **Project Sorting** setting indicates how the projects are sorted. The options are:
 - **Default:** Projects are sorted in the order in which the projects are returned from Intelligence Server.
 - **By Name:** (Default) The projects are sorted by project name, in ascending alphabetical order.
 - **By Description:** The projects are sorted by project description, in ascending alphabetical order.

Both the **Server Sorting** and **Project Sorting** settings are found in the Default Server Properties area on the Web Administrator page.

- You can also define a custom sort order for displaying projects, by typing a number in the **Project Sort Index** field in the Project Display page in the Project Defaults level of the Preferences. Any projects with a custom sort index are sorted before all other projects, regardless of how the **Server Sorting** and **Project Sorting** settings are defined. If two projects have the same sort order index, then they are listed in the order specified by the **Project Sorting** setting described above.

GLOSSARY

aggregate function A numeric function that acts on a column of data and produces a single result. Examples include SUM, COUNT, MAX, MIN, and AVG.

aggregate table A fact table that stores data that has been aggregated along one or more dimensions.

See **pre-aggregation**.

aggregation The combining of numeric data at a specific attribute level. The most common function is sum, which creates an additive total.

See also **pre-aggregation**.

allocation An optional aspect of a fact extension that allows distribution of values according to a user-defined calculation expression.

Compare **degradation**.

Analytical Engine A component of MicroStrategy Intelligence Server responsible for performing complex mathematical calculations, consolidations, and advanced analytical functions needed to create a report. The Analytical Engine enables you to manipulate report data by sorting, pivoting, page-by, calculating thresholds, calculating subtotals and metrics. The Analytical Engine evaluates a filter without

re-executing the report and also facilitates hierarchical display of custom groups.

application-level partition In application-level partitioning, the application rather than the database server manages the partition tables. MicroStrategy supports two methods of application-level partitioning: metadata partition mapping and warehouse partition mapping.

Compare **database-level partition**.

application object MicroStrategy object used to provide analysis of and insight into relevant data. Application objects are developed in MicroStrategy Developer and they are the building blocks for reports and documents. Application objects include these object types: report, document, template, filter, metric, custom group, consolidation, prompt.

Apply function A function that allows you to insert custom SQL into an expression.

See also **pass-through expression**.

atomic The lowest level of granularity. Cannot be decomposed into smaller parts.

attribute A data level defined by the system architect and associated with one or more columns in a data warehouse lookup table. Attributes include data classifications like Region, Order, Customer, Age, Item, City, and Year. They provide a means for aggregating and filtering at a given level.

See also:

- **attribute element**
- **attribute form**
- **child attribute**
- **constant attribute**
- **derived attribute**
- **parent attribute**

attribute element A value of any of the attribute forms of an attribute. For example, New York and Dallas are elements of the attribute City; January, February, and March are elements of the attribute Month.

attribute form One of several columns associated with an attribute that are different aspects of the same thing. ID, Name, Last Name, Long Description, and Abbreviation could be forms of the attribute Customer. Every attribute supports its own collection of forms.

attribute relationship See **relationship**.

attribute role A database column that is used to define more than one attribute. For example, Billing City and Shipping City are two attributes that have the same table and columns defined as a lookup table.

banding A method of organizing values according to a set of descriptive or meaningful data ranges called buckets. For example, customers in the age ranges of 10–20, 21–30, and 31–40, where each set of ages is a band. Banding is also used for display purposes, where every other row is a different color and the two colors alternate.

Compare **consolidation**.

base table A fact table that stores data at the lowest level of dimensionality.

cache A special data store holding recently accessed information for quick future access. This is normally done for frequently requested reports, whose execution is faster because they need not run against the database. Results from the data warehouse are stored separately and can be used by new job requests that require the same data. In the MicroStrategy environment, when a user runs a report for the first time, the job is submitted to the database for processing. However, if the results of that report are cached, the results can be returned immediately without having to wait for the database to process the job the next time the report is run.

category In a graph, the set of data along the X-axis. Categories generally correspond to the rows of a grid report. An example of a category is a bar in a bar graph.

child attribute The lower-level attribute in an attribute relationship.

See also:

- **parent attribute**
- **relationship**

compound attribute An attribute that has more than one key (ID) form.

compound key In a relational database, a primary key consisting of more than one database column.

compound metric A type of metric defined by a formula based on arithmetic operators and non-group functions. Arithmetic operators are +, -, *, and /; non-group functions are OLAP and scalar functions such as running sum or rank. The operators and functions can be applied to facts, attributes, or metrics.

Examples are `RunningAvg(Cost_Metric)` and `Sum(Cost_Metric) + Sum(Profit_Metric)`.

- compression ratio** The average number of child records combined to calculate one parent record. For example, the compression of ratio between monthly data and yearly data is 12:1. This is used to determine where aggregate tables would have the greatest impact. The larger the compression ratio between two attributes, the more you stand to gain by creating an aggregate table that pre-calculates the higher-level data.
- configuration object** A MicroStrategy object appearing in the system layer and usable across multiple projects. Configuration objects include these object types: users, database instances, database login IDs, schedules.
- consolidation** An object that can be placed on a template and is made up of an ordered collection of elements called consolidation elements. Each element is a grouping of attribute elements that accommodates inter-row arithmetic operations.
- Compare **custom group**.
- consolidation element** A line item in a consolidation based on attribute elements. For example, Year=2002 / Year=2003.
- constant attribute** See **implicit attribute**.
- custom group** An object that can be placed on a template and is made up of an ordered collection of elements called custom group elements. Each element contains its own set of filtering qualifications.
- dashboard** A popular means of displaying and distributing data from business intelligence projects. Dashboards provide key metrics as well as summary information.
- data definition** Report execution steps that establish how the data is accessed and manipulated in the data warehouse.

data mining A technique that is generally used to find hidden predictive information from a large amount of data. This process involves using existing information to gain new insights into business activities by applying predictive models, using analysis techniques such as regression, classification, clustering, and association.

Data Explorer A portion of the interface used to browse through data contained in the warehouse. Users can navigate through hierarchies of attributes that are defined by the administrator to find the data they need.

database-level partition In database-level partitioning (sometimes called server-level partitioning), the database server rather than MicroStrategy manages the partitioned tables. The original table is not physically broken into smaller tables. Instead, the database server logically partitions the table according to parameters specified by the database administrator. You do not need to take any action in MicroStrategy to support the partitioning. Since only the logical table is displayed to the end user, the partitioning is transparent to MicroStrategy.

Compare **application-level partitioning**.

data mart 1) A database, usually smaller than a data warehouse, designed to help managers make strategic decisions about their business by focusing on a specific subject or department.

2) A database instance used to store result sets saved to data mart tables.

data mart report A special kind of report that saves its report data in a database rather than returning those results to the user. Data mart reports either create a new table in the database to store the report data or append the report data into an existing table.

data mart table A relational table that is used to store the report data from a data mart report.

data warehouse 1) A database, typically very large, containing the historical data of an enterprise. Used for decision support or business intelligence, it organizes data and allows coordinated updates and loads.

2) A copy of transaction data specifically structured for query, reporting, and analysis.

degradation A type of fact extension in which values at one level of aggregation are reported at a second, lower attribute level.

Compare **allocation**.

derived attribute An attribute calculated from a mathematical operation on columns in a warehouse table. For example, Age can be calculated from the expression [Current Date–Birth Date].

See also:

- **attribute**
- **implicit attribute**

derived metric A metric based on data already available on the report. It is calculated on the Intelligence Server, not in the database. Use a derived metric to perform column math, which is calculations on other metrics, on report data after it has been returned from the database.

dimensionality See **level**.

drill A method of obtaining supplementary information after a report has been executed. The new data is retrieved by requering the Intelligent Cube or database at a different attribute or fact level.

See also:

- **page-by**
- **pivot**
- **sort**
- **subtotal**

dynamic aggregation Rollup of metric values that occurs when an attribute is moved from the report grid to the Report Objects. Whenever the attributes in the Report Objects are not the same as the attributes on the grid, dynamic aggregation has occurred. Dynamic aggregation happens on-the-fly, in memory.

dynamic relationship When the relationship between elements of parent and child attributes changes. These changes often occur because of organizational restructuring; geographical realignment; or the addition, reclassification, or discontinuation of items or services. For example, a store may decide to reclassify the department to which items belong.

entry level The lowest level set of attributes at which a fact is available for analysis.

extraction, transformation, and loading (ETL) 1) The process used to populate a data warehouse from disparate existing database systems.
2) Third-party software used to facilitate such a process.

fact 1) A measurement value, often numeric and typically aggregatable, stored in a data warehouse.

2) A schema object representing a column in a data warehouse table and containing basic or aggregated numbers—usually prices, or sales in dollars, or inventory quantities in counts.

See also **metric**.

fact table A database table containing numeric data that can be aggregated along one or more dimensions. Fact tables can contain atomic or summarized data.

Compare:

- **aggregate table**
- **base table**

filter A MicroStrategy object that specifies the conditions that the data must meet to be included in the report results. Using a filter on a report narrows the data to consider only the information that is relevant to answer your business question, since a report queries the database against all the data stored in the data warehouse.

A filter is composed of at least one qualification, which is the actual condition that must be met for the data to be included on a report. Multiple qualifications in a single filter are combined using logical operators. Examples include “Region = Northeast” or “Revenue > \$1 million”.

A filter is normally implemented in the SQL WHERE clause.

formatting layer The part of a report that allows you to control how a report looks. The basic formatting layers are zones, which are the rows and headers of a report, and grid units, which are the attribute values. Other formatting layers, such as thresholds and subtotals, can be thought of as extensions of these two basic types.

formatting zone Determines what formatting is applied to any data or object located in the zone. When an object on a report is moved from one formatting zone to another (as a result of pivoting, for example), the formatting of the object changes based on the new zone.

Freeform SQL A MicroStrategy reporting feature that allows you to use your own customized SQL statements to retrieve data from any relational databases that are included in a MicroStrategy project.

grid unit The individual attributes, metrics, consolidations, and custom groups that can be placed on a report grid.

hierarchy A set of attributes defining a meaningful path for element browsing or drilling. The order of the attributes is typically—though not always—defined such that a higher attribute has a one-to-many relationship with its child attributes.

implicit attribute An attribute that does not physically exist in the database because it is created at the application level. Such an attribute has its expression defined as a constant value, though nothing is saved in a column. For example, you may wish to create columns in the database with a value of 1 for every row to get around COUNT limitations. You do not have to actually create the column, though, because in the Attribute Editor, you can just enter a “1” in the expression to create a count. Implicit attributes are useful in analyzing and retrieving information. When analyzing data, you can use constant attributes to create a COUNT to keep track of the number of rows returned. You can use constant attributes when building metrics, where you can sum the column holding the constant to create a COUNT. Any constant is acceptable.

Compare **derived attribute**.

Intelligent Cube A copy of the report data saved in memory and used for manipulation of the view definition. This division allows multiple reports with different views to share a common data definition.

joint children Joint child relationships are another type of many-to-many relationship where one attribute has a many-to-many relationship to two otherwise unrelated attributes. These relationships can be modeled and conceptualized like traditional attributes, but like facts, they exist at the intersection of multiple attribute levels. For example, consider the relationship between three attributes: promotion, item, and quarter. In this case, promotion has a many-to-many relationship to both item and quarter. An example of a promotion might be a “Red Sale” where all red items are on sale. A business might run this promotion

around Valentine's Day (Q1) and again at Christmas time (Q4).

key form One of a set of attribute forms required for unique identification of an element in an attribute. Also called the ID or ID form.

See also **attribute form**.

level 1) In a data warehouse, facts are said to be stored at a particular level defined by the attribute IDs present in the fact table. For example, if a fact table has a Date column, an Item_ID column, and a fact column, that fact is stored at the Date/Item level.

2) With regard to metric calculation, the level is the level of calculation for the metric. For example, a metric on a report with Year and Store attributes would be calculated at the Year/Store level.

See also **level of aggregation**.

level of aggregation The point in an attribute hierarchy where aggregation is performed. For example, in the geographical State--City--Store hierarchy there are three possible levels of aggregation.

link A connection in one report or document to another report or document. A link lets an analyst execute another document or report (the target) from a document or report (the source), and to pass parameters to answer any prompts that are in the target.

logical data model A graphical representation of data that is arranged logically for the general user, as opposed to the physical data model or warehouse schema, which arranges data for efficient database use.

logical table Logical tables are MicroStrategy objects that form the foundation of a schema. Logical tables in the MicroStrategy

schema consist of attributes and facts. There are three types of logical tables:

- A logical table is created for each physical table that is imported into a project, using the Warehouse Catalog. This type of logical tables maps directly to physical tables in the data warehouse.
- A logical view does not point directly to a physical table and is defined using a SQL query against the data warehouse.
- A table alias points directly to a physical table. A table alias can have a different name from the physical table.

See also:

- **logical view**
- **table alias**

logical view One of the three types of logical tables in the MicroStrategy environment. The other two types are logical tables and table aliases. A logical view does not point directly to a physical table and is defined using a SQL query against the data warehouse. Using MicroStrategy, you can create logical views, which can be used in the same way as the logical tables. This means that you define attributes, facts, and other schema objects based on the logical views.

See also **logical table**.

locked hierarchy A hierarchy that has at least one attribute that may not be browsed by end users. Application Designers typically lock hierarchies if there are so many attribute elements that element browsing is not usable.

managed object A schema object unrelated to the project schema, which is created by the system and stored in a separate system folder. Managed objects are used to map data to attributes, metrics, hierarchies and other schema objects for Freeform SQL, Query Builder, and MDX cube reports.

many-to-many An attribute relationship in which multiple elements of a parent attribute can relate to multiple elements of a child attribute, and vice versa.

See also:

- **one-to-one**
- **one-to-many**
- **many-to-one**
- **relationship**

many-to-one An attribute relationship in which (1) multiple elements of a parent attribute relate to only one element of a child attribute, and (2) every element of the child attribute can relate to multiple elements of the parent.

See also:

- **one-to-one**
- **one-to-many**
- **many-to-many**
- **relationship**

metadata A repository whose data associates the tables and columns of a data warehouse with user-defined attributes and facts to enable the mapping of the business view, terms, and needs to the underlying database structure. Metadata can reside on the same server as the data warehouse or on a different database server. It can even be held in a different RDBMS.

metric 1) A business calculation defined by an expression built with functions, facts, attributes, or other metrics. For example:
`Sum(dollar_sales)` or `[Sales] - [Cost]`.

2) The MicroStrategy object that contains the metric definition. It represents a business measure or key performance indicator.

See also **fact**.

MOLAP Multidimensional online analytical processing.

multidimensional cache Copy of the report data saved in memory. This cache is used for manipulation of the view definition. Also called an Intelligent Cube.

multidimensional expression A query language similar to SQL. MDX is defined by Microsoft. An MDX expression returns a multidimensional result set (dataset) that consists of axis data and cell data. MDX is used to query cubes, which are used by SAP BI to store data. When accessing the data from SAP BI to generate reports, the MicroStrategy Intelligence Server generates MDX.

See also **SAP BI**.

nonaggregatable metric A metric that is not additive along all dimensions. For example, “Stock On Hand at End of Week” is not additive across time: the stock on hand at the end of the week is not the sum of the stock on hand at end of each day in the week.

one-to-many An attribute relationship in which every element of a parent attribute can relate to multiple elements of a child attribute, while every element of the child attribute relates to only one element of the parent. The one-to-many attribute relationship is the most common in data models.

See also:

- **one-to-one**
- **many-to-many**
- **many-to-one**
- **relationship**

one-to-one An attribute relationship in which every element of the parent attribute relates to exactly one element of the child attribute, and vice versa.

See also:

- **one-to-many**
- **many-to-one**
- **many-to-many**
- **relationship**

page-by Segmenting data in a grid report by placing available attributes, consolidations, and metrics on a third axis called the Page axis. Since a grid is two-dimensional, only a slice of the cube can be seen at any one time. The slice is characterized by the choice of elements on the Page axis. By varying the selection of elements, the user can page through the cube.

See also:

- **drill**
- **pivot**
- **sort**
- **subtotal**

parent attribute The higher-level attribute in an attribute relationship with one or more children.

See also:

- **child attribute**
- **relationship**

partial relationship An attribute relationship in which elements of one attribute relate to elements of a second attribute, while the opposite is not necessarily true.

See also:

- **relationship**
- **one-to-many**
- **many-to-one**
- **many-to-many**

partition A relational database table broken down into smaller component tables. This can be done at the database level or at the application level.

See also:

- **application-level partition**
- **database-level partition**

partition base table A warehouse table that contains one part of a larger set of data. Partition tables are usually divided along logical lines, such as time or geography. Also referred to as a PBT.

See also **partition mapping**.

partition mapping The division of large logical tables into smaller physical tables based on a definable data level, such as month or department. Partitions minimize the number of tables and records within a table that must be read to satisfy queries issued against the warehouse. By distributing usage across multiple tables, partitions improve the speed and efficiency of database queries.

partition mapping table A warehouse table that contains information used to identify the partitioned base tables as part of a logical whole. Also referred to as a PMT.

See also:

- **partition base table**
- **partition mapping**

pass-through expression An expression that provides access to special functions or syntactic constructs that are not standard in MicroStrategy, but are offered by various relational database management system (RDBMS) platforms.

See also **Apply function**.

physical warehouse schema A detailed graphic representation of your business data as it is stored in the data warehouse. It organizes the logical data model in a method that make sense from a database perspective.

See also **schema**.

pivot To reconfigure data on a grid report by placing report objects (attributes, metrics, consolidations) on different axes. Also, to reconfigure a grid report by interchanging row and column headers, and hence the associated data. Subset of cross-tab.

See also:

- **drill**
- **page-by**
- **sort**
- **subtotal**

PMML An XML standard used to represent data mining models that thoroughly describes how to apply a predictive model. It was developed by the Data Mining Group, DMG, an independent consortium consisting of over two dozen companies including MicroStrategy.

predictive input metric A metric used in data mining that encapsulates the definition of another attribute or metric. There are three types of predictive input metrics: attribute-based input metrics, metric-based input metrics, and conditional input metrics.

pre-aggregation Aggregation, or the calculation of numeric data at a specific attribute level, that is completed before reports are run, with the results stored in an aggregate table.

See also:

- **aggregate table**
- **aggregation**

prompt 1) MicroStrategy object in the report definition that is incomplete by design. The user is asked during the resolution phase of report execution to provide an answer that completes the information. A typical example with a filter is choosing a specific attribute on which to qualify.

2) In general, a window requesting user input, as in “type login ID and password at the prompt.”

qualification The actual condition that must be met for data to be included on a report. Examples include “Region = Northeast” or “Revenue > \$1 million”. Qualifications are used in filters and custom groups. You can create multiple qualifications for a single filter or custom group, and then set how to combine the qualifications using the logical operators AND, AND NOT, OR, and OR NOT.

quality relationship The relationship between a parent attribute and two or more “joint child” attributes. The parent attribute is referred to as a “quality” because its definition is complete only with the intersection of its joint children.

regression A data mining technique that analyzes the relationship between several predictive inputs, or independent variables, and a dependent variable that is to be predicted.

relationship An association specifying the nature of the connection between one attribute (the parent) and one or more other attributes (the children). For example, City is a child attribute of State.

See also:

- **parent attribute**
- **child attribute**
- **partial relationship**
- **quality relationship**
- **one-to-one**
- **one-to-many**
- **many-to-one**
- **many-to-many**

report The central focus of any decision support investigation, a report allows users to query for data, analyze that data, and then present it in a visually pleasing manner.

See also:

- **filter**
- **template**

report creation The process of building reports from existing, predesigned reports in MicroStrategy Developer or in MicroStrategy Web.

report design The process of building reports from basic report components using the Report Editor in MicroStrategy Developer or MicroStrategy Web.

SAP BI SAP Business Intelligence is the business intelligence tool developed by SAP. MicroStrategy's integration with SAP BI allows users to conduct reporting and analysis with the data from SAP BI. For information on SAP BI, please refer to documentation by SAP; for information on MicroStrategy's integration with SAP BI, refer to the *MDX Cube Reporting Guide*.

See also **multidimensional expression**.

- schema** 1) The set of tables in a data warehouse associated with a logical data model. The attribute and fact columns in those tables are considered part of the schema itself.
- 2) The layout or structure of a database system. In relational databases, the schema defines the tables, the fields in each table, and the relationships between fields and tables.
- schema object** MicroStrategy object created, usually by a project designer, that relates the information in the logical data model and physical warehouse schema to the MicroStrategy environment. These objects are developed in MicroStrategy Architect, which can be accessed from MicroStrategy Developer. Schema objects directly reflect the warehouse structure and include attributes, facts, functions, hierarchies, operators, partition mappings, tables, and transformations.
- scorecard** A popular means of displaying and distributing data from business intelligence projects. Scorecards typically follow a specific methodology and are focused on key metrics within a business area.
- series** In a graph, it generally corresponds to the rows of a grid report. Series are represented as legend items in a graph.
- shadow metric** A metric that represents the attribute to be included in a predictive model. It allows an attribute to be used as a predictor.
- shortcut metric** A metric based on metrics already included in a report. They provide a quick way to add new metrics to that report. Shortcut metrics belong to one of these types: percent-to-total metrics, transformation metrics, rank metrics, and running sum metrics.
- simple metric** A type of metric defined by a formula containing a mathematical expression based on at least one group function, such as sum or average, which is applied to facts, attributes, or other metrics. It can also contain non-group

functions or arithmetic operators, in addition to the required group function.

Examples are `Avg(Sum((Cost_Fact + Profit_Fact)))` and `Avg((Cost_Metric + Profit_Metric))`.

smart metric A setting for a compound metric that allows you to change the default evaluation order. Smart metrics calculate subtotals on the individual elements of the compound metric. For example, a smart metric uses the formula `Sum(Metric1)/Sum(Metric2)` rather than `Sum(Metric1/Metric2)` when calculating subtotals on a report.

sort Arranging data according to some characteristic of the data itself (alphabetical descending, numeric ascending, and so forth).

See also:

- **drill**
- **page-by**
- **pivot**
- **subtotal**

source system Any system or file that captures or holds data of interest.

subtotal A totaling operation performed for a portion of a result set.

See also:

- **drill**
- **page-by**
- **pivot**
- **sort**

system hierarchy The superset hierarchy containing all attributes in a project. Unlike a browse hierarchy, it is not explicitly created but is automatically deduced by the MicroStrategy platform from all information available to it.

Compare **user hierarchy**.

table size The estimated size of a database table in terms of number of rows.

table alias One type of logical table. A table alias is created outside of the Warehouse Catalog and points directly to a physical table. A table alias can have a different name from the physical table. One physical table can have more than one table alias.

See also **logical table**.

template A MicroStrategy object that serves as a base on which you can build other objects of the same type. You can create a template for almost any kind of MicroStrategy object, such as filters or reports.

- Object templates allow you to start with a predefined structure when creating a new object. You can use object templates for many MicroStrategy objects, including metrics, documents, reports, and report templates.
- Report templates define the layout of general categories of information in a report. In a report template, you specify the information that you want to retrieve from your data source, and the way that you want the data to be displayed in Grid view. A report template does not include filter information. Report templates are often referred to as just as templates.

threshold Used to create conditional formatting for metric values. For example, if revenue is greater than \$200, format that cell to have a blue background with bold type.

transformation A schema object that maps a specified time period to another time period, applying an offset value, such as current month minus one month. Although the vast majority are based on

time, a transformation can also map different objects, such as defunct product codes to new ones.

Time transformations are used in metrics to compare values at different times, such as this year versus last year or current date versus month-to-date.

transformation metric An otherwise simple metric that takes the settings of the transformation applied to it. For example, a metric calculates total sales. Add a transformation for last year and the metric now calculates last year's total sales.

user hierarchy Named sets of attributes and their relationships, arranged in specific sequences for a logical business organization. They are user-defined and do not need to follow the logical model.

Compare **system hierarchy**.

view definition Report execution steps which represent how the data is viewed and manipulated in the Intelligence Server. The view definition determines how the final report data set generated in the data definition steps is manipulated.

INDEX

Numerics

- 3D surface graph [416](#)
 - Auto Arrange [488](#)
 - best practices [417](#)
 - example [416](#)

A

- access control list
 - Freeform SQL report and [632](#)
 - Query Builder report and [727](#)
- across level subtotals [289](#)
- add-in function. See custom function.
- Advanced Metric Assistant, creating multiple metrics with [99](#)
- advanced qualification [128](#)
 - activating [128](#)
 - custom expression [154](#), [156](#)
 - joint element list [172](#)
- advanced subtotals
 - across a level [289](#)
 - by position [289](#)
 - group by [290](#)
- aggregation, grouping in a level metric [23](#)
- Alerter. See threshold.
- alias for a metric column [123](#)
- alignment formatting default [894](#)
- analysis, time-series [92](#)
- Analytical Engine, null checking in [863](#)
- Apply function [164](#)
 - custom expression [164](#)
 - example [167](#), [168](#), [169](#)
 - handling by the SQL Engine [120](#)
 - syntax [166](#)
- ApplyAgg [165](#)
- ApplyComparison [165](#)
- applying series effect to a graph report [370](#)
- ApplyLogical [165](#)
- ApplyOLAP [165](#)
- ApplySimple [165](#)
- area graph [382](#)
 - best practices [383](#)
 - combination graph [407](#)
 - curved effect [383](#)
 - example [382](#)
- association
 - drill map [238](#)
 - level [239](#)
 - removing [240](#)

- association rules analysis [793](#)
 - example [848](#)
- attribute [defined on 3](#)
 - element [3](#)
 - on a report [336](#)
 - grouping multiple in a joint element list [172](#)
 - on a report [336](#)
 - virtual (consolidation) [207](#)
- attribute element list qualification [127](#)
- attribute element prompt
 - link and [568, 569](#)
- attribute form qualification [127](#)
- attribute form selection option for intermediate passes [875](#)
- attribute join type [335, 336](#)
 - changing [339](#)
 - object prompt and [341](#)
- attribute level of a set qualification [146](#)
- attribute qualification [127](#)
 - attribute element list qualification [127](#)
 - attribute form qualification [127](#)
 - Date and Time Editor [136](#)
 - date offset [135](#)
 - dynamic date [135](#)
 - exported filter element [145](#)
 - imported filter element [143](#)
 - prompted [175](#)
- attribute qualification prompt [245](#)
 - in a filter [175](#)
- attribute selection option for intermediate passes [875](#)
- attribute-to-attribute qualification filter [130](#)
 - creating [130, 160](#)
 - custom expression in [160](#)
 - example [131, 133, 160](#)
- Auto Arrange in a graph [488](#)
- automatic layout in a graph [484](#)

- autostyle [328](#)
 - default format [893](#)
 - distributing [329](#)
- average
 - moving [115](#)
 - running [116](#)
- axis formatting
 - graph [480](#)
 - report [324](#)

B

- banding [defined on 188](#)
 - count [188](#)
 - formatting [324](#)
 - formatting default [896](#)
 - metric value [189](#)
 - output level [188](#)
 - point [189](#)
 - qualification [188](#)
 - size [188](#)
 - type [188](#)
- bar graph [384](#)
 - best practices [391](#)
 - combination graph [407](#)
 - example [385](#)
- bevel effect on a graph object [370](#)
- border of a report [325](#)
 - default formatting [895](#)
- box and whisker plot. See boxplot.
- boxplot [427](#)
 - best practices [430](#)
 - example [428](#)
- browsing
 - project order in Developer and Web [913](#)
 - standards in MicroStrategy [911](#)
- bubble chart [405](#)
 - best practices [407](#)

- example [406](#)
- budgeting graph
 - lipstick bar graph [387](#)
 - overlapping bar graph [387](#)
- bulk exporting [363](#)
- by position subtotals [289](#)

C

- cache [defined on 361](#)
 - default setting [362](#)
 - disabling [362](#)
 - disabling for a report [362](#)
 - enabling [362](#)
 - enabling for a report [362](#)
 - Intelligent Cube [288, 362](#)
 - invalidating [362](#)
 - report [362](#)
- Cartesian join [866](#)
 - warning [866](#)
- categories [defined on 378](#)
 - in a graph [378](#)
 - specifying number of [380](#)
- ChangeView command [903](#)
- character sort order [912](#)
- choose from all attributes in a hierarchy
 - prompt [244](#)
- choose from attribute element list
 - prompt [245](#)
- circle graph. See pie chart.
- cluster analysis [790](#)
 - example [838](#)
- color palette for a graph [436](#)
- color style for a graph [436](#)
- column alias for a metric [123](#)
- column subtotal formatting [324](#)
- combination graph [407](#)
- comparison operator [889](#)
- compound metric [defined on 13, 95](#)

- count function [114](#)
- moving average [115](#)
- moving sum [116](#)
- non-group function [112](#)
- n-tile [117](#)
- rank function [112](#)
- running average [116](#)
- running sum [116](#)
- smart metric [98](#)
- smart subtotals [98](#)
- condition. See filter.
- conditional metric [63](#)
 - advanced options [70, 89](#)
 - default [87](#)
- combining embedding method and re-move related elements
 - settings [77](#)
- creating [69](#)
 - Advanced Metric Assistant [99](#)
- embedding method [70](#)
- level and condition [65](#)
- link and [569, 574](#)
- multiple conditions [65](#)
- remember option setting [87](#)
- removing related report filter elements [75](#)
- report filter interaction [70](#)
- conditional predictive input metric [782](#)
 - creating [783](#)
- confusion matrix [789](#)
- consolidation [defined on 207](#)
 - custom group comparison [221](#)
 - element. See consolidation element.
 - evaluation order [213, 349](#)
 - evaluation order error example [349](#)
 - formatting [322](#)
 - importing an element [212](#)
 - row level math [208](#)
 - SQL generation and [214](#)

- subtotal [215](#)
- virtual attribute [207](#)
- consolidation element [defined on 209](#)
- formatting [209, 325](#)
- from attributes in different hierarchies [212](#)
- from different levels of the same hierarchy [211](#)
- from the same attribute [211](#)
- importing [212](#)
- using existing [212](#)
- contribution metric [15](#)
- conversion rate, prompting for [250](#)
- converting currency [250](#)
- count distinct [114](#)
- count function [114](#)
- count metric [114](#)
- currency conversion [250](#)
- current date offset [135](#)
- curved effect
 - applying to a graph object [370](#)
 - applying to a line graph [394](#)
 - applying to an area graph [383](#)
- custom expression [154](#)
 - Apply function in [164](#)
 - attribute-to-attribute qualification filter [160](#)
 - qualification filter [156](#)
 - example [158](#)
 - string manipulation in [158](#)
 - relationship qualification filter [161, 162](#)
 - string manipulation in [157](#)
 - using a date [158](#)
- custom format [310](#)
 - character/text data [313](#)
 - color [315](#)
 - conditional [316](#)
 - currency [315](#)
 - date [313](#)
 - example [316](#)
 - numeric data [311](#)
 - time [313](#)
- custom function [119](#)
 - R Integration Pack [120](#)
- custom graph [408](#)
- custom group [defined on 182](#)
 - advantages [187](#)
 - banding qualification [188](#)
 - consolidation comparison [221](#)
 - creating [193](#)
 - creating multiple simultaneously [193](#)
 - element [183](#)
 - element header [183](#)
 - element header position [195](#)
 - example [185](#)
 - formatting [322, 325](#)
 - hierarchical display [191](#)
 - keeping the group structure [199](#)
 - prompt [201](#)
 - report filter interaction with [200](#)
 - sorting [197](#)
 - by item metric value [199](#)
 - sorting by metric values of items [199](#)
 - SQL generation and [187](#)
 - store inventory example [185](#)
 - structure on report [191](#)
 - subtotals [196](#)
 - position [196](#)
 - totals [196](#)
 - position [196](#)
- custom prompt style for MicroStrategy Web [261](#)
- custom report subtotals [294](#)
- customizing a drill map [228](#)

D

data access using a security filter

Freeform SQL [633](#)

Query Builder [728](#)

data definition [defined on](#) [286](#)

evaluation order [344](#)

object [287](#)

data mart [defined on](#) [743](#)

creating [744](#)

governing settings [758](#)

metric alias [123](#)

objective of [743](#)

optimizing [754](#)

performance parameters [758](#)

report [defined on](#) [744](#)

running a report against [756](#)

SQL statement [759](#)

table [defined on](#) [744](#)

used as a source table [749](#)

data mining [defined on](#) [763](#)

association rules analysis [793](#)

example [848](#)

cluster analysis [790](#)

example [838](#)

confusion matrix [789](#)

dataset report [772](#)

decision tree analysis [791](#)

example [841](#)

example [831](#), [836](#)

exponential regression [811](#)

integrating with third-party
application [823](#)

linear regression [811](#)

example [832](#)

logistic regression [789](#)

example [836](#)

market basket analysis [793](#)

example [848](#)

multivariate regression model [788](#)

PMML [768](#)

regression [785](#)

seasonal regression [832](#)

standard regression model [816](#)

time series analysis [798](#)

training metric [809](#)

Training Metric Wizard [810](#)

training the model [809](#)

tree regression model [815](#)

univariate regression model [788](#)

workflow [769](#)

database

connecting Freeform SQL to [732](#)

connecting Query Builder to [732](#)

database instance

VLDB property [882](#)

web service [639](#)

database scoring [764](#)

database-specific function [164](#)

dataset report for data mining [772](#)

guidelines for creating [774](#)

non-MicroStrategy [784](#)

date

dynamic [135](#)

in a custom expression [158](#)

in an Apply function [169](#)

Date and Time Editor [136](#)

date offset [135](#)

decision tree analysis [791](#)

example [841](#)

default

drill path [231](#)

evaluation order [343](#)

inherited value [860](#)

link [565](#)

Design View [274](#)

detail. See fact.

- dimensional subtotals [104](#)
 - dimensionality. See level.
 - distributing reports [352](#)
 - Object Templates folder [361](#)
 - Public Objects folder [361](#)
 - report [361](#)
 - re-using an object [352](#)
 - Shared Reports folder [361](#)
 - Distribution Services bulk exporting reports [363](#)
 - division by zero [863](#)
 - document
 - Freeform SQL report in [610](#)
 - moving sum example [116](#)
 - n-tile function example [117](#)
 - Query Builder report in [698](#)
 - rank example [113](#)
 - running sum example [116](#)
 - widget in [507](#)
 - drill map [225](#), [226](#)
 - See also drilling.
 - association [238](#)
 - level [239](#)
 - removing [240](#)
 - template [240](#)
 - creating [228](#)
 - custom [228](#)
 - default [226](#), [238](#)
 - deleting [226](#)
 - destination [229](#)
 - drill path type [229](#)
 - drill path. See also drill path. [226](#)
 - grid unit level [239](#)
 - project [240](#)
 - project default [226](#)
 - project level [239](#)
 - report level [239](#)
 - template [240](#)
 - template level [239](#)
 - drill path [226](#)
 - base template [236](#)
 - creating [228](#)
 - custom [228](#)
 - default [227](#), [231](#)
 - destination [229](#)
 - display mode [237](#)
 - display name [231](#)
 - drilled-on object [236](#)
 - filter interaction [232](#)
 - keep parent [236](#)
 - page-by [236](#)
 - priority [231](#)
 - properties [230](#)
 - set name [231](#)
 - system hierarchy [227](#)
 - threshold [237](#)
 - type [229](#)
 - VLDB properties [238](#)
 - drilling [defined on 226](#)
 - drill map. See also drill map. [226](#)
 - drill path. See also drill path. [226](#)
 - filter interaction [227](#), [232](#)
 - Freeform SQL report and [627](#)
 - Query Builder report and [725](#)
 - shortcut to an existing drill map [230](#)
 - to a template [230](#)
 - dropping temp table method [878](#)
 - dynamic date [135](#)
 - Date and Time Editor [136](#)
 - prompt and [139](#)
 - dynamic graph title [455](#)
- ## E
- Editor command [904](#)
 - element of a consolidation. See also consolidation element. [209](#)

- embedded
 - filter [128, 357](#)
 - See also shortcut-to-a-filter qualification.
 - shortcut vs. [177](#)
 - template [354](#)
- embedding method [70](#)
 - changing default [87](#)
 - combining with remove related elements setting [77](#)
 - example [72, 87](#)
- empty object template [358](#)
 - user-defined [359](#)
- equi-join [868](#)
- evaluation order
 - consolidation [213](#)
 - example [349](#)
 - data [344](#)
 - example [345, 349](#)
 - default [343](#)
 - defining [344](#)
 - derived metric [345](#)
 - incompatibility error example [345, 349](#)
 - report [342](#)
 - view [344](#)
 - example [345, 349](#)
- example
 - attribute-to-attribute qualification
 - filter [131, 133](#)
 - currency conversion [250](#)
 - custom expression qualification
 - filter [158](#)
 - custom format [316](#)
 - data mining. See example, data mining.
 - dynamic date [136](#)
 - evaluation order data [345, 349](#)
 - evaluation order error [345, 349](#)
 - First function [105](#)
 - graph. See example, graph.
 - level prompt [252](#)
 - link. See example, link.
 - metric qualification [280](#)
 - metric. See example, metric.
 - nonaggregatable metrics and level grouping [29](#)
 - relationship qualification filter [163](#)
 - Report Details pane [510](#)
 - report details. See example, report details.
 - running sum [116](#)
 - SQL aggregation [2](#)
 - subtotals across level [292](#)
 - subtotals by position [290](#)
 - subtotals group by [293](#)
 - subtotals, user-defined [105](#)
 - user-defined subtotals [105, 108](#)
 - value prompt [250](#)
 - view evaluation order [345, 349](#)
 - year-to-date transformation [93](#)
- example, data mining [831, 836](#)
 - cluster analysis [838](#)
 - decision tree analysis [841](#)
 - linear regression [832](#)
 - logistic regression [836](#)
 - seasonal regression [832](#)
- example, graph
 - 3D surface graph [416](#)
 - area graph [382](#)
 - bar graph [385](#)
 - boxplot [428](#)
 - bubble chart [406](#)
 - funnel graph [421](#)
 - gauge graph [418](#)
 - graph style [381](#)
 - histogram [415](#)
 - line graph [393](#)

- pareto chart [424](#)
 - pie chart [395](#)
 - polar chart [401](#)
 - radar chart [404](#)
 - scatter plot [398](#)
 - stock graph [410](#)
 - example, link
 - answering a hierarchy prompt [577](#)
 - answering a target prompt with source prompt answers [571](#)
 - conditional metric and [574](#)
 - ignoring a target prompt [572](#)
 - linking reports [560](#)
 - multiple links [565](#)
 - prompt answer for any other prompts not listed [581](#)
 - example, metric
 - compound [96](#)
 - conditional metric with multiple qualifications [65](#)
 - embedding method [72, 87](#)
 - level [37](#)
 - level grouping with a nonaggregatable metric [29](#)
 - rank [113](#)
 - removing the report level [35](#)
 - reusing metrics with the Filter setting [56](#)
 - transformation [92](#)
 - example, report details [517](#)
 - filter details [545, 547](#)
 - alias display [551](#)
 - content of [540](#)
 - drill filter [542](#)
 - filter type [539](#)
 - logical operator display [548](#)
 - prompt details [531](#)
 - report details delimiters [514](#)
 - report details preferences [556](#)
 - template details [522, 524](#)
 - Excel
 - connecting Freeform SQL to [733](#)
 - connecting Query Builder to [733](#)
 - Execute command [905](#)
 - ExecuteDocument command [906](#)
 - ExecuteReport command [906](#)
 - exponential regression [811](#)
 - exporting [363](#)
 - filter element [145](#)
 - graph [504](#)
- ## F
- fact [defined on 2](#)
 - fallback table type [879](#)
 - filter [defined on 126, 275](#)
 - qualification types. See filter qualification types.
 - adding to a report [177](#)
 - attribute level [146](#)
 - comparison operator [889](#)
 - conditional metric and [70](#)
 - creating. See filter, creating.
 - custom expression qualification in [156](#)
 - date offset [135](#)
 - drilling and [227](#)
 - dynamic date [135, 139](#)
 - embedded [177, 357](#)
 - exporting a filter element [145](#)
 - Freeform SQL report and [623](#)
 - importing a filter element [143](#)
 - linked [177](#)
 - local [177](#)
 - logical operator [127, 886](#)
 - metric [63](#)
 - metric and report interaction [70, 75, 87](#)
 - operator [885](#)

- output level [146](#)
- pattern operator [891](#)
- prompted. See also filter, prompted. [174](#)
- qualification types. See filter qualification types.
- Query Builder report and [717](#)
- rank and percent operator [890](#)
- report and metric interaction [70, 75, 87](#)
- report as filter [283](#)
- security [633, 728](#)
- shortcut to [177, 357](#)
- stand-alone [177](#)
- filter details [537](#)
 - alias display [551](#)
 - attribute element list qualification display [544](#)
 - example [545](#)
 - attribute form and set qualification display [547](#)
 - configuring [551](#)
 - content of [538, 540](#)
 - example [540](#)
 - drill filter [542](#)
 - example [542](#)
 - filter type [538, 539](#)
 - example [539](#)
 - logical operator [548](#)
 - report details preferences interaction [538](#)
- filter object prompt [176](#)
- filter qualification types [127](#)
 - advanced qualification [129](#)
 - attribute qualification [127](#)
 - attribute-to-attribute qualification [160](#)
 - joint element list qualification [172](#)
 - relationship qualification [161](#)
 - set qualification [127](#)
 - shortcut-to-a-filter qualification [128](#)
 - shortcut-to-a-report qualification [128](#)
- filter, creating
 - attribute qualification using a dynamic date [137](#)
 - attribute-to-attribute qualification [130, 160](#)
 - custom expression qualification [156](#)
 - dynamic date based on a prompt answer [141](#)
 - joint element list [173](#)
- filter, prompted [174](#)
 - attribute qualification prompt [175](#)
 - filter definition prompt [244](#)
 - metric qualification prompt [175](#)
 - object prompt [176](#)
 - report object prompt [176](#)
- find and replace [329](#)
 - privilege [330](#)
- First function [105, 119](#)
- Flash Mode graph visualization [507](#)
- font
 - anti-aliasing applied to graph text [371](#)
 - formatting default [895](#)
- Format Cells dialog box [310](#)
- formatting
 - autostyle [328](#)
 - custom [310](#)
 - default project [359](#)
 - default value [893](#)
 - graph. See also graph formatting. [434](#)
 - grid unit [318](#)
 - layer. See also formatting layer. [317](#)
 - report [309](#)
 - threshold [320](#)
 - using empty object template [359](#)
 - zone [defined on 318](#)
- formatting layer [defined on 317](#)

- all metrics [324](#)
 - axis [324](#)
 - banding [324](#)
 - column subtotals [324](#)
 - consolidation [325](#)
 - custom group [325](#)
 - grid unit [324](#)
 - metric [323](#)
 - metrics label [324](#)
 - order of layers [322](#)
 - report border [325](#)
 - report metric [324](#)
 - row subtotals [325](#)
 - threshold [325](#)
 - formula join type for a compound metric [99](#)
 - Freeform SQL functionality [604](#)
 - Freeform SQL report [604](#)
 - access control list [632](#)
 - common table expression [608](#)
 - creating. See Freeform SQL report creation.
 - derived table [608](#)
 - document [610](#)
 - drilling [627](#)
 - filter [623](#)
 - managed object [735](#)
 - multi-pass SQL [607](#)
 - MultiSource Option [628](#)
 - prompt [624](#)
 - query syntax [606](#)
 - security filter [633](#)
 - security for data access [632](#)
 - shortcut-to-a-report qualification [628](#)
 - SQL support [607](#)
 - standard report versus [609](#)
 - Freeform SQL report creation
 - from a database [732](#)
 - from a text file [734](#)
 - from a web service [638](#)
 - from an Excel file [733](#)
 - using a stored procedure [616](#)
 - Freeform SQL used to create an Intelligent Cube [740](#)
 - full outer join support [868](#)
 - function
 - add-in. See custom function.
 - Apply [164](#)
 - ApplyAgg [165](#)
 - ApplyComparison [165](#)
 - ApplyLogical [165](#)
 - ApplyOLAP [165](#)
 - ApplySimple [165](#)
 - custom [119](#)
 - database-specific [164](#)
 - First [119](#)
 - Last [119](#)
 - OLAP [116](#)
 - Pass-through [164](#)
 - plug-in [119](#)
 - funnel graph [421](#)
 - best practices [423](#)
 - example [421](#)
- ## G
- gant chart [430](#)
 - gauge graph [418](#)
 - best practices [420](#)
 - displaying multiple gauges [420](#)
 - example [418](#)
 - gradient, applying to a graph object [370](#)
 - grand totals
 - default [102](#)
 - displaying [102, 291](#)
 - function [103](#)
 - total subtotal function [103](#)

- graph **367**
 - adding data to **372**
 - Auto Arrange **488**
 - best practices **368**
 - categories **378**
 - categories and series **380**
 - copying into a third-party application **505**
 - dynamic title **455**
 - exporting **504**
 - footnotes **454**
 - creating **454**
 - formatting. See graph formatting.
 - insufficient data for **377**
 - label. See graph label.
 - layout. See graph layout.
 - legend. See graph legend.
 - moving an object on **378**
 - object placement in **376**
 - resizing **488**
 - scaling **484, 489**
 - selector and **370**
 - series **378**
 - sizing **484**
 - thresholds **446, 449**
 - title **454**
 - creating **454**
 - variables for a title **455**
 - viewing **371**
- graph formatting
 - axis label **480**
 - background picture **453**
 - bevel effect **453**
 - color **436**
 - effects **436**
 - footnotes **454**
 - gradient **452**
 - gradient (using MicroStrategy Web) **444**
 - graph object manual format **451**
 - graph series (using MicroStrategy Web) **442**
 - label **475**
 - layout **484**
 - legend **476**
 - metric color **437**
 - inheriting **441**
 - numeric values **468**
 - inheriting **469**
 - pattern **453**
 - positioning a chart element **484**
 - shading **436**
 - texture **453**
 - thresholds **446**
 - example **449**
 - title **454, 456**
 - tooltip **481**
 - transparency **452**
- graph label
 - displaying **475**
 - formatting **475**
- graph layout **484**
 - Auto Arrange **488**
 - automatic **484**
 - automatic vs. manual **485**
 - example **487**
 - manual **486**
 - selecting **485**
- graph legend
 - displaying
 - using MicroStrategy Developer **476**
 - using MicroStrategy Web **478**
 - formatting **478**
 - using MicroStrategy Developer **478**
 - manually positioning **487**
- graph style
 - 3D surface **416**
 - advanced styles **408**

- applying custom [499](#)
- area [382](#)
- bar graph [384](#)
- boxplot [427](#)
- bubble chart [405](#)
- budgeting graph [387](#)
- combination [407](#)
- creating custom [499](#)
- custom [408](#)
- example and usage scenario [381](#)
- funnel [421](#)
- gant chart [430](#)
- gauge graph [418](#)
- histogram [414](#)
- insufficient data for [377](#)
- line graph [392](#)
- lipstick bar graph [387](#)
- minimum object requirements for [374](#)
- object placement in [376](#)
- overlapping bar graph [387](#)
- pareto chart [424](#)
- pie chart [395](#)
- polar chart [400](#)
- radar chart [402](#)
- report object placement for [376](#)
- scatter plot [398](#)
- selecting [376](#)
- stock graph [409](#)
- Graph view [371](#)
- Grid Graph view [368, 372](#)
- grid unit [defined on 318](#)
 - formatting [318, 324](#)
 - subtotal formatting [319](#)
- Grid/Graph, link in [563](#)
- group by
 - ID attribute [877](#)
 - non-ID attribute [878](#)
 - subtotals [290](#)

- grouping elements in a joint element
 - list [172](#)
- growth percentage [92](#)

H

- hierarchical sorting [298](#)
- hierarchy as the target of a level metric [47](#)
- hierarchy prompt and linking [570, 577](#)
- histogram [414](#)
 - best practices [415](#)
 - example [415](#)
- homepage in MicroStrategy
 - Developer [900](#)
- HTTP authentication for an XQuery statement [644](#)

I

- importing
 - data [6](#)
 - filter element [143](#)
 - table [705](#)
- inner join [121](#)
- integer constant in a metric [861](#)
- Intelligent Cube [defined on 288](#)
 - cache [362](#)
 - Freeform SQL [740](#)
 - personal [288](#)
 - Query Builder [740](#)
 - report cache [362](#)
 - shared [288](#)
- intermediate table type [879](#)
- internationalization, converting currency for [250](#)
- inventory metric [23](#)

J

- join

- attribute [335](#)
 - Cartesian [866](#)
 - equi-join [868](#)
 - formula [99](#)
 - inner [121](#)
 - metric [121](#)
 - outer [121](#)
 - outer join support VLDB property [868](#)
 - Query Builder report and [706](#)
 - report [121](#)
 - type [99](#)
 - joint element list qualification [172](#)
- K**
- key performance indicator (KPI). See fact.
- L**
- label on a graph. See also graph label. [474](#)
- Last function [119](#)
- layout of a graph. See also graph layout. [484](#)
- legend on a graph. See also graph legend. [476](#)
- level [defined on 14](#)
- grouping [23](#)
 - grouping for a nonaggregatable metric [23](#)
 - metric. See also level metric. [14](#)
 - prompt. See also level prompt [251](#)
 - properties [17](#)
 - displaying [17](#)
 - report level [34](#)
 - removing [35](#)
 - target [22](#)
- level metric [14](#)
- advanced options [53, 59, 89](#)
 - aggregation in [23](#)
 - allowing other users to add extra units
 - to the definition [59](#)
 - applying the metric filter to the metric calculation [53](#)
 - attribute level [22](#)
 - elements of [16](#)
 - filter setting [53](#)
 - grouping [23](#)
 - hierarchy as the target [47](#)
 - properties [17](#)
 - property symbols defined [17](#)
 - removing the report level [35](#)
 - report level [14, 34](#)
 - security filter [41](#)
 - target [22](#)
- level prompt [251](#)
- creating [254](#)
 - example [252](#)
 - object prompt vs. [252](#)
- line graph [392](#)
- applying a curved effect [394](#)
 - best practices [394](#)
 - combination graph [407](#)
 - example [393](#)
- linear regression [811](#)
- example [832](#)
- link [559, defined on 563](#)
- components [564](#)
 - conditional metric and [569](#)
 - copying [587](#)
 - creating [583](#)
 - prerequisites [583](#)
 - default [565](#)
 - in MicroStrategy Developer [565](#)
 - in MicroStrategy Web [566](#)
 - example [560](#)
 - MDX cube report as the source [576](#)
 - multiple [563](#)
 - default link [565](#)
 - example [565](#)

- prerequisites [562](#)
- prompt and [564](#)
- prompt answer method. See also
prompt answer method for a
link. [567](#)
- source [563](#)
- target [564](#)
- linked template. See shortcut to a tem-
plate.
- linking reports and documents [559](#)
 - See also link.
 - comparison of [564](#)
 - example [560](#)
 - prerequisites [562](#)
- lipstick bar graph [387](#)
- local filter [177](#)
- logical operator
 - exclusion [888](#)
 - filter [127](#)
 - functional description [886](#)
 - intersection [887](#)
 - union [887](#)
- logistic regression [789](#)
 - confusion matrix [789](#)
 - example [836](#)

M

- managed object [defined on 735](#)
 - editing [739](#)
- market basket analysis
 - data mining [793](#)
 - example [848](#)
- maximum SQL size [882](#)
- MDX cube report, linking from [576](#)
- MDX cube, Intelligent Cube based on [740](#)
- merging header cells [896](#)
- metric [4](#), [defined on 12](#)
 - column alias [123](#)

- compound [13, 95](#)
- conditional. See also conditional
metric. [63](#)
- contribution [15](#)
- count [114](#)
- creating multiple simultaneously [99](#)
- creating, using Advanced Metric
Assistant [99](#)
- default subtotal function [103](#)
- dimensionality. See level.
- division by zero [863](#)
- formula [12](#)
- formula join type [99](#)
- function plug-in [119](#)
- integer constant in [861](#)
- inventory [23](#)
- join [121](#)
- join default [122](#)
- join type [99, 122](#)
- level. See also level metric. [14](#)
- nested [60](#)
- nonaggregatable [23](#)
- null checking [862](#)
- predictive [770](#)
- predictive input [776](#)
- prompted [63](#)
- simple [12](#)
- smart [98](#)
- sorting hierarchically [298](#)
- subtotal dimensionality [863](#)
- subtotals [102, 289](#)
- total subtotal function [103](#)
- transformation. See also transforma-
tion metric. [90](#)
- type [12](#)
- VLDB properties [123, 861](#)
- zero checking [863](#)

Metric Editor

- condition [67](#)
- level [17](#)
- metric join type [121](#)
- Subtotals/Aggregation tab [103](#)
- transformation [94](#)
- metric filter
 - conditional metric and [70](#)
 - report filter and [70](#)
- metric formatting
 - defining a graph color [437](#)
 - metric headers [324](#)
 - metric level [323](#)
 - metric values [324](#)
 - metrics label [324](#)
- Metric Formula Join Type dialog box [99](#)
- metric level. See level metric.
- metric qualification [127](#), [280](#)
 - advanced options [151](#)
 - example [280](#)
 - in relationship qualification filter [163](#)
 - merging attribute qualifications [151](#)
 - metric-to-metric comparison [150](#)
 - output level [148](#)
 - prompt [245](#)
 - prompted [175](#)
 - resolving in a report [151](#)
- metric set qualification. See metric qualification.
- metric-based predictive input metric [780](#)
- metric-to-metric comparison [150](#)
- MicroStrategy Developer
 - anchor [900](#)
- commands [899](#)
 - ChangeView [903](#)
 - Editor [904](#)
 - Execute [905](#)
 - ExecuteDocument [906](#)
 - ExecuteReport [906](#)
 - Open [907](#)
 - Reset [908](#)
 - Shortcut [909](#)
 - syntax [900](#)
 - viewing [902](#)
- setting the homepage [900](#)
- MicroStrategy Transaction Services [675](#)
 - report [676](#)
- MicroStrategy Web
 - browsing and sorting order for projects [913](#)
 - configuring graph image format for [503](#)
 - prompt display in [258](#)
- moving
 - average [115](#)
 - sum [116](#)
- multiple links [563](#)
- multiple-key sorting [298](#)
- MultiSource Option
 - Freeform SQL report [628](#)
 - Query Builder report [726](#)
- multivariate regression model [788](#)

N

- nested aggregation metric [60](#)
- nested metric [60](#)
- nonaggregatable metric [defined on 23](#)
 - grouping [23](#)
- non-group function [112](#)
 - moving average [115](#)
 - moving sum [116](#)
- n-tile [117](#)

- rank [112](#)
- running average [116](#)
- running sum [116](#)
- n-tile function [117](#)
 - example [117](#)
- null checking [862](#)
 - Analytical Engine [863](#)
- null value sorting [306](#)
- number formatting default [894](#)

O

- object name delimiter for report details [514](#)
- object prompt [246](#)
 - attribute join type and [341](#)
 - level prompt vs. [252](#)
- object re-use [352](#)
- object template [357](#)
 - default [360](#)
 - empty [358](#)
 - making available to users [360](#)
 - Object Templates folder [361](#)
 - user-defined empty [359](#)
 - viewing folder [361](#)
- object. See attribute.
- OLAP function [116](#)
- Open command [907](#)
- operator
 - comparison [889](#)
 - filter [885](#)
 - logical [886](#)
 - pattern [891](#)
 - rank and percent [890](#)
- optimizing a data mart [754](#)
- Oracle SQL hint [878](#)
- outer join [121](#)
- output level
 - banding [188](#)
 - filter [146](#)
 - metric qualification [148](#)
 - relating to the filter qualification [148](#)
 - relationship qualification [147](#)
 - set qualification [146](#)
- overlapping bar graph [387](#)

P

- pareto chart [424](#)
 - best practices [427](#)
 - example [424](#)
- Pass-through expression. See Apply function.
- Pass-through function. See Apply function.
- pattern formatting default [895](#)
- pattern operator [891](#)
- performance indicator. See fact.
- personal Intelligent Cube [defined on 288](#)
- pie chart [395](#)
 - best practices [397](#)
 - example [395](#)
- Plug-In Wizard [119](#)
- PMML [defined on 768](#)
- polar chart [400](#)
 - best practices [402](#)
 - example [401](#)
- predesigned report
 - accessing [361](#)
- predictive input metric [776](#)
 - conditional [782](#)
- predictive metric
 - aggregating [828](#)
 - attribute as input [776](#)
 - conditional metric as input [782](#)
 - in a report [829](#)
 - level metric as input [780](#)
 - performance [770](#)
- predictive model

- creating [785](#)
- importing [823](#)
- workflow [808](#)
- Predictive Model Markup Language. See PMML.
- Predictive Model Viewer [830](#)
- prerequisites for Advanced Reporting Guide [xx](#)
- preserving all final pass result elements [868](#)
- preserving all lookup table elements [869](#)
- priority of a drill path [231](#)
- project sorting and browsing order [913](#)
- project time line. See gantt chart.
- prompt [4](#), [defined on](#) [241](#)
 - attribute qualification [245](#)
 - choose from all attributes in a hierarchy [244](#)
 - choose from attribute element list [245](#)
 - custom group and [201](#)
 - custom prompt style for MicroStrategy Web [261](#)
 - display style for MicroStrategy Web [259](#)
 - dynamic date filter and [139](#)
 - expression properties for MicroStrategy Web [265](#)
 - filter [174](#)
 - filter definition [244](#)
 - Freeform SQL report and [624](#)
 - function and [243](#)
 - level [251](#)
 - link and. See prompt answer method.
 - metric qualification [245](#)
 - MicroStrategy Web display [258](#)
 - object [246](#)
 - Query Builder report and [720](#)
 - scheduled report and [243](#)
 - search object [246](#)
 - style properties for MicroStrategy Web [262](#)
 - system prompt [258](#)
 - type [243](#)
 - user login system prompt [258](#)
 - value [249](#)
- prompt answer method for a link [564](#), [567](#)
 - All valid units [570](#), [577](#)
 - example [577](#)
 - Answer with the same prompt [568](#), [570](#)
 - example [571](#)
 - Any other prompts [564](#), [581](#)
 - example [581](#)
 - by prompt type
 - attribute element prompt [568](#), [569](#)
 - hierarchy prompt [570](#), [577](#)
 - value prompt [568](#)
 - Current unit [570](#), [577](#)
 - example [577](#)
 - Default answer [568](#)
 - Dynamically [568](#)
 - example [560](#)
 - Empty answer [569](#), [571](#)
 - example [572](#)
 - Prompt user [568](#)
 - selecting [585](#)
 - Static element list [569](#), [574](#)
 - example [574](#)
 - MDX cube data and [576](#)
- prompt details [530](#)
 - configuring [535](#)
 - example [531](#)
 - report details preferences
 - interaction [533](#)
- prompted
 - custom group [201](#)
 - filter [174](#), [175](#)
 - attribute qualification [175](#)
 - shortcut-to-a-filter

- qualification [176](#)
 - shortcut-to-a-report qualification [176](#)
- metric [63](#)
- Public Objects folder [361](#)

Q

- qualification
 - advanced [129](#)
 - attribute [127](#)
 - attribute prompt [245](#)
 - banding [188](#)
 - filter [127](#)
 - logical operator [127](#)
 - metric [127](#)
 - prompted attribute [175](#)
 - prompted metric [175](#)
 - prompted shortcut-to-a-filter [176](#)
 - prompted shortcut-to-a-report [176](#)
 - relationship [127](#)
 - set [127](#)
 - shortcut-to-a-filter [128](#)
 - shortcut-to-a-report [128](#)
 - type [127](#)
- qualify on a metric prompt [245](#)
- Query Builder [604](#)
 - creating an Intelligent Cube [740](#)
- Query Builder report
 - access control list [727](#)
 - attribute form [716](#)
 - column [712](#)
 - creating [700](#)
 - from a database [732](#)
 - from a text file [704, 734](#)
 - from an Excel file [703, 733](#)
 - document [698](#)
 - drilling [725](#)
 - filter [717](#)

- join [706](#)
- managed object [735](#)
- MultiSource Option [726](#)
- prompt [720](#)
- security filter [728](#)
- security for data access [727](#)
- selecting a table [705](#)
- shortcut-to-a-report qualification [726](#)
- standard report versus [697](#)

R

- R (analytics) integration with MicroStrategy [120](#)
- R Integration Pack [120](#)
- radar chart [402](#)
 - best practices [405](#)
 - example [404](#)
- rank and percent operator [890](#)
- rank function [112](#)
- RDBMS function [164](#)
- regression [defined on 785](#)
- relating the output level and filter qualification [148](#)
- relation in a relationship qualification filter [163](#)
- relationship qualification [127, 162](#)
 - advanced options [153](#)
 - applying a qualification independently [153](#)
 - example [163](#)
 - output level [147](#)
 - filter relation and [148](#)
 - relationship table [164](#)
- relationship qualification filter. See relationship qualification.
- relationship set qualification. See relationship qualification.
- remember option setting (conditional metric) [87](#)

- removing related report filter elements in a metric [75](#)
 - default [87](#)
 - embedding method and [77](#)
- replace. See find and replace.
- report [4](#), [defined on 271](#)
 - cache [362](#)
 - data definition [286](#)
 - Design View [274](#)
 - distributing [352](#), [361](#)
 - embedded template [354](#)
 - evaluation order [342](#)
 - default [343](#)
 - defining [344](#)
 - execution flow [286](#)
 - exporting [363](#)
 - filter [275](#)
 - formatting. See also report formatting. [309](#)
 - Intelligent Cube [288](#)
 - join [121](#)
 - MicroStrategy Transaction Services [675](#)
 - object re-use [352](#)
 - Object Templates folder [361](#)
 - Public Objects folder [361](#)
 - reducing SQL passes [282](#)
 - report limit [276](#)
 - Shared Reports folder [361](#)
 - shortcut to [283](#)
 - sorting [298](#)
 - advanced [298](#)
 - hierarchical [298](#)
 - multiple-key [298](#)
 - SQL optimization [282](#)
 - subtotals. See also subtotals. [102](#)
 - Transaction Services [676](#)
 - validating changes to a template [356](#)
 - view definition [286](#)
 - VLDB properties [865](#)
 - web service data [659](#)
- report as filter. See shortcut-to-a-report qualification.
- report border formatting [325](#)
- report cache invalidated with find and replace [330](#)
- report creation [defined on 272](#)
- Report Data Options
 - attribute join type [335](#)
 - evaluation order [214](#), [344](#)
 - metric join type [121](#)
 - report limit [276](#)
- report description in the report details [516](#)
- report design [defined on 272](#)
 - multiple simultaneously [272](#)
- report details [509](#)
 - configuring [519](#)
 - disabling [511](#)
 - displaying [511](#)
 - enabling [511](#)
 - example [517](#)
 - filter details [537](#)
 - hiding [511](#)
 - level of configuration [511](#)
 - document [512](#)
 - order of precedence [512](#)
 - project level [512](#)
 - report level [512](#)
 - text field [511](#)
 - object name delimiter [514](#)
 - prompt details [530](#)
 - report description [516](#)
 - report details preferences and [512](#), [518](#)
 - template details [520](#)
 - types [513](#)
- Report Details pane [509](#)

- example 510
 - report details preferences 554
 - customizing 556
 - disabling 513
 - enabling 556
 - example 556
 - filter details interaction 538
 - prompt details interaction 533
 - report details interaction 512, 518
 - template details interaction 526
 - report filter
 - conditional metric and 70
 - custom group and 200
 - drilling and 227
 - removing related elements of 75
 - report limit and 276, 278
 - report formatting
 - autostyle 328
 - autostyle default 893
 - axis layer 324
 - banding layer 324
 - column subtotal layer 324
 - consolidation layer 325
 - custom 310
 - custom group layer 325
 - default values 893
 - formatting layers 317
 - order of 322
 - grid unit 318
 - grid unit layer 324
 - metric header layer 324
 - metric layer 323, 324
 - metric value layer 324
 - metrics label layer 324
 - report border layer 325
 - report metric layer 324
 - row subtotal layer 325
 - subtotal 319
 - threshold 320
 - threshold layer 325
 - zone 318
 - report level 14, 34
 - removing 35
 - report limit
 - report filter and 278
 - report filter versus 276
 - report object prompt filter 176
 - report post statement 872
 - report pre statement 872
 - report qualification. See shortcut-to-a-report qualification.
 - Reset command 908
 - resizing a graph 488
 - REST architecture 638
 - re-using an object 352
 - row subtotal formatting layer 325
 - running
 - average 116
 - sum 116
- ## S
- saving a report with a shortcut to a stand-alone filter 178, 355
 - scaling a graph 484, 489
 - scatter graph. See scatter plot.
 - scatter plot 398
 - best practices 399
 - example 398
 - scheduled report containing a prompt 243
 - scheduling bulk exports 363
 - scope of analysis. See Intelligent Cube.
 - search object 246
 - searching for dependents 353
 - seasonal regression example 832
 - security
 - Freeform SQL report 632

- Query Builder report [727](#)
- XQuery statement [644](#)
- security filter [defined on 633](#)
 - creating [635](#)
 - Freeform SQL and [633](#)
 - in a level metric [41](#)
 - Query Builder report and [728](#)
- segmentation [117](#)
- selector, controlling a graph with [370](#)
- series [defined on 378](#)
 - graph [378](#)
 - specifying the number of [380](#)
- set qualification [127](#)
 - attribute level [146](#)
 - metric qualification [127](#)
 - output level [146](#)
 - relationship qualification [127](#)
- Shared Reports folder [361](#)
- shortcut
 - to a filter [177](#)
 - saving a report with [178, 355](#)
 - to a report [283](#)
 - to a template [353](#)
 - validating changes [356](#)
- Shortcut command [909](#)
- shortcut-to-a-filter qualification [128](#)
 - advanced options [151](#)
 - merging attribute qualifications [151](#)
 - prompted [176](#)
 - resolving in a report [151](#)
- shortcut-to-a-report qualification [128](#)
 - example [283](#)
 - Freeform SQL reports [628](#)
 - prompted [176](#)
 - Query Builder report [726](#)
- simple metric [defined on 12](#)
- sizing a graph [484](#)
- smart metric [98](#)
- smart subtotals [98](#)
- SOAP protocol [638](#)
- sorting [298](#)
 - advanced [298](#)
 - attribute form [301](#)
 - character sort order [912](#)
 - custom group [197, 199](#)
 - default order [298](#)
 - hierarchical [298](#)
 - multiple attribute forms [301](#)
 - multiple-key [298](#)
 - null values [306](#)
 - project order in Developer and Web [913](#)
 - standards in MicroStrategy [911](#)
- source of a link [563](#)
- source table [749](#)
- SQL
 - common table expression [608](#)
 - creating a Transaction Services report [676](#)
 - date formatting [883](#)
 - derived table [608](#)
 - Freeform SQL report [604](#)
 - in a document [610](#)
 - standard report versus [609](#)
 - hint [878](#)
 - maximum size [882](#)
 - metric alias [123](#)
 - multi-pass SQL [607](#)
 - passes, reducing in a report [282](#)
 - query syntax [606](#)
 - report post statement [872](#)
 - report pre statement [872](#)
 - support [607](#)
 - table post statement [873](#)
 - VLDB property [859](#)
- stock graph [409](#)
 - best practices [412](#)

- example [410](#)
- string manipulation [158](#)
 - using a custom expression [157](#)
- sub query type [873](#)
- Subtotal Editor [104](#)
- subtotal function [103](#)
- subtotals [defined on 102, 289](#)
 - across a level [289](#)
 - example [292](#)
 - advanced [289](#)
 - applied level option [289](#)
 - by position [289](#)
 - example [290](#)
 - column subtotals formatting layer [324](#)
 - consolidation [215](#)
 - custom [294](#)
 - custom group [196](#)
 - dimensional [104](#)
 - Dimensionality Aware VLDB
 - property [863](#)
 - function
 - default [103](#)
 - multiple [104](#)
 - nested [104](#)
 - predefined [103](#)
 - standard subtotal [103](#)
 - group by [290](#)
 - example [293](#)
 - memory usage and [297](#)
 - metric dimensionality [863](#)
 - metric in the formula [104](#)
 - report [289](#)
 - row subtotals formatting layer [325](#)
 - smart [98](#)
 - tasks. See subtotals - tasks.
 - user-defined [104](#)
 - weighted [108](#)
- subtotals - tasks
 - creating [104, 106](#)

- disabling [111](#)
- displaying on a report [102, 291](#)
- formatting [319](#)
- sum
 - moving [116](#)
 - running [116](#)
- support. See technical support.
- system hierarchy drill path [227](#)
- system prompt [258](#)
 - user login [258](#)

T

- table
 - importing [705](#)
 - selecting in a Query Builder report [705](#)
 - VLDB properties
 - Cartesian join [866](#)
 - option [880](#)
 - post statement [873](#)
 - space [880](#)
- target of a link [564](#)
- technical support [xxx](#)
- template [358](#)
 - embedded [354](#)
 - empty object [358](#)
 - object [357](#)
 - shortcut to [353](#)
 - validating changes to [356](#)
- Template Dependency Validator [356](#)
- template details [520](#)
 - base report [521](#)
 - example [522](#)
 - base template name [523](#)
 - example [524](#)
 - configuring [520, 528](#)
 - report details preferences
 - interaction [526](#)

- view report [521](#)
 - example [522](#)
 - temporary table
 - dropping [878](#)
 - intermediate table type [879](#)
 - text file, exporting a report to [363](#)
 - third-party web service, connecting to [638](#)
 - threshold [defined on 320](#)
 - formatting [320, 325](#)
 - graph report [446](#)
 - time series analysis [92, 798](#)
 - title, formatting in a graph [456](#)
 - tooltip for a graph [481](#)
 - totals
 - disabling [111](#)
 - formatting [319](#)
 - metric [102](#)
 - overview [289](#)
 - report [289](#)
 - smart [98](#)
 - user-defined subtotals [104](#)
 - training metric [809](#)
 - dependent metric [815](#)
 - exponential regression [811](#)
 - independent metric [815](#)
 - linear regression [811](#)
 - output model [819](#)
 - segmentation metric [815](#)
 - tree level metric [815](#)
 - Training Metric Wizard [785, 810](#)
 - training the model [809](#)
 - Transaction Services. See MicroStrategy Transaction Services.
 - transformation [defined on 90](#)
 - transformation metric [90](#)
 - creating [94](#)
 - Advanced Metric Assistant [99](#)
 - multiple transformations [95](#)
 - year-to-date [93](#)
 - transparency on a graph object [370](#)
- ## U
- union multiple insert [878](#)
 - univariate regression model [788](#)
 - updating data with MicroStrategy Transaction Services [675](#)
 - user audience [xxi](#)
 - user input with a prompted filter [174](#)
 - user login system prompt [258](#)
 - user-defined empty object template [359](#)
 - user-defined function. See custom function.
 - user-defined subtotals [104](#)
 - creating [106](#)
 - example [105, 108](#)
 - using MicroStrategy Web [478](#)
- ## V
- validating changes to a template [356](#)
 - value prompt [249](#)
 - link and [568](#)
 - variable for a graph title [455](#)
 - variance [92](#)
 - view definition [defined on 286](#)
 - evaluation order [344](#)
 - Intelligent Cube [288](#)
 - object [287](#)
 - virtual attribute (consolidation) [207](#)
 - VLDB property [859](#)
 - accessing
 - advanced report VLDB properties [866](#)
 - database instance VLDB properties [882](#)
 - metric VLDB properties [861](#)
 - report VLDB properties [866](#)

Attribute Form Selection Option for Intermediate Passes [875](#)
Attribute Selection Option for Intermediate Passes [875](#)
Cartesian Join Warning [866](#)
database instance level [882](#)
default inherited value [860](#)
Drop Temp Table Method [878](#)
Fallback Table Type [879](#)
Full Outer Join Support [868](#)
Group by ID Attribute [877](#)
Group by Non-ID Attribute [878](#)
hierarchy of [860](#)
Integer Constant in Metric [861](#)
Intermediate Table Type [879](#)
Maximum SQL Size [882](#)
metric level [861](#)
Null Check [862](#)
Null Checking for Analytical Engine [863](#)
order of precedence [860](#)
Preserve All Final Pass Result Elements [868](#)
Preserve All Lookup Table Element [869](#)
report level [865](#)
Report Post Statement [872](#)
Report Pre Statement [872](#)
SQL Date Format [883](#)
SQL Hint [878](#)
Sub Query Type [873](#)
Subtotal Dimensionality Aware [863](#)
Table Option [880](#)
Table Post Statement [873](#)
Table Space [880](#)
Union Multiple Insert [878](#)
Where Clause Driving Table [874](#)
Zero Check [863](#)

W

web service
as a data source [638](#)
connecting MicroStrategy to [639](#)
creating a report [659](#)
security in an XQuery statement [644](#)
XQuery for data retrieval from [641](#)
Web Services Security (WS-Security) for an XQuery statement [645](#)
weighted subtotals [108](#)
Where Clause Driving Table VLDB property [874](#)
widget [507](#)
write-back. See MicroStrategy Transaction Services.
WS-Security. See Web Services Security.

X

XQuery [638](#)
creating a Transaction Services report [676](#)
retrieving data from a web service [641](#)
security [644](#)

Z

zero checking [863](#)
zone formatting [318](#)
subtotals [319](#)