MicroStrategy®

Analytics and Mobility

In-memory Analytics Guide

MicroStrategy ONE

MicroStrategy ONE

April 2024

Copyright © 2024 by MicroStrategy Incorporated. All rights reserved.

Trademark Information

The following are either trademarks or registered trademarks of MicroStrategy Incorporated or its affiliates in the United States and certain other countries:

Dossier, Enterprise Semantic Graph, Expert.Now, Hyper.Now, HyperIntelligence, HyperMobile, HyperVision, HyperWeb, Intelligent Enterprise, MicroStrategy, MicroStrategy 2019, MicroStrategy 2020, MicroStrategy 2021, MicroStrategy AI, MicroStrategy Analyst Pass, MicroStrategy Architect, MicroStrategy Architect Pass, MicroStrategy Auto, MicroStrategy Cloud, MicroStrategy Cloud Intelligence, MicroStrategy Command Manager, MicroStrategy Communicator, MicroStrategy Consulting, MicroStrategy Desktop, MicroStrategy Developer, MicroStrategy Distribution Services, MicroStrategy Education, MicroStrategy Embedded Intelligence, MicroStrategy Enterprise Manager, MicroStrategy Federated Analytics, MicroStrategy Geospatial Services, MicroStrategy Identity, MicroStrategy Identity Manager, MicroStrategy Identity Server, MicroStrategy Insights, MicroStrategy Integrity Manager, MicroStrategy Intelligence Server, MicroStrategy Library, MicroStrategy Mobile, MicroStrategy Narrowcast Server, MicroStrategy ONE, MicroStrategy Object Manager, MicroStrategy Office, MicroStrategy OLAP Services, MicroStrategy Parallel Relational In-Memory Engine (MicroStrategy PRIME), MicroStrategy R Integration, MicroStrategy Report Services, MicroStrategy SDK, MicroStrategy System Manager, MicroStrategy Transaction Services, MicroStrategy Usher, MicroStrategy Web, MicroStrategy Workstation, MicroStrategy World, Usher, and Zero-Click Intelligence.

The following design marks are either trademarks or registered trademarks of MicroStrategy Incorporated or its affiliates in the United States and certain other countries:

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.

CONTENTS

OLAP Services features ... 6

The benefits of using OLAP Services ... 10

Privileges required for Developer and Web users ... 11

Report types: Standard and OLAP reports ... 13

Standard OLAP analysis features ... 14

2. Sharing Sets of Data Among Reports: Intelligent Cubes ... 18

Sharing Intelligent Cubes ... 19

Updating Intelligent Cubes without re-processing: Incremental Refresh ... 48

Improving the performance of large Intelligent Cubes: Partitioning ... 58

Reporting and analyzing data with OLAP Services features ... 59

Reporting and analyzing data with Intelligent Cubes ... 59

Reporting on Intelligent Cubes with dynamic sourcing ... 80

Overview: Large, in-memory datasets in MicroStrategy ... 82

Creating an application that uses a partitioned dataset ... 83

Editing and updating your dataset ... 89

5. Derived Elements ... 93

Types of derived elements ... 97

Creating derived elements ... 106

Defining derived element functionality and formatting ... 151

Interaction with other reporting features ... 164

6. Derived Metrics ... 177

Creating a derived metric ... 179

Editing derived metrics ... 198
Formatting derived metrics ... 198
Deleting derived metrics ... 199
View filter effects on derived metrics ... 199
Derived element effects on derived metrics ... 200

7. View Filters ... 201

Comparing view filters to report filters and report limits ... 204
Creating a view filter ... 211
Deleting a view filter ... 238
View filter effects on reporting features ... 239

8. Dynamic Sourcing ... 254

Scenarios that benefit from dynamic sourcing ... 255
Best practices for supporting dynamic sourcing ... 257
Configuring dynamic sourcing ... 266
Using Cube Advisor to support dynamic sourcing ... 287
Tracking the use of dynamic sourcing ... 300

9. Dynamic Aggregation ... 322

Using dynamic aggregation ... 325
Functions used in dynamic aggregation ... 327
View filter effect on dynamic aggregation ... 343

A. Efficient Functions for Partitioned Datasets ... 344

Functions for partitioned datasets ... 345

B. Best Practice for MicroStrategy PRIME ... 352

Prerequisites ... 354

Parallel queries 360
Partitioning cubes 365
Sizing 370
Cube incremental refresh 372
Data source 372
Document / dashboard best practice 373
Concurrency 378
Web or mobile access 378

OLAP Services features

As an extension of MicroStrategy Intelligence Server, OLAP Services uses Intelligent Cube Technology—an in-memory version of report data that can be manipulated by the MicroStrategy Analytical Engine. MicroStrategy Developer, Web, and Office users can slice and dice data in reports within Intelligent Cubes without having to re-execute SQL against the data warehouse. *Chapter 2, Sharing Sets of Data Among Reports: Intelligent Cubes* describes the role of OLAP Services in the overall MicroStrategy architecture and how you can take advantage of Intelligent Cubes to achieve the best efficiency and performance in your application.

As a MicroStrategy user, you may already take advantage of the many standard OLAP features that MicroStrategy provides out of the box, such as page-by, pivoting, sorting, and subtotals.

With the OLAP Services feature, you can perform additional OLAP analysis, using the following features:

- *Displaying data on the fly: dynamic aggregation, page 7*
- *Creating metrics on the fly: derived metrics, page 8*
- *Defining attribute elements on the fly: derived elements, page 8*
- *Filtering data on the fly: view filters and metric filters, page 9*
- *Importing data as an Intelligent Cube, page 9*

Providing OLAP analysis with Intelligent Cubes

The main feature supporting MicroStrategy OLAP Services is MicroStrategy Intelligent Cube Technology™, which allows you to create multi-dimensional cubes (sets of data) that are stored within MicroStrategy Intelligence Server.

Rather than returning data from the data warehouse for a single report, you can return sets of data, called Intelligent Cubes, from your data warehouse, and save them directly to Intelligence Server's memory. Intelligent Cubes

Copyright © 2024 All Rights Reserved

can be shared as a single in-memory copy, to be used by many different reports created by multiple users.

For detailed information on Intelligent Cubes, refer to *Chapter 2, Sharing Sets of Data Among Reports: Intelligent Cubes*.

Displaying data on the fly: dynamic aggregation

When you create a non-OLAP Services report in MicroStrategy, data is retrieved from the data warehouse and aggregated to the logical level determined by the objects on the report.

Dynamic aggregation allows metric values to be aggregated at different levels, depending on the attributes included on the report, without having to re-execute it against the data warehouse. Dynamic aggregation occurs when the attributes included on the report layout changes—for example, when you move an attribute or attribute form off of the report layout to the Report Objects pane, or when you move an attribute or attribute form from the Report Objects pane back onto the report layout. As the attributes included on the report layout change, metric values are dynamically aggregated to the new level of the report.

For example, a report with revenue data and attributes for year and region displays revenue data for each region during each year. You can move the attribute for year from the report layout to the Report Objects pane to display a region's revenue data for all years rather than display each yearly revenue total separately.

Metrics can also be moved between the report layout and the Report Objects pane, but this does not affect the level of aggregation for the report.

By default, the Analytical Engine selects the best aggregation function to use for each metric. However, you can also specify the function for each metric. You can use any of the standard predefined subtotal functions or define your own functions using user-defined subtotals.

Copyright © 2024 All Rights Reserved

For more detailed information on dynamic aggregation, refer to *Chapter 9, Dynamic Aggregation*.

Creating metrics on the fly: derived metrics

Derived metrics perform calculations on the fly with the data available in a report. They are an easy way to present data already available on the report in different ways, providing further analysis of data. You can use derived metrics to quickly perform on-the-fly analyses such as margins, contributions, and differences between metrics included on the report.

These metrics are created based on existing metrics in the report. Since derived metrics are evaluated in-memory, their computation does not require any SQL to execute in the database.

Since derived metrics are created within a report, they can only be used for the report in which they are created. They cannot be saved as individual objects in the project, and therefore cannot be applied to other reports in the project.

Click here for more detailed information on derived metrics.

Defining attribute elements on the fly: derived elements

A derived element is a grouping of attribute elements on a report. These groups provide a new view of report data for analysis and formatting purposes. For example, you can group data for the months of December, January, and February into a single element that combines and displays the data for the entire winter season.

Rather than having to define consolidations or custom groups, you can use derived elements to create these groups on the fly while viewing a report. Derived elements are evaluated on the report dataset without regenerating or re-executing SQL.

Derived elements are defined by using a list, filter, or calculation to combine attribute element data.

Copyright © 2024 All Rights Reserved

For detailed information on derived elements, see *Chapter 5, Derived Elements*.

Filtering data on the fly: view filters and metric filters

A view filter dynamically restricts the data being displayed on the report without re-executing the report against the warehouse. This is different from a report filter, which restricts how much data is retrieved from the data warehouse. View filters improve the response time of reports, and decrease database load.

It is important to note that you can use a report filter and view filter on the same report. The report filter returns a set of data for the report, which the view filter then restricts further. You should therefore avoid defining contradictory filtering criteria in both, otherwise you many encounter situations where no data is displayed.

For detailed information on view filters, see *Chapter 7, View Filters*.

Importing data as an Intelligent Cube

You can use the Import Data feature in MicroStrategy Web to import data from sources such as Microsoft Excel spreadsheets, comma-separated text files, and external databases into your project. This data is stored as Intelligent Cubes.

Some uses for this data include:

- 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 and documents without having to "model" the data source as part of the entire project
- Modifying the data in your data source, then republishing the Intelligent Cube to quickly update the data in your reports and documents

For detailed information on the Import Data feature, see the MicroStrategy Web Help.

Copyright © 2024 All Rights Reserved

The benefits of using OLAP Services

OLAP Services combines the benefits of both MOLAP and ROLAP analyses.

Generally speaking, MOLAP tools place data in multi-dimensional cubes and perform consolidations in advance, allowing you to run queries fast. ROLAP tools allow you to use complex SQL queries against relational tables to obtain multi-dimensional views of data on the fly.

Each type of analysis has its own advantages and disadvantages. However, OLAP Services provides MOLAP and ROLAP analysis on the same report, which offers many distinct benefits, summarized below:

- **Analyze reports at the "speed of thought", and manipulate them in real time**

  Using OLAP Services, you can get fast response times for reports that use data directly from in-memory Intelligent Cubes, instead of from the data warehouse. You can create and analyze new reports in real time through interactive OLAP Services manipulations.

- **Share Intelligent Cube data securely**

  MicroStrategy's centralized metadata and Intelligence Server architecture allows Intelligent Cube data to be shared in a secure fashion.

- **Schedule Intelligent Cube execution and maintenance**

  To reduce stress on the Intelligence Server, you can schedule when Intelligent Cubes are executed. This allows you to take advantage of Intelligence Server down time to execute Intelligent Cubes without affecting performance for your user community.

  You can also schedule when Intelligent Cubes are re-executed to synchronize their data with changes to the data in your data warehouse.

- **Drill from summary data to transaction-level details**

Copyright © 2024 All Rights Reserved

You can drill from predefined reports to conduct advanced analysis and take full advantage of the Intelligent Cube feature. Drilling is allowed within an Intelligent Cube for quick-response MOLAP analysis. Drilling can also be enabled outside of an Intelligent Cube for full ROLAP analysis.

- **Use MicroStrategy Developer, Office, or Web**

  Using OLAP Services, you can perform the same multidimensional analysis whether you use MicroStrategy Developer, Office, or Web.

- **Apply security restrictions on users and objects**

  Reporting with OLAP Services and Intelligent Cubes adhere to the same standards of data access security as the rest of your MicroStrategy project.

- **Increase user self-service and productivity**

  Since accessing Intelligent Cubes for OLAP analysis does not require runtime processing on the data warehouse and can use schedules to reduce IT management, users have increased flexibility to create and modify their own reports to suit their unique work environment.

Privileges required for Developer and Web users

You can assign related privileges to different user groups on Developer or Web to perform specific tasks. The following list summarizes which privileges can be assigned for which user groups:

- **Common privileges**: The predefined MicroStrategy Web Reporter and Analyst user groups are assigned the set of common privileges by default. Within the set of common privileges, the following privilege is specific to OLAP Services:

  - **Drill within Intelligent Cube**: This privilege allows you to drill within an Intelligent Cube, which means that the drill can be resolved through

Copyright © 2024 All Rights Reserved

OLAP Services and therefore does not need to generate and execute SQL against the warehouse.

- **Use Dynamic Sourcing**: This privilege lets you to use Dynamic Sourcing, which allows non-OLAP Services reports use Intelligent Cubes that satisfy their data requirements.
- **Use OLAP Services**: This privilege lets you create and execute reports and documents that use OLAP Services.

- **Web Analyst**: The predefined MicroStrategy Web Analyst group is assigned the set of Web Analyst privileges by default. This group also inherits all of the privileges assigned to the Web Reporter group. Within the set of Web Analyst and Reporter privileges, the following privileges are specific to OLAP Services users:
  - **Web add/remove units to/from grid in document in View mode**: This privilege allows you to add to or remove report objects from an existing grid report in a Report Services document.
  - **Web create derived metrics and derived attributes**: This privilege allows you to create new calculations based on other metrics already on a base report.
  - **Web number formatting**: This privilege allows you to change number formats for all metrics on grid reports.
  - **Web use Report Objects window**: This privilege allows you to use the Report Objects pane. With the Report Objects pane, you can use dynamic aggregation to change the attributes available on the report layout.
  - **Web use View Filter Editor**: This privilege allows you to add or modify a view filter for a report.
- **Analyst**: The predefined MicroStrategy Analyst group is assigned this set of privileges by default. Within the set of Analyst privileges, the following privileges are specific to OLAP Services:

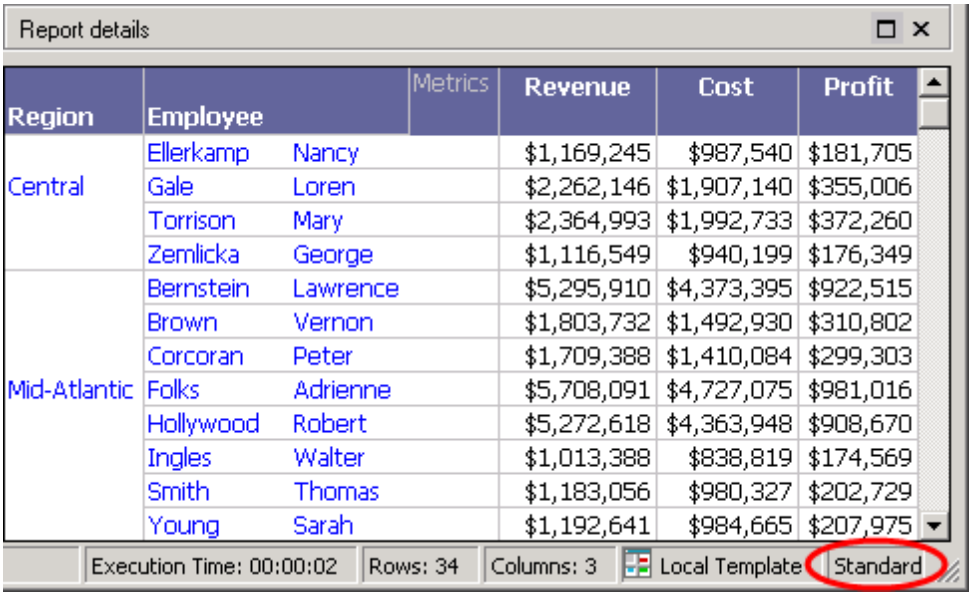
Copyright © 2024 All Rights Reserved

- **Create derived metrics**: This privilege allows you to create new calculations based on other metrics already on a base report.
- **Use Report Objects window**: This privilege allows you to use the Report Objects pane. With the Report Objects pane, you can use dynamic aggregation to change the attributes available on the report layout.
- **Use view filter editor**: This privilege allows you to add or modify a view filter for a report.

For information on all user privileges, see the System Administration Guide.

Report types: Standard and OLAP reports

During the report design process, if none of the OLAP Services features are applied on the report, then the report remains a standard report. This is indicated by the word **Standard** in the bottom right corner of the Report Editor, as shown below.

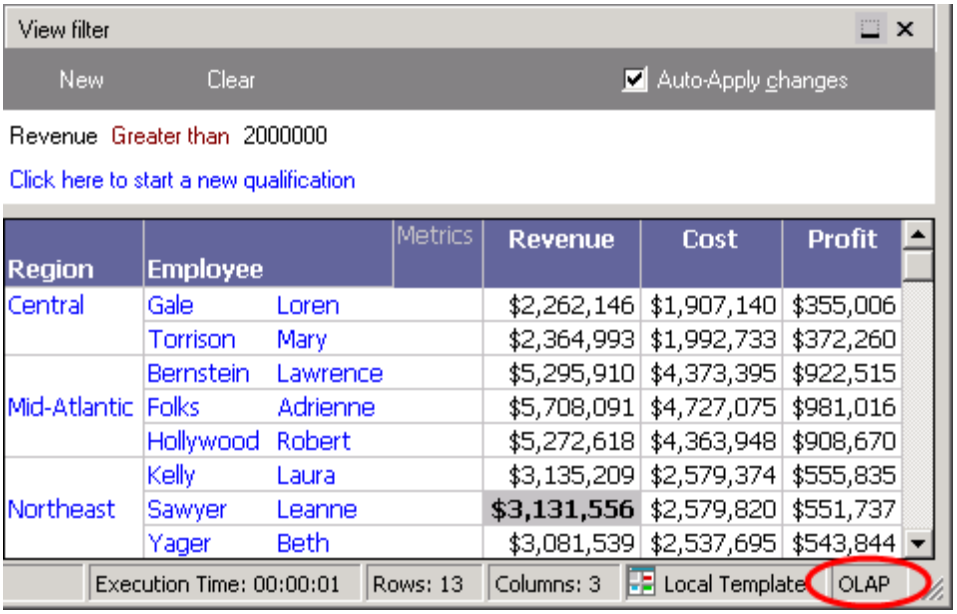


In Developer, the moment you use an OLAP Services feature, whether it is dynamic aggregation, a derived metric, a derived element, or a view filter, the standard report becomes an OLAP report, as shown in the following

Copyright © 2024 All Rights Reserved

image. Notice that **Standard** is replaced by **OLAP** after a view filter is applied.

The report type change from Standard to OLAP is displayed only in Developer. The same indication is not shown on reports in MicroStrategy Web.



When a saved OLAP Services report is opened again, it remains an OLAP Services report—while a Standard report can be turned into an OLAP Services report, but an OLAP Services report cannot be turned into a Standard report.

Standard OLAP analysis features

Whether you use MicroStrategy Developer or Web, once you execute a report, you can perform various OLAP manipulations on the report using features such as banding, sorting, pivoting, page-by, and so on. These features do not cause the report to be re-executed against the warehouse, and therefore have a much faster response time.

Copyright © 2024 All Rights Reserved

These standard OLAP features are different than the OLAP Services features covered in this guide. Each standard OLAP feature is described briefly in the following sections below. These features listed below are covered in detail in the Basic Reporting Help and Advanced Reporting Help:

- Aliasing

  When displaying a report, you can use the aliasing feature to rename any object on the report grid, such as attribute names, consolidation names, custom group names, and metric names. You can perform this task from both MicroStrategy Developer and Web.

- Banding

  Banding allows you to color groups of rows or columns so that they form bands of data that are easy to locate and analyze. Banding can also make it easier to make sense of a very large report, because the large amounts of data are broken up into visual groups. If you need to keep track of values that mean different things in different columns (for example, dollars in one column and inventory quantities in another column), banding can help you avoid reading the wrong number.

  Banding is a method of organizing or grouping data values in a grid report according to certain criteria. You can band rows or columns in several ways. You can band based on the number of rows or columns (for example, alternating color every 5 rows). You can also band data based on the row and column headers (for example, sorting the Units Sold column in order, then applying alternating colors to sets of values). For information on applying banding to a report, see the Basic Reporting Help.

- Outline mode

  Outline mode allows you to create an indented grouping of related attribute elements. You can collapse and expand sections of related data. This function is particularly useful in instances where the information displayed would otherwise involve repetitive entries. For example, in the case of a grid report showing sales by year, each year is broken down by

Copyright © 2024 All Rights Reserved

month. With outline mode enabled, the data is organized into groups, with months of each year nested below the years.

- Page-by

  Page-by is a way to segment data in a grid report by placing available attributes, consolidations, or metrics on a third axis called the Page axis. Based on the varying objects on the axis, you can view the report data in separate pages. This feature is most useful when you have an extremely long report with many objects, and you need to scroll to see all the data. You can page by many objects, such as attributes, metrics, hierarchies, consolidations, custom groups, and so on.

- Pivoting

  Pivoting enables you to rearrange the columns and rows in a report to view data from different perspectives. With data pivoting, you can do the following:

  - Move an object from the row header to the column header
  - Move an object from the column header to the row header
  - Change the order of objects in the row header
  - Change the order of objects in the column header

- Sorting

  Sorting allows you to specify an ascending or descending order for a particular row or column to present the report data. You can select what objects you want to sort, the sorting criteria, and the sorting order. MicroStrategy Developer and Web offer quick sort, advanced sort, and hierarchical sort.

- Subtotals

  Using the Subtotal feature, you can add, remove, and edit the subtotals at different levels for metrics on the report. The subtotal functions available

Copyright © 2024 All Rights Reserved

include sum, count, min, max, average, mean, median, and so on. You might choose to display all subtotals, a grand total only, or subtotals across levels where you select the object to be subtotaled. Additionally, Developer allows you to construct custom subtotals that, for example, allow you to enable subtotals for selected metrics only.

- Thresholds

  A threshold highlights data that meets conditions defined by you. Highlighting data can include using different cell formats, symbols, images, or replacement text.

Copyright © 2024 All Rights Reserved

SHARING SETS OF DATA AMONG REPORTS: INTELLIGENT CUBES

Copyright © 2024 All Rights Reserved

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. MicroStrategy Intelligent Cube Technology™ allows you to create these Intelligent Cubes, which operate within MicroStrategy Intelligence Server.

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.

Sharing Intelligent Cubes

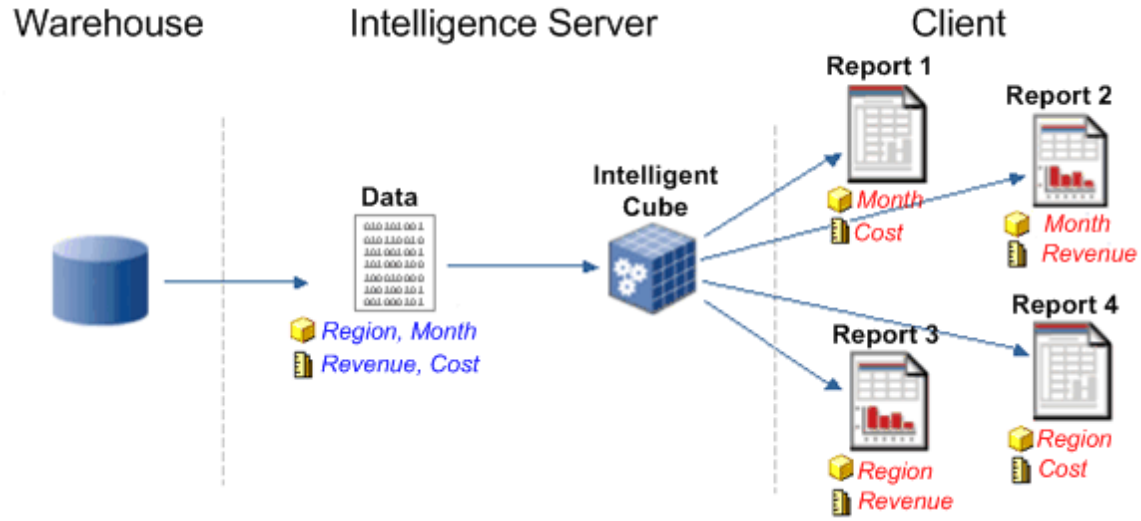
MicroStrategy 9.0 and later enhances the scalability and manageability of OLAP Services with the addition of sharable Intelligent Cubes. Intelligent Cubes allow multiple reports to retrieve data from a single shared in-memory set of data.

You create Intelligent Cubes and publish them as a shared data source for the users to build reports from. Intelligent Cubes provide the fast response time and analytic calculations that are often associated with Multidimensional Online Analytic Processing (MOLAP) cubes, while also benefiting from the ability to use Relational Online Analytic Processing (ROLAP) by drilling into the full set of data outside of the Intelligent Cube. In addition, Intelligent Cubes are fully scalable, limiting excessive data consumption and redundant data by allowing you to build only the sets of data you require.

With Intelligent Cubes you can return a specific set of data from your data warehouse. Users can then create reports that display and analyze a subset of the set of data defined in an Intelligent Cube. This process is different than the common approach of creating a report that directly accesses the data warehouse.

Copyright © 2024 All Rights Reserved

Intelligent Cubes act as a layer between your data warehouse and MicroStrategy reports that analyze and display data, as illustrated below.



The abstraction that Intelligent Cubes provide between your data warehouse and reports can improve the performance of your business intelligence application in the following ways:

- Reports that connect to an Intelligent Cube can perform reporting and analysis manipulations within the Intelligent Cube without hitting the data warehouse. These manipulations are executed much faster than running a new query against a data warehouse.
- The data that reports can access is restricted to the data within the Intelligent Cube. Users can still perform a few ROLAP manipulations such as drilling that can access the data warehouse. However, these types of manipulations that cause re-execution against the data warehouse are not as accessible as they are when using standard reports.

  For example, when using a standard report, users can access any attribute defined for the project and include it on the report. A manipulation of this type requires re-execution against the data warehouse. Conversely, a user working in a report that connects to an Intelligent Cube can only add

Copyright © 2024 All Rights Reserved

attributes to the report grid if the attributes are included in the Intelligent Cube.

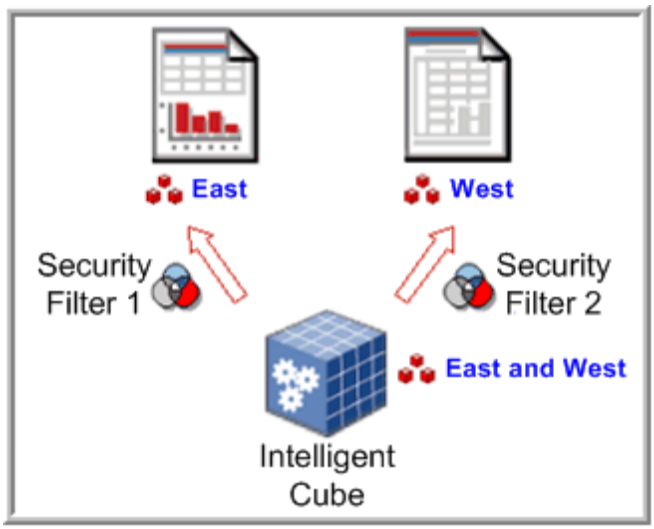
Therefore, Intelligent Cubes help to limit the amount of processing done in the data warehouse and improve performance.

- Security filters are applied separately for each user at the level of the report connecting to the Intelligent Cube, rather than having to create multiple Intelligent Cubes for each security filter. For more information, see *Maintaining data access security, page 21*.

Maintaining data access security

Reports that connect to Intelligent Cubes adhere to many of the same standards of data access security as the rest of your MicroStrategy project. You can control users' access to data with security filters. For more information on security filters, see the System Administration Help.

User and group security filters are applied automatically on reports that connect to an Intelligent Cube, as shown below:



This approach allows a single Intelligent Cube to be used by multiple security filters, rather than having to create separate Intelligent Cubes for each security filter. By using a single Intelligent Cube to support all the

Copyright © 2024 All Rights Reserved

security filters for a project, data access security is implemented automatically with minimal burden on Intelligence Server memory.

However, there are some differences in security filter resolution for reports that connect to Intelligent Cubes as compared to reports that directly access the data warehouse.

Security filter resolution for reports connected to Intelligent Cubes

Security filter resolution for reports that connect to Intelligent Cubes differs as compared to reports that directly access the data warehouse. These differences relate to what attributes are on a user's security filter, and how these relate to the attributes and fact entry levels of metrics available in the Intelligent Cube used for the report.

If all attributes in a user's security filter are in the Intelligent Cube that is used for the report, then security filters can be resolved using the standard process. However, if some of the attributes in a user's security filter are not in the Intelligent Cube used for the report, security filter resolution can differ from the standard process.

While this scenario is uncommon, it can cause users of reports that access Intelligent Cubes to experience one of the following results:

- No data is returned for the report.

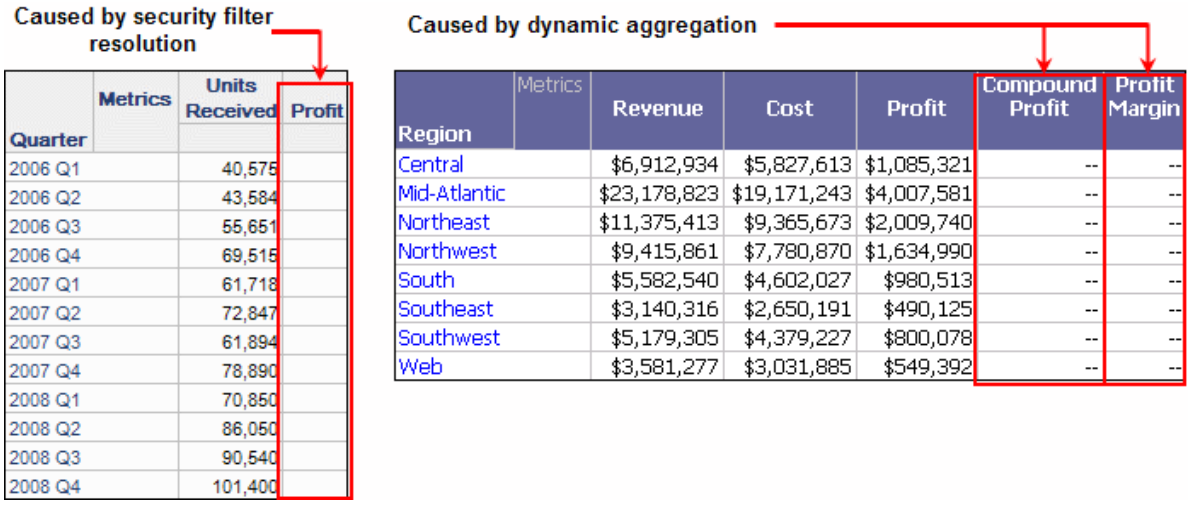
  This could also be caused by the user creating a view filter that is too restrictive, or by the user's normal security filter resolution.

- A metric or metrics are displayed with no data returned for the metric.

  Another possible cause of metrics not properly returning data on reports connected to Intelligent Cubes is the use of dynamic aggregation (see *Chapter 9, Dynamic Aggregation*). However, when this is caused by dynamic aggregation, null values are displayed for the metric rather than not displaying any information at all. The image below shows the

Copyright © 2024 All Rights Reserved

difference between security filter resolution and dynamic aggregation as the cause for metrics not displaying any data.



Quarter	Metrics	Units Received	Profit
2006 Q1		40,575	
2006 Q2		43,584	
2006 Q3		55,651	
2006 Q4		69,515	
2007 Q1		61,718	
2007 Q2		72,847	
2007 Q3		61,894	
2007 Q4		78,890	
2008 Q1		70,850	
2008 Q2		86,050	
2008 Q3		90,540	
2008 Q4		101,400	

Region	Metrics	Revenue	Cost	Profit	Compound Profit	Profit Margin
Central		$6,912,934	$5,827,613	$1,085,321	--	--
Mid-Atlantic		$23,178,823	$19,171,243	$4,007,581	--	--
Northeast		$11,375,413	$9,365,673	$2,009,740	--	--
Northwest		$9,415,861	$7,780,870	$1,634,990	--	--
South		$5,582,540	$4,602,027	$980,513	--	--
Southeast		$3,140,316	$2,650,191	$490,125	--	--
Southwest		$5,179,305	$4,379,227	$800,078	--	--
Web		$3,581,277	$3,031,885	$549,392	--	--

By default, null values are represented by dashes (--) on reports. For information on changing the display of null values, see *Changing the display of null values, page 340*.

If a user is experiencing one of the two scenarios listed above due to security filter resolution, the following resolutions can be considered:

Resolution	Pros	Cons
The user continues to use the report that accesses the Intelligent Cube.	Data access security is maintained. No additional resources are needed to modify the Intelligent Cube or to create a new report.	Some data that may be available to the user by directly querying a data source may not be available in the report that accessed an Intelligent

Copyright © 2024 All Rights Reserved

Resolution	Pros	Cons
		Cube.
The user creates or views a report with the same definition that directly queries a data source rather than accessing an Intelligent Cube.	The user is able to verify the full results that can be returned for such a report.	A new report that directly queries a data source must be created. The new report cannot take advantage of the improved query performance of accessing an Intelligent Cube.
Add the attributes used in a user's security filter to the Intelligent Cube and publish the updated Intelligent Cube.	The security filter resolution for the user can use the standard process and return the same data as if the report were directly querying a data source. This is also helpful if multiple users could benefit from the same change to the Intelligent Cube definition.	The Intelligent Cube must be published again to reflect the new definition. Publishing the Intelligent Cube can require substantial system resources. Including additional attributes requires more memory for

Copyright © 2024 All Rights Reserved

Resolution	Pros	Cons
		the Intelligent Cube to be stored on Intelligence Server.

Security filter resolution when attributes in a user's security filter are not in the Intelligent Cube used for the report

When attributes in a user's security filter are not in the Intelligent Cube used for the report, the outcome depends on how the attributes are related to those in the Intelligent cube, as described below:

- **Attributes in the security filter are related to attributes in the Intelligent Cube**: No data is returned, to maintain data access security.

  For example, an Intelligent Cube includes the attributes Year and Region, and the metric Revenue. A user creates a report that connects to this Intelligent Cube, and includes Year and Revenue on the report. The user's security filter is defined on the attribute Quarter to return data only for the first quarter of 2008.

  By including the Year attribute on the report, this report would return information for all quarters in each year. However, the user is allowed to only see data for the first quarter of 2008. To maintain this data access security, no data is returned for the report.

- **Attributes in the security filter are not related to attributes in the Intelligent Cube**: The data returned depends on whether metrics in the Intelligent Cube report fact data based on attributes related to those in the user's security filter:

  - **A metric in the Intelligent Cube reports fact data based on an attribute related to one in the security filter**: The user's security filter

Copyright © 2024 All Rights Reserved

prevents any data from being returned.

For example, an Intelligent Cube includes the attributes Year and Region, and the metric Revenue, which is based on the fact Revenue. This fact, in turn, is reported based on the attributes Item, Day, and Call Center. A user creates a report that connects to this Intelligent Cube, and includes Year and Revenue on the report. However, this user's security filter is defined on the attribute Category to return data only from Books.

Since Revenue is based on Item, Day and Call Center only, it cannot be reported based on the Category attribute. In such a case, no data will be reported for the Revenue metric.

- **A metric in the Intelligent Cube reports fact data based on an attribute unrelated to those in the security filter**: The security filter does not apply any restriction, and displays the data for metrics based on the fact.

- The table below describes the security filter resolution for scenarios listed above:

	A fact in the Intelligent Cube is reported based on an attribute in the security filter	A fact in the Intelligent Cube is not reported based on any of the attributes in the security filter
Attributes in the security filter are related to attributes in the Intelligent Cube	No data is returned, to maintain data access security.	No data is returned, to maintain data access security.
Attributes in the security filter are not related to attributes in the Intelligent Cube	Data can be returned using the standard security filter resolution. However, to maintain data access	Data can be returned using the standard security filter resolution. In this scenario the

Copyright © 2024 All Rights Reserved

	A fact in the Intelligent Cube is reported based on an attribute in the security filter	A fact in the Intelligent Cube is not reported based on any of the attributes in the security filter
	security, no data is displayed for any metrics where fact data is reported based on attributes related to those in the security filter.	security filter does not need to restrict any data, and the metric data can also be displayed.

Data access security with connection mapping

In MicroStrategy, you can also use connection mapping to control the data that users have access to when they run reports. You can also apply this security when users create reports that connect to an Intelligent Cube. For information on maintaining data access security with connection mapping to Intelligent Cubes, see the System Administration Help.

Setting permissions for individual Intelligent Cubes

You can use an Intelligent Cube's Access Control List (ACL) to set specific access permissions for users. For example, you can restrict some users to only create reports based on an Intelligent Cube, but not re-execute the Intelligent Cube.

You can use the ACL Editor in -MicroStrategy Web to assign the following permission groups to users, for each Intelligent Cube:

Copyright © 2024 All Rights Reserved

Group	Description	Permissions granted
Consume	Grants permission to create and execute reports based on this Intelligent Cube.	Browse Read Use
Add	Grants permission to create and execute reports based on this Intelligent Cube, and republish/re-execute the Intelligent Cube to update the data.	Browse Read Use Execute
Collaborate	Grants permission to create and execute reports based on this Intelligent Cube, republish/re-execute the Intelligent Cube to update the data, and modify the Intelligent Cube.	Browse Read Write Delete Use Execute

For information on ACLs and access permissions, see the System Administration Help.

Creating Intelligent Cubes

Creating Intelligent Cubes is as easy as creating reports.

You need the Use Intelligent Cube Editor privilege to create Intelligent Cubes. This privilege is part of OLAP Services.

After you have created your Intelligent Cube, you can publish its set of data to be shared by multiple reports. For more information on publishing an Intelligent Cube, see *Publishing Intelligent Cubes, page 43*.

Copyright © 2024 All Rights Reserved

Prerequisites to creating Intelligent Cubes

Make sure you review the following prerequisites before you create an Intelligent Cube in MicroStrategy. They can save you time and make your Intelligent Cubes more effective for your reporting environment:

While creating Intelligent Cubes, bear in mind that Intelligent Cubes can deplete Intelligence Server's system resources. Create Intelligent Cubes for logical subsets of your data, rather than using them as a reflection of your entire data warehouse. For information on managing the size of Intelligent Cubes, see *Managing Intelligent Cubes* in the System Administration Help.

- Define your business query: Before you can determine what objects to place on an Intelligent Cube, you need to know what data you want to make available for reports to access directly. To define your Intelligent Cube, make sure you consider the following questions:
  - What subset of business queries does the Intelligent Cube need to provide data for? Intelligent Cubes allow you to create sets of data that can support multiple reports that answer variations to similar business queries.
  - Do you have reports that currently access your data warehouse that could benefit from accessing an Intelligent Cube instead? To support this scenario, you can use dynamic sourcing to connect these reports to Intelligent Cubes that you create. You can also use MicroStrategy Cube Advisor to create Intelligent Cubes that these reports can access. For information on using Cube Advisor to support dynamic sourcing, see *Using Cube Advisor to support dynamic sourcing, page 287*.
- Look for existing Intelligent Cubes: Before you create an Intelligent Cube, search through MicroStrategy to see whether a similar Intelligent Cube already exists that can serve the same purpose as the Intelligent Cube that you intend to create. This can not only save you time, it can help you avoid unnecessary duplication in your MicroStrategy project. You can search a project for Intelligent Cubes, or you can view Intelligent Cubes

Copyright © 2024 All Rights Reserved

created for your projects in the Intelligent Cube Monitor. For information on using the Intelligent Cube Monitor, see the System Administration Help .

- Review the feature support for Intelligent Cubes: The MicroStrategy features that are available for Intelligent Cubes differ from those available for reports.

Supporting various features with Intelligent Cubes

Intelligent Cubes act as a set of data for other reports to connect to and use OLAP Services to report, analyze, and display data. Since Intelligent Cubes function simply as a set of data, Intelligent Cubes do not need to contain objects that modify the display of data. The display of data is handled by the reports that access the Intelligent Cube.

For these reasons, the following objects and features cannot be included in Intelligent Cubes in the same ways that they can be included for reports:

- **Consolidations and custom groups**: Consolidations and custom groups cannot be included in Intelligent Cubes or in reports that access Intelligent Cubes. However, you can include derived elements in reports that access Intelligent Cubes. Derived elements combine the analysis features of consolidations and custom groups, while executing against the Intelligent Cube rather than the data warehouse. For information on derived elements, see *Chapter 5, Derived Elements*.
- **OLAP Services features**: View filters, derived metrics, and derived elements cannot be included in Intelligent Cubes, but these features are the main analysis tools for reports that access Intelligent Cubes. For an overview of OLAP Services features and how these can be created in reports, see About MicroStrategy OLAP Services.
- **Transformation shortcut metrics**: These metrics can be created on reports using a similar workflow as derived metrics that are created as shortcut metrics. For information about the difference between

Copyright © 2024 All Rights Reserved

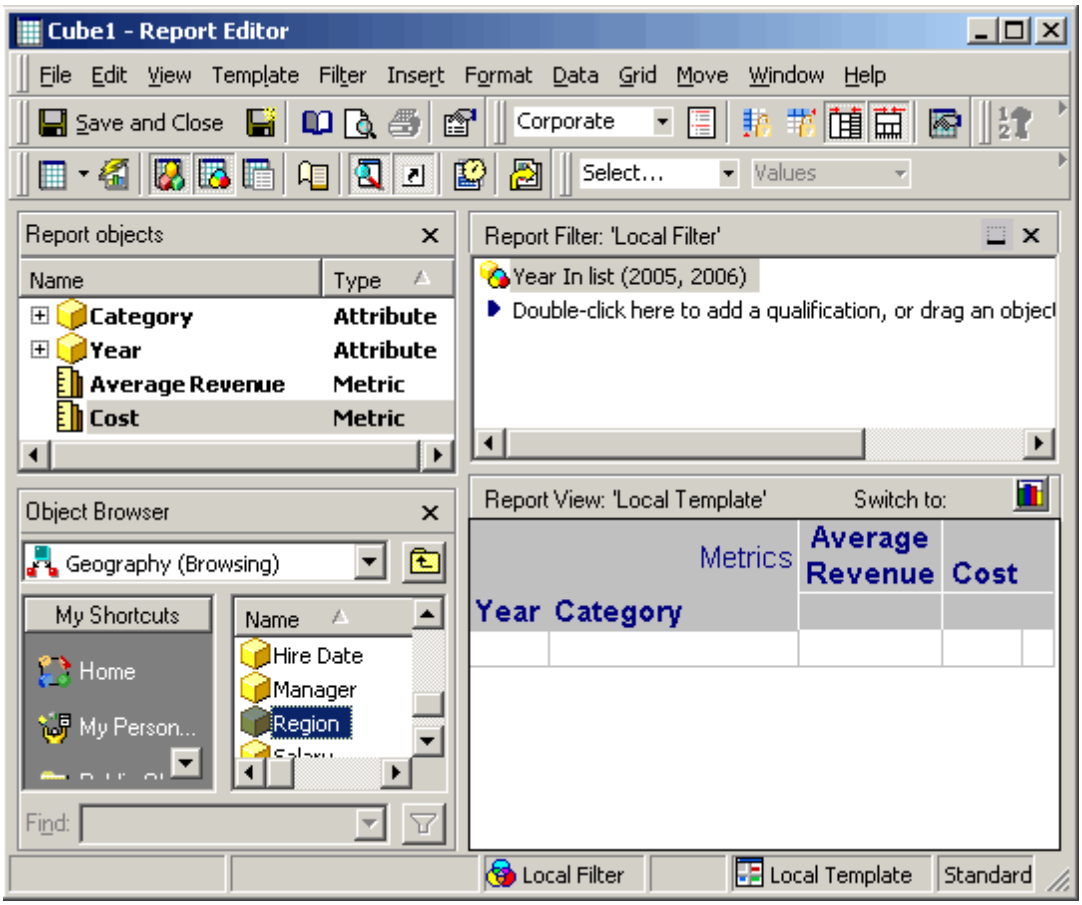
transformation shortcut metrics and shortcut metrics that are also derived metrics, see *Using rank and percent-to-total metric analysis , page 189*. As an alternative, you can create transformation metrics using the Metric Editor. Metrics that use transformations in this way can be included in Intelligent Cubes. To create transformation metrics using the Metric Editor, see the Advanced Reporting Help.

- **Prompts**: Prompts cannot be included in Intelligent Cubes. However, reports that access Intelligent Cubes can use prompts that access only data included in the Intelligent Cube. This saves processing time by ensuring that re-execution against the data warehouse is not necessary.

Building an Intelligent Cube

Building an Intelligent Cube is similar to building a report. You can add data for your Intelligent Cube by including objects such as attributes, metrics, filters, and so on as report objects or report filters. For a list of objects and features that cannot be included in the definition of Intelligent Cubes, see above.

Copyright © 2024 All Rights Reserved



Since Intelligent Cubes are used simply to share a set of data, no data or report results are displayed when you execute an Intelligent Cube. However, executing an Intelligent Cube publishes the Intelligent Cube, which can then be accessed as a set of data for multiple reports (see *Publishing Intelligent Cubes, page 43*).

Prerequisites

- You need the Use Intelligent Cube Editor privilege to create Intelligent Cubes. This privilege is part of OLAP Services.

Copyright © 2024 All Rights Reserved

To build an Intelligent Cube

1. In Developer, from the **File** menu select **New**, and then **Intelligent Cube**. The New Intelligent Cube dialog box opens.

   If the New Intelligent Cube dialog box does not open, from the **Tools** menu, select **Developer Preferences**. Expand the Object Templates category, select **General**, and from the Show templates for the following objects list, select **Report**. Click **OK** to accept your changes, and then repeat the previous step to open the New Grid dialog box.

2. Select **Empty Intelligent Cube** and click **OK**. The Report Editor opens.
3. Add objects such as attributes, metrics, and so on for the Intelligent Cube, the same way you would add report objects.
4. Create a filter for the Intelligent Cube as needed.

   If you create a filter on an Intelligent Cube, any data that is restricted from the Intelligent Cube is not available for any reports that connect to the Intelligent Cube. While this can help reduce the size of the Intelligent Cube, it also reduces the amount of data available in the Intelligent Cube.

5. Click **Save and close** to save the Intelligent Cube and close the Report Editor.
6. To publish an Intelligent Cube, see *Publishing Intelligent Cubes, page 43*.

Building Intelligent Cubes from MDX data sources, or using Freeform SQL

You can create Intelligent Cubes with data from an MDX data source, such as SAP BW, Hyperion Essbase, or Microsoft Analysis Services. To create an Intelligent Cube based on MDX data in Developer, open the **File** menu, select **New**, and select **Intelligent Cube**. In the New Intelligent Cube dialog

Copyright © 2024 All Rights Reserved

box, select the **MDX Sources** tab. For instructions on configuring MDX sources, refer to the MDX Cube Reporting Help.

You can also create Intelligent Cubes by directly querying your data using SQL. You can use Freeform SQL to write your own SQL statements, or Query Builder to create a query using a graphical interface. These options are available in the New Intelligent Cube dialog box, under the **ODBC Sources** tab. For information on using Freeform SQL and Query Builder to connect to ODBC data sources, see the Advanced Reporting Help.

Converting reports into Intelligent Cubes

If you have an existing report that contains a set of objects you want to use in an Intelligent Cube, you can convert the report into an Intelligent Cube.

Creating an Intelligent Cube by converting a report allows you to view the set of data you are including in your Intelligent Cube. This is an easy way to share a report's set of data among multiple reports.

Intelligent Cubes do not display report results in the same way as reports can. Instead of building an Intelligent Cube from the start, you can build a report; execute it; view the report data in grid, graph, or grid and graph mode; and then convert the report to an Intelligent Cube. With this method you can verify that you have the set of data you want, rather than just the required report objects, before you publish it as an Intelligent Cube.

When you convert a report to an Intelligent Cube, some parts of the report are not included in the resulting Intelligent Cube. Intelligent Cubes are not used for the same display and analysis purposes as a report. Intelligent Cubes simply act as a sharable set of data. Therefore, when a report is converted into an Intelligent Cube, some of the display and analysis features are no longer necessary.

The procedure below explains the high-level steps for converting a report to an Intelligent Cube. The procedure assumes you have already created a report.

Copyright © 2024 All Rights Reserved

To convert a report to an Intelligent Cube

1. Right-click the report and select **Edit**. The Report Editor opens.
2. Choose **Data** > **Intelligent Cube Options** > **Convert to Intelligent Cube**.
3. If the report contains objects that cannot be included in the Intelligent Cube, one of the following messages is displayed:
   - If the report includes OLAP Services features such as view filters, derived metrics, or dynamic aggregation, you are prompted to automatically remove these features as part of the conversion process. Click **Yes** to have these features automatically removed so that the report can be converted into an Intelligent Cube.
   - If the report includes features such as consolidations, custom groups, or prompts, a warning message is displayed that explains that these objects cannot be included in the Intelligent Cube. Click **OK** to close the warning message and then manually remove the objects from the report. You can then attempt to convert the report to an Intelligent Cube again.
4. After the conversion process is completed successfully, save the Intelligent Cube.
5. You must publish an Intelligent Cube to make it available for multiple reports to access and report on its set of data. To publish an Intelligent Cube, see *Publishing Intelligent Cubes, page 43*.

Updating Intelligent Cubes using schedules

While you are creating or modifying an Intelligent Cube, you can schedule when an Intelligent Cube should be re-executed against the data warehouse to update its data. For example, you can schedule to re-execute and republish an Intelligent Cube when a database load occurs.

Copyright © 2024 All Rights Reserved

The procedure below describes the steps to create a subscription that schedules the re-execution and republication of a single Intelligent Cube. You can also schedule multiple Intelligent Cubes to be updated using a single subscription, which is described in *Publishing Intelligent Cubes using a schedule, page 45*.

Prerequisites

- An Intelligent Cube has been created.
- You have created the schedule to subscribe the publication of an Intelligent Cube to. For information on creating schedules, see the System Administration Help.

To schedule an Intelligent Cube to be updated

1. In Developer, browse to an Intelligent Cube.
2. Right-click the Intelligent Cube, choose **Schedule Delivery To** > **Refresh Cube**. The Subscription Editor opens.
3. Select a schedule from the Schedule drop-down list.
4. Select other schedule options as required, and click **OK**.

Once the schedule is triggered the Intelligent Cube is re-executed against the data warehouse and published to the Intelligent Cube Monitor.

Enabling ROLAP drilling for reports accessing Intelligent Cubes

Reports that access an Intelligent Cube can be granted full ROLAP access to the data warehouse through drilling. This means that you can enable drilling outside of the Intelligent Cube to access the full set of data available in the data warehouse.

While this extends the analysis and data access capabilities of reports that access Intelligent Cubes, drilling outside of an Intelligent Cube can require additional load on the Intelligence Server and data warehouse. This is

Copyright © 2024 All Rights Reserved

because drilling outside of an Intelligent Cube requires a new report to be executed against the data warehouse.

All reports that access Intelligent Cubes can drill within the data included in an Intelligent Cube. This provides ROLAP-type analysis without having to re-execute against the data warehouse. For example, an Intelligent Cube includes Year and Quarter. A report accessing the Intelligent Cube only includes Year on the report. On the report, you can drill down from Year to Quarter, which returns the results without any extra load on the data warehouse or Intelligence Server.

The decision to enable or disable drilling outside of an Intelligent Cube depends on a few factors. You should consider the size and complexity of your Intelligent Cubes when deciding whether to enable drilling outside of an Intelligent Cube. While enabling drilling outside of relatively small Intelligent Cubes can give the benefit of ROLAP analysis through drilling, enabling this analysis on relatively large Intelligent Cubes has the potential to cause increased load on your data warehouse and Intelligence Server.

By enabling drilling outside of an Intelligent Cube, reports that access the Intelligent Cube have ROLAP access to data through drilling. For example, an Intelligent Cube includes Year but not Quarter. A report accessing the Intelligent Cube includes Year and drilling outside of the Intelligent Cube is enabled. On the report, you drill down from Year to Quarter. This causes a new report to be created and executed against the data warehouse.

The benefit of enabling this extra analysis can come with a performance cost. As mentioned above, when you drill outside of an Intelligent Cube a new report is created and executed against the data warehouse. This action does not take advantage of the data stored in the Intelligent Cube.

When this drilled-to report is created, only objects that were on the report layout of the report you drilled from are included in the drilled-to report. Any objects that are only in the Report Objects pane of the report you drilled from are not included in the drilled-to report. This can help reduce the size of the drilled-to report. However, if you are drilling from a report that accesses

Copyright © 2024 All Rights Reserved

a large Intelligent Cube, it is possible that a user could include all objects of an Intelligent Cube on a report. Drilling outside of the Intelligent Cube on such a report could cause excessive load on the data warehouse and Intelligence Server.

Prerequisites

- You need the Use Intelligent Cube Editor privilege. This privilege is part of OLAP Services.

To enable or disable drilling outside of an Intelligent Cube for reports accessing a specific Intelligent Cube

1. In Developer, browse to an Intelligent Cube.
2. Right-click the Intelligent Cube, and select **Edit**. The Intelligent Cube is opened in the Report Editor.
3. Choose **Data** > **Configure Intelligent Cube**. The Intelligent Cube Options dialog box opens.
4. From the **Categories** list, expand **Options**, and then select **General**.
5. In the **Drilling** area, clear the **Use default settings** check box.
6. Select the **Allow reports to drill outside the Intelligent Cube** check box.

   You can clear this check box to disable drilling outside of an Intelligent Cube.
7. Click **OK**.
8. Click **Save and Close**.
9. To make your changes available for reports accessing the Intelligent Cube, you must publish the Intelligent Cube, which is described in *Publishing Intelligent Cubes, page 43*.

Copyright © 2024 All Rights Reserved

To enable or disable drilling outside of an Intelligent Cube for reports accessing any Intelligent Cubes in a project

This procedure enables drilling outside of all Intelligent Cubes within a project.

1. In Developer, log in to a project with a user account with administrative privileges.
2. Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.
3. From the **Categories** list, expand **Intelligent Cubes**, and then select **General**.
4. Select the **Allow reports to drill outside the Intelligent Cube** check box.
5. You can clear this check box to disable drilling outside of Intelligent Cubes.
6. Click **OK**.
7. To make your changes available for reports accessing the Intelligent Cubes, you must publish the Intelligent Cubes, which is described in *Publishing Intelligent Cubes, page 43*.

Localizing Intelligent Cubes in multiple languages

If your MicroStrategy project supports the localization of information into multiple languages, you can also support this localization to provide Intelligent Cubes in multiple languages.

When an Intelligent Cube is defined to support a language, data is available for the Intelligent Cube in that language. When a user logs in to MicroStrategy in the locale for that language and creates or views a report that accesses the Intelligent Cube, all data that has been localized is returned.

Copyright © 2024 All Rights Reserved

For example, a report connected to an Intelligent Cube includes Region, Category, Profit, and Discount. The Intelligent Cube is defined to include information for the English and Italian locales. This report connected to the Intelligent Cube is executed by a user with an English locale. This same report is also executed by a user with an Italian locale. The report results displayed to the users in the two locales are shown below:

Report viewed in English locale

Region	Category (Metrics)	Profit	Discount
Central	Books	$55,495	$ 7,813
	Electronics	$935,123	$ 162,014
	Movies	$16,347	$ 7,799
	Music	$78,357	$ 35,082
Mid-Atlantic	Books	$46,461	$ 6,648
	Electronics	$3,933,032	$ 736,291
	Movies	$14,351	$ 6,508
	Music	$13,737	$ 6,021
Northeast	Books	$455,964	$ 62,663
	Electronics	$1,500,514	$ 270,429
	Movies	$27,367	$ 12,099
	Music	$25,895	$ 11,363
Northwest	Books	$19,734	$ 2,757
	Electronics	$1,604,110	$ 289,165
	Movies	$5,639	$ 2,712
	Music	$5,508	$ 2,436
South	Books	$216,663	$ 34,020
	Electronics	$739,063	$ 136,079
	Movies	$12,905	$ 6,155
	Music	$11,882	$ 6,145
Southeast	Books	$25,609	$ 3,599
	Electronics	$419,847	$ 71,154
	Movies	$37,704	$ 17,212
	Music	$6,965	$ 3,197
Southwest	Books	$41,576	$ 5,739
	Electronics	$686,421	$ 126,344
	Movies	$60,787	$ 29,791
	Music	$11,294	$ 5,229
Web	Books	$29,195	$ 4,339
	Electronics	$469,635	$ 88,912
	Movies	$42,611	$ 20,360
	Music	$7,952	$ 3,786

Report viewed in Italian locale

Region	Category (Metriche)	Profit	Sconto
Atlantico centrale	Libri	$46,461	$ 6,648
	Elettronica	$3,933,032	$ 736,291
	Film	$14,351	$ 6,508
	Musica	$13,737	$ 6,021
Centrale	Libri	$55,495	$ 7,813
	Elettronica	$935,123	$ 162,014
	Film	$16,347	$ 7,799
	Musica	$78,357	$ 35,082
Nord-est	Libri	$455,964	$ 62,663
	Elettronica	$1,500,514	$ 270,429
	Film	$27,367	$ 12,099
	Musica	$25,895	$ 11,363
Nord-ovest	Libri	$19,734	$ 2,757
	Elettronica	$1,604,110	$ 289,165
	Film	$5,639	$ 2,712
	Musica	$5,508	$ 2,436
Sud	Libri	$216,663	$ 34,020
	Elettronica	$739,063	$ 136,079
	Film	$12,905	$ 6,155
	Musica	$11,882	$ 6,145
Sud-est	Libri	$25,609	$ 3,599
	Elettronica	$419,847	$ 71,154
	Film	$37,704	$ 17,212
	Musica	$6,965	$ 3,197
Sud-ovest	Libri	$41,576	$ 5,739
	Elettronica	$686,421	$ 126,344
	Film	$60,787	$ 29,791
	Musica	$11,294	$ 5,229
Web	Libri	$29,195	$ 4,339
	Elettronica	$469,635	$ 88,912
	Film	$42,611	$ 20,360
	Musica	$7,952	$ 3,786

The same report returns data that matches the user's locale. This is because the Intelligent Cube has been defined to include localized data.

The SQL view of these reports shown below demonstrates how the data is returned in different languages:

Copyright © 2024 All Rights Reserved

SQL generated for English locale

```
Cube SQL Statements:

Pass0 -    Duration: 0:00:00.00


select     [Category]@[CATEGORY_ID],
           [Category]@[CATEGORY_DESC],
           [Region]@[REGION_ID],
           [Region]@[REGION_NAME],
           sum([Profit])@{[Category],[Region]},
           sum([Discount])@{[Category],[Region]}
from       Localized I Cube
```

SQL generated for Italian locale

```
Istruzioni SQL cubo:

Passaggio0 -    Durata: 0:00:00.00


select     [Category]@[CATEGORY_ID],
           [Category]@[CATEGORY_DESC_IT],
           [Region]@[REGION_ID],
           [Region]@[REGION_NAME_IT],
           sum([Profit])@{[Category],[Region]},
           sum([Discount])@{[Category],[Region]}
from       Localized I Cube
```

Notice that different columns of data were returned for the report based on the locale used to run the report. This demonstrates a column-based solution to localizing your data. It is recommended that you use a column-based solution rather than a connection mapping-based solution to localize your data for Intelligent Cubes. For information on this recommendation, see the System Administration Guide.

Localizing an Intelligent Cube does not localize any data, it only returns data that has already been configured in a MicroStrategy project as part of a localization solution. For information on defining localization rules for your MicroStrategy projects, see the Project Design Help.

Providing localized data causes the Intelligent Cube size to be larger than if it supported only a single locale. However, providing localized data in Intelligent Cubes is necessary if users expect reports that return Intelligent Cube data to return localized data to reflect their locale.

The steps below let you define Intelligent Cubes to support various languages.

Prerequisites

- An Intelligent Cube has been created.
- The project for the Intelligent Cube has been configured to support multiple languages. For information on localizing projects, see the Project Design Help.

Copyright © 2024 All Rights Reserved

- You need the Use Intelligent Cube Editor privilege. This privilege is part of OLAP Services.

To support multiple languages in an Intelligent Cube

1. In Developer, browse to an Intelligent Cube.
2. Right-click the Intelligent Cube and select **Edit**. The Intelligent Cube opens in the Report Editor.
3. Choose **Data** > **Configure Intelligent Cube**. The Intelligent Cube Options dialog box opens.
4. From the **Categories** list, expand **Options**, and then select **Data Languages**.
5. Select from the following localization options:
   - **Use project default data language (Default)**: Select this option to provide data only in the default language for the project. The default language for the project is highlighted in blue.
   - **All project data languages**: Select this option to provide data for the languages available for the project.
   - **Select specific languages**: Select this option to select from all languages available for the project. Data is provided for the languages you select.

     All of these options are based on the language options defined for the project, which can be modified from the Project Configuration Editor. For information on localizing projects, see the Project Design Help.
6. Click **OK** and then **Save and Close**.

To make your changes available for reports accessing the Intelligent Cube, you must publish it.

Copyright © 2024 All Rights Reserved

Enabling dynamic sourcing of Intelligent Cubes

The general workflow of reporting on data in Intelligent Cubes is to create a report that is linked to a published Intelligent Cube. Dynamic sourcing extends the accessibility of Intelligent Cubes by allowing standard reports to access any published Intelligent Cubes that can satisfy the data requirements of the report. This connection is performed automatically without the user having to explicitly connect their report to a published Intelligent Cube. This allows users performing ROLAP analysis to work completely within a pre-created set of data without having to manually find and select the Intelligent Cube that fits their analysis.

See *Chapter 8, Dynamic Sourcing* for more information.

Publishing Intelligent Cubes

Publishing an Intelligent Cube retrieves data from the data warehouse and stores this data as an Intelligent Cube in Intelligence Server memory. Once an Intelligent Cube is stored in Intelligence Server memory, multiple reports can be created to view and analyze the set of published data.

When an Intelligent Cube is published, the Intelligent Cube Monitor displays the Intelligent Cube. You can manage your published Intelligent Cubes from the Intelligent Cube Monitor. For more information on managing Intelligent Cubes, see the System Administration Help.

Publishing or re-publishing an Intelligent Cube can require significant memory resources. It is recommended that you use incremental refresh settings to add new data to the Intelligent Cube. For information on setting up incremental refreshes, see *Updating Intelligent Cubes without re-processing: Incremental Refresh, page 48*.

Publishing Intelligent Cubes manually

The procedure below provides the steps to manually publish an Intelligent Cube.

Copyright © 2024 All Rights Reserved

The act of publishing an Intelligent Cube can require memory resources approximately two times greater than the size of an Intelligent Cube. This can affect performance of your Intelligence Server as well as the ability to publish the Intelligent Cube. To plan for these memory requirements, see the System Administration Help.

Prerequisites

- You need the Publish Intelligent Cube (Developer) and/or Web Publish Intelligent Cube (Web) privileges. These privileges are part of OLAP Services.

To publish an Intelligent Cube manually

The steps below show you how to publish an Intelligent Cube in Developer. You can follow the same high-level steps in Web to browse to an Intelligent Cube, and then run the Intelligent Cube to publish it.

1. In MicroStrategy Developer, browse to the location of the Intelligent Cube to publish.

   If you define Intelligent Cubes to use the connection mapping configured for your project, the Intelligent Cubes are published with the connection mapping defined for the user account employed to publish the Intelligent Cubes. For information on defining Intelligent Cubes to support connection mapping, see the System Administration Help.

2. Right-click the Intelligent Cube, and select **Run**.

   Rather than displaying report results, a message is displayed indicating that the Intelligent Cube has been published. After the Intelligent Cube is published, the Intelligent Cube appears in The Intelligent Cube Monitor, which you can use to manage your published Intelligent Cubes.

Copyright © 2024 All Rights Reserved

Publishing Intelligent Cubes using a schedule

As an alternative manually publishing Intelligent Cubes, you can schedule the publication of Intelligent Cubes to occur at a time when the load on your Intelligence Server and data warehouse is low. This allows you to publish Intelligent Cubes without affecting performance time of Intelligence Server during peak usage.

You can also schedule the publication of Intelligent Cubes to update Intelligent Cube data when events occur that would make the Intelligent Cube data outdated. For example, you can schedule to republish an Intelligent Cube when a database load occurs. The procedure below creates a subscription that can schedule multiple Intelligent Cubes to be updated. Alternatively, you can define an Intelligent Cube to be updated on a schedule during Intelligent Cube creation, which is described in *Updating Intelligent Cubes using schedules, page 35*.

You schedule an Intelligent Cube for publication by creating a subscription for the Intelligent Cube. The procedure below provides steps to create a subscription to schedule the publication of an Intelligent Cube.

You cannot create subscriptions for Intelligent Cubes that have been created using the Import Data feature in MicroStrategy Web. For information on the Import Data feature, refer to the MicroStrategy Web Help.

The act of publishing an Intelligent Cube can require memory resources approximately two times greater than the size of an Intelligent Cube. This can affect performance of your Intelligence Server as well as the ability to publish the Intelligent Cube. To plan for these memory requirements, see the System Administration Help.

To schedule the publication of Intelligent Cubes

1. In MicroStrategy Developer, log in to a project source using administrative privileges.

Copyright © 2024 All Rights Reserved

2. Choose **Administration** > **Scheduling** > **Subscription Creation Wizard**. The Subscription Creation Wizard opens.
3. If the welcome page is shown, review the information and click **Next**.
4. On the Subscription Wizard - Specify Characteristics page, begin defining the subscription using the following drop-down lists:
   - **Choose a schedule for this set of subscriptions**: Select a schedule to trigger the publication of Intelligent Cubes. A schedule can be time- or event-triggered.
   - **Choose a project from which reports/documents will be delivered to the recipients**: Select a project that contains Intelligent Cubes to schedule for publication. When publishing Intelligent Cubes, the Intelligent Cubes are not delivered to a specific recipient, they are instead published to the Intelligent Cube Monitor.
   - **Choose a Delivery Type**: Select **Cube Refresh**. This delivery type publishes Intelligent Cubes to the Intelligent Cube Monitor.
5. Click **Next**. The Subscription Wizard - Choose Reports/Documents page opens.
6. In the **Available objects** pane, browse to and select Intelligent Cubes to schedule for publication. Once selected, click the right arrow (>) to move any selected Intelligent Cubes to the **Selected objects** pane.
7. Click **Next**. The Subscription Wizard - Choose Recipients page opens.
8. Keep the default user that is selected or select a different user with administrative privileges, and click **Next**. The Subscription Wizard - Specify Subscription Properties page opens.
9. You can choose from the following subscription properties:
   - **Run subscription immediately**: Select this check box if the Intelligent Cube should be published immediately. This causes any

Copyright © 2024 All Rights Reserved

Intelligent Cubes included in the subscription to be published immediately. The Intelligent Cubes will also be re-published based on the schedule used in the subscription.

- **Expire subscription on**: Select this check box if the subscription should expire. If you select an expiration date, the subscription is deleted on that date and Intelligent Cubes included in the subscription are no longer scheduled for publication. The Intelligent Cubes can be published manually.

10. Click **Next**. The Subscription Wizard - Summary page opens.
11. Review the summary information, and click **Finish** to create the subscription. If you selected to run the subscription immediately, the Intelligent Cubes are executed against the data warehouse and published to the Intelligent Cube Monitor.

You can view and edit the new subscription from the Subscription Manager. For example, you can select a different schedule to use for the subscription or modify the expiration date for the subscription. For information on subscriptions and the Subscription Manager, see the System Administration Help.

Unpublishing an Intelligent Cube

You can remove a published Intelligent Cube so that its data is no longer accessible by reports. This action does not delete the Intelligent Cube object saved in a MicroStrategy project, it only removes the Intelligent Cube from the Intelligent Cube Monitor.

Unpublishing an Intelligent Cube prevents reports that access the Intelligent Cube from being able to load the Intelligent Cube into Intelligence Server memory. This gives you more administrative control of when to make an Intelligent Cube available to reports. However, be aware that unpublishing an Intelligent Cube means that any reports that access the Intelligent Cube cannot be executed.

Copyright © 2024 All Rights Reserved

This same administrative control can be enforced by de-activating an Intelligent Cube. De-activating an Intelligent Cube saves the Intelligent Cube to secondary storage, such as a hard disk. When you re-activate the Intelligent Cube, the copy in secondary storage is loaded back into Intelligence Server memory. This option is ideal when an Intelligent Cube should not be used for some length of time, but after that should be available again in its current form. For information on de-activating an Intelligent Cube, see the System Administration Help.

To unpublish an Intelligent Cube

1. In MicroStrategy Developer, log in to a project source with administrative privileges.
2. From the **Folder List**, expand **Administration**, then expand **System Monitors**, then expand **Caches**, and select **Intelligent Cubes**. The Intelligent Cube Monitor is displayed.
3. Right-click an Intelligent Cube and select **Delete**. The Intelligent Cube is unpublished from the Intelligent Cube Monitor and its data cannot be accessed by reports.

Reporting and analyzing data with Intelligent Cubes

Intelligent Cube Technology allows reports to use OLAP Services features to display and analyze data using an in-memory copy of data rather than querying the data warehouse. See Reporting on Intelligent Cubes for more information.

Updating Intelligent Cubes without re-processing: Incremental Refresh

By default, if the data in an Intelligent Cube needs to be updated, it is re-executed. All the data for the Intelligent Cube is loaded from the data

Copyright © 2024 All Rights Reserved

warehouse into Intelligence Server's memory, and the existing data for the Intelligent Cube is overwritten.

However, if the Intelligent Cube is updated based on one or more attributes, such as Month or Region, you can set up incremental refresh settings to update the Intelligent Cube with only new data. This can reduce the time and system resources necessary to update the Intelligent Cube periodically.

For example, you have an Intelligent Cube that contains weekly sales data. At the end of every week, this Intelligent Cube must be updated with the sales data for that week. You can set up incremental refresh settings so that only data for one week is added to the Intelligent Cube.

Prerequisites

For an Intelligent Cube to qualify for an incremental refresh, it must meet the following requirements:

- The updates to the Intelligent Cube must be according to attributes only.
- For example, an Intelligent Cube that must be updated with data for a month or region qualifies for an incremental refresh.
- The Intelligent Cube must not be re-processed based on a metric.
- For example, an Intelligent Cube that contains data for the top 200 stores by Revenue does not qualify for an incremental refresh.

sDefining an incremental refresh for an Intelligent Cube

Once you have defined an Intelligent Cube, you can use the following methods to define the incremental refresh options, depending on your requirements:

- Define Intelligent Cube republish settings. This is recommended if the Intelligent Cube must be updated along only one attribute, or if updates along multiple attributes must be made simultaneously.
- Define an incremental refresh filter or report. This is recommended if the Intelligent Cube must be updated along different dimensions at different

Copyright © 2024 All Rights Reserved

times, or if you want to use the results of a report to update the Intelligent Cube.

If you choose to define an incremental refresh filter or report, you should not re-publish the Intelligent Cube by double-clicking, or by publishing it on a schedule. Doing so will overwrite any changes made by the incremental refresh filter or report.

Defining Intelligent Cube refresh settings

The refresh settings change the way the Intelligent Cube is updated when you re-execute it manually, or when it is published on a schedule, as described in *Publishing Intelligent Cubes using a schedule, page 45*.

This is recommended if the Intelligent Cube must be updated along only one attribute, or if Intelligent Cube must be updated along multiple attributes simultaneously.

Prerequisites

- The Intelligent Cube's definition must include a filter that qualifies on the attributes for which the Intelligent Cube must be updated.

  For example, if the Intelligent Cube must be updated with new data for the Store attribute, the filter must qualify on Store.

To define Intelligent Cube refresh settings

1. In Developer, navigate to the Intelligent Cube you want to define republish settings for.
2. Right-click on the Intelligent Cube, and select **Edit**. The Intelligent Cube Editor opens.
3. On the toolbar, choose **Data** > **Configure Intelligent Cube**. The Intelligent Cube Options dialog box opens.

Copyright © 2024 All Rights Reserved

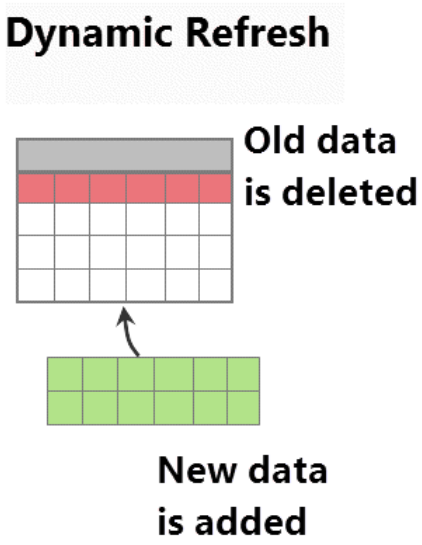
4. Navigate to the **Data Refresh** category.
5. Choose one of the following options:

- **Full Refresh**: This is the default. The Intelligent Cube's SQL is re-executed, and all the data is loaded from the data warehouse into Intelligence Server's memory.

  Use this option under the following conditions:

  - If the data in the Intelligent Cube is outdated.
  - If the Intelligent Cube must be re-processed based on a metric. For example, an Intelligent Cube that contains data for the top 200 stores by Profit must be re-calculated every time it is updated, and thus must use the Full Refresh option.

- **Dynamic refresh**: The Intelligent Cube's filter is evaluated. If new data is returned, it is added to the Intelligent Cube, and data that no longer meets the filter's criteria is deleted from the Intelligent Cube. The dynamic refresh option is illustrated in the image below.



Copyright © 2024 All Rights Reserved

Select this option for Intelligent Cubes that have a rolling set of data—for example, an Intelligent Cube that always contains data for the past six months.

If the data to be added and deleted cannot be determined, the **Full Refresh** option is used as a fallback.

- **Update**: The Intelligent Cube's filter is evaluated. If new data is returned, it is added to the Intelligent Cube, and if the data returned is already in the Intelligent Cube, it is updated where applicable.

  Select this option if your data is updated often—for example, an Intelligent Cube that contains daily sales data, and is updated at the end of every day.

- **Insert**: The Intelligent Cube's filter is evaluated. If new data is returned, it is added to the Intelligent Cube. Data that is already in the Intelligent Cube is not altered.

  Select this option if old data does not change once it is saved to your data warehouse.

Defining an incremental refresh filter or report

For complex update requirements, such as updating an Intelligent Cube for different dimensions at different intervals, you can define an incremental refresh filter or report to update your Intelligent Cube.

For example, an Intelligent Cube contains monthly sales data for 2009 and 2010 for year-on-year comparison. Once the year 2011 begins, you only need to keep the data for 2010, and the data for 2009 can be removed from the Intelligent Cube. You can define one incremental refresh that runs at the end of every month, and adds that month's data to the Intelligent Cube, and a second incremental refresh that deletes the previous year's data at the end of every year.

You can also add incremental refreshes to add data for other dimensions, such as Country. For example, you have an Intelligent Cube that contains

Copyright © 2024 All Rights Reserved

data for the USA, UK and France, and you want to add data for Germany. You can define an incremental refresh that fetches all the data for Germany, and adds it to the cube.

You can define incremental refreshes using filters or reports, as described below:

- Filter: The data returned by a filter is compared to the data that is already in the cube. By default, the filter defined for the Intelligent Cube is used as the filter for the incremental refresh.
- Report: The results of a report are used to populate the Intelligent Cube. By default, the report template used is the same as the Intelligent Cube's template.

If you choose to define an incremental refresh filter or report, you should not re-publish the Intelligent Cube by double-clicking it, or by publishing it on a schedule. Doing so will overwrite any changes made by the incremental refresh filter or report.

Defining an incremental refresh filter

You can define an incremental refresh filter to update the data in an Intelligent Cube. This is the default option for both ROLAP and MDX Intelligent Cubes, and is unavailable for Intelligent Cubes created using Freeform SQL queries or Query Builder.

To define an incremental refresh filter

1. Navigate to the Intelligent Cube for which you want to define the incremental refresh.
2. Right-click the Intelligent Cube, and select **Define incremental refresh**. The Incremental Refresh Options dialog box opens.
3. Under **Refresh type**, select one of the following options.

Copyright © 2024 All Rights Reserved

- **Update**: The incremental refresh filter is evaluated. If new data is available, it is fetched and added to the Intelligent Cube, and if the data available is already in the Intelligent Cube, it is updated where applicable.
- **Insert**: The incremental refresh filter is evaluated. If new data is available, it is fetched and added to the Intelligent Cube. Data that is already in the Intelligent Cube is not altered.
- **Delete**: The incremental refresh filter is evaluated. The data that meets the filter's criteria is deleted from the cube. For example, if the Intelligent Cube contains data for 2021, 2022, and 2023, and the filter is for data for 2022, all the data for 2022 is deleted from the cube.
- **Update only**: The incremental refresh filter is evaluated. If the data available is already in the Intelligent Cube, it is updated where applicable. No new data is added to the Intelligent Cube.
- **Replace**: The incremental refresh filter is evaluated. Old cube data is deleted using the report filter and replaced with new data. Your metadata must be upgraded to MicroStrategy 2021 or above to use this option.

You can change these options at any time by opening the incremental refresh in the Report Editor, and from the **Data** menu, selecting **Configure incremental refresh options**.

4. Click **OK**. The Report Editor opens with a new incremental refresh. If the Intelligent Cube's definition included a filter, it appears in the Report Filter pane.
5. In the Report Filter pane, edit the filter if applicable, or create a new filter.
6. To preview the data that will be updated in the Intelligent Cube, from the **View** menu, select **Preview Data**. The data is displayed in a grid

Copyright © 2024 All Rights Reserved

view.

If your security filter prevents you from viewing some data, the preview only displays data that you can view. However, when the incremental refresh is executed, all the data for the filter is updated in the Intelligent Cube, regardless of security filters.

7. To execute the incremental refresh immediately, click the **Run Report** toolbar button.
8. To save and close the incremental refresh, click **Save and Close**.

Defining an incremental refresh report

You can define a report to update an Intelligent Cube. The results of a report are compared to the data in the Intelligent Cube, and the Intelligent Cube is updated accordingly.

If you are updating an Intelligent Cube based on a Freeform SQL or Query Builder report, this is the only available option.

Prerequisites

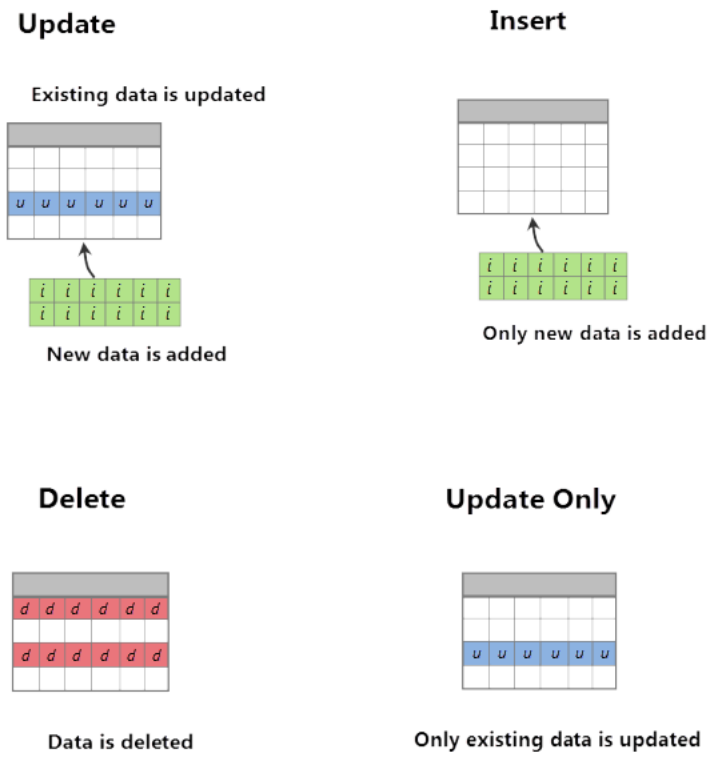
- The report must use all the attributes, and at least one metric from the Intelligent Cube that is being updated. Note that for metrics that are not on the report's template, data is not updated.
- All attributes and metrics in the report's definition should be identical to the attributes and metrics in the Intelligent Cube. You can verify this by checking the Global Unique Identifiers (GUIDs), by right-clicking the attribute or metric, and choosing **Properties**.

To define an incremental refresh report

1. Navigate to the Intelligent Cube for which you want to define the incremental refresh.

Copyright © 2024 All Rights Reserved

2. Right-click the Intelligent Cube, and select **Define Incremental Refresh**. The Incremental Refresh Options dialog box opens.
3. Under **Refresh type**, select one of the following options. The differences between the options are illustrated in the image below.



- **Update**: The incremental refresh report is evaluated. If new data is available, it is fetched and added to the Intelligent Cube, and if the data returned is already in the Intelligent Cube, it is updated where applicable.
- **Insert**: The incremental refresh report is evaluated. If new data is available, it is fetched and added to the Intelligent Cube. Data that is already in the Intelligent Cube is not altered.

Copyright © 2024 All Rights Reserved

- **Delete**: The incremental refresh report is evaluated. The data that is meets the report's definition is deleted from the cube. For example, if the Intelligent Cube contains data for 2021, 2022, and 2023, and the report returns data for 2022, all the data for 2022 is deleted from the cube.

  If you are using the **Delete** option,you may use a report with only a subset of the attributes present in the Intelligent Cube.

- **Update only**: The incremental refresh report is evaluated. If the data available is already in the Intelligent Cube, it is updated where applicable. No new data is added to the Intelligent Cube.

  The **Replace** option is not available here, since it can only be used for incremental refresh filters.

  You can change these options at any time by opening the incremental refresh in the Report Editor, and from the **Data** menu, selecting **Configure Incremental Refresh options**.

4. Navigate to the **Advanced** category.
5. Select **Report** and click **OK**. The Report Editor opens, with a new incremental refresh report. By default, the report's template contains all the attributes and metrics from the Intelligent Cube.
6. Edit the report, if necessary.
7. To preview the data that will be updated in the Intelligent Cube, from the **View** menu, select **Preview Data**. The data is displayed in a grid view.

   If your security filter prevents you from viewing some data, the preview only displays data that you can view. However, when the incremental refresh is executed, all the data for the filter is updated in the Intelligent Cube, regardless of security filters.

Copyright © 2024 All Rights Reserved

8. To execute the incremental refresh immediately, click **Run Report**.
9. To save and close the incremental refresh, click **Save and Close**.

Improving the performance of large Intelligent Cubes: Partitioning

If your Intelligent Cubes contain large quantities of data, you can improve their performance by dividing the datasets into multiple segments, called partitions. The partitions are processed simultaneously, distributed across the processor cores of your Intelligence Server.

To partition an Intelligent Cube

1. In Developer, log in to your project and navigate to the Intelligent Cube.
2. Right-click the Intelligent Cube, and select **Edit**. The Intelligent Cube Editor opens.
3. Choose **Data** > **Configure Intelligent Cube**. The Intelligent Cube Options dialog box opens.
4. Select the **Data Partition** category. The Data Partition options are displayed.
5. From the **Partition Attribute** drop-down, select the partition attribute for the Intelligent Cube. For the criteria to determine an appropriate distribution key, see *Requirements for the partition attribute, page 86*.
6. In the **Number of Partitions** field, type the number of partitions to create. The number should be less than or equal to the number of processor cores on your Intelligence Server machine.
7. If your data warehouse supports parallel SQL queries, enable the **Fetch data slices in parallel from the warehouse** check box.
8. Click **OK**.

Copyright © 2024 All Rights Reserved

Reporting and analyzing data with OLAP Services features

While creating and viewing reports in MicroStrategy, you can use various OLAP Services features to report and analyze data using an in-memory copy of data, rather than querying the data warehouse.

This provides both OLAP Services features and full ROLAP analysis on your report with no overhead to create or maintain an Intelligent Cube, or to link your report to an Intelligent Cube.

Along with all of the standard reporting features available in MicroStrategy, the OLAP Services features listed below can be used on reports:

- *Chapter 9, Dynamic Aggregation*
- *Chapter 6, Derived Metrics*
- *Chapter 7, View Filters*

The sections listed above describe how to use various OLAP Services features with reports. For information on how to create reports and use all the standard reporting features available in MicroStrategy, see the Basic Reporting Help and the Advanced Reporting Help.

Reporting and analyzing data with Intelligent Cubes

You can perform all of your reporting and data analysis within an Intelligent Cube, without executing any queries against the data warehouse. Once an Intelligent Cube is created and then published so its data is available, multiple reports can be created to connect to the Intelligent Cube (see *Sharing Intelligent Cubes, page 19*).

Reporting on an Intelligent Cube provides quick access to data, as the data has been pre-aggregated. This returns results much faster than querying the

Copyright © 2024 All Rights Reserved

data warehouse. Reporting on Intelligent Cubes also allows you to use all of the OLAP Services features, including derived elements, which allow you to group attribute elements in a report on the fly, to provide a new view of report data for analysis and formatting (see *Chapter 5, Derived Elements*).

Reports that connect to an Intelligent Cube are restricted to only the data available within the Intelligent Cube. This ensures that report results are returned quickly, and it also prevents full ROLAP analysis. Report filters, consolidations, and custom groups are not available for reports that connect to Intelligent Cubes.

While reporting on Intelligent Cubes, there are a few scenarios that can produce unexpected results. See *Troubleshooting reports connected to Intelligent Cubes, page 77* for more information.

This section discusses the features available for reports that connect to Intelligent Cubes, and how the reports utilize standard reporting features and OLAP Services features to execute reporting and analysis manipulations completely within the Intelligent Cube. This section includes information on the small differences in workflows and standards when using standard reporting features in these reports:

- *Creating reports that connect to Intelligent Cubes, page 61*
- *Creating Report Services documents that connect to Intelligent Cubes, page 64*
- *Creating Visual Insight dashboards that connect to Intelligent Cubes, page 65*
- *Analyzing data using standard OLAP Services features, page 72*
- *Run-time reporting with prompts, page 73*
- *Relational analysis with drilling, page 74*
- *Troubleshooting reports connected to Intelligent Cubes, page 77*

Copyright © 2024 All Rights Reserved

Creating reports that connect to Intelligent Cubes

To create a report that connects to an Intelligent Cube, you must choose an Intelligent Cube to connect to when creating the report. The procedures below describe how to create a report that connects to an Intelligent Cube.

Creating reports that connect to Intelligent Cubes, in Developer

Prerequisites

- An Intelligent Cube has been created and published.
- You need the Define Intelligent Cube Report privilege. This privilege is part of OLAP Services.

To create a report that connects to an Intelligent Cube, in MicroStrategy Developer

1. In Developer, log into a project.
2. Choose **File** > **New** > **Report**. The New Grid dialog box opens.
3. Select the **Intelligent Cubes** tab.
4. Browse to an Intelligent Cube, select it, and then click **OK**.

The Report Editor opens with all of the objects of the Intelligent Cube included in the Report Objects pane on the left. You can begin to create your report.

In Developer, you can also create a report that connects to an Intelligent Cube by right-clicking an Intelligent Cube and selecting **Create Report**.

Creating reports that connect to Intelligent Cubes, in Web

Prerequisites

Copyright © 2024 All Rights Reserved

- An Intelligent Cube has been created and published.
- You need the Web Define Intelligent Cube Report privilege. This privilege is part of OLAP Services.

To create a report that connects to an Intelligent Cube, in MicroStrategy Web

1. In MicroStrategy Web, log in to a project.
2. Click the MicroStrategy icon, and select **Create Report**. A page with report creation options is displayed.
3. Click **Intelligent Cube Report**. The Select Intelligent Cube dialog box opens.
4. Browse to an Intelligent Cube, select it, and then click **OK**.

The report opens with all of the objects of the Intelligent Cube included in the Report Objects pane on the left. You can begin to create your report.

Connecting to a different Intelligent Cube

You can modify the Intelligent Cube a report connects to for its data requirements. This allows you to switch to an Intelligent Cube that fits your reporting requirements without having to create a new report.

However, the Intelligent Cube you switch to for the report should have the same or at least similar data as was previously available in the report. If the report contained some objects that are not in the Intelligent Cube you switch to, this can cause the objects to be unavailable for the report or have other unintended functionality. A warning message is displayed for any mismatch in data between the report and the Intelligent Cube you want to link the report to.

Changing the Intelligent Cube that a report points to is possible in Developer only.

Copyright © 2024 All Rights Reserved

Prerequisites

- A report has been created that connects to a published Intelligent Cube.
- A second Intelligent Cube has been created and published that has the same or similar data as the report you want to link to it.
- You need the Define Intelligent Cube Report privilege. This privilege is part of OLAP Services.

To connect to a different Intelligent Cube

1. In Developer, browse to a report that is connected to an Intelligent Cube, right-click the report, and select **Edit**. The report opens in the Report Editor.
2. Choose **Data** > **Intelligent Cube Options** > **Point grid to Intelligent Cube**. The Select a cube dialog box opens.
3. Browse to the new Intelligent Cube you want to link the report to, select it, and then click **Open**.
4. If there are data mismatches between the report and the Intelligent Cube you are attempting to link it to, a warning message is displayed that lists the differences. Attributes on the report that are not contained in the new Intelligent Cube will have to be removed from the report to display valid report results. Missing metrics are converted into derived metrics and may be able to display correct data, but they should be removed if no data is returned.

   If the differences are acceptable, click **OK**.

   The report is updated to connect to the Intelligent Cube selected. You should remove any attributes and metrics if they no longer return correct results.

Copyright © 2024 All Rights Reserved

Creating Report Services documents that connect to Intelligent Cubes

You can use one or more Intelligent Cubes as datasets in Report Services documents. The steps to add Intelligent Cubes to Report Services documents are described below.

For detailed information on creating and using Report Services documents, refer to the Document Creation Help.

Prerequisites

1. An Intelligent Cube has been created and published.
2. You need the Define Intelligent Cube Report privilege. This privilege is part of OLAP Services.
3. If you want to use multiple Intelligent Cubes as datasets for your document, you must have the Execute Report that Uses Multiple Data Sources privilege.
4. If you want to use multiple Intelligent Cubes as datasets for your document, you must have the Import Table from Multiple Data Sources privilege.

To create a Report Services document that connects to Intelligent Cubes

1. In Web, click **New Document**. The Create Document page opens, showing document templates that you can use.
2. Click a template for the document. A new document opens, in Design mode.
3. In the Dataset Objects pane, click **Add Dataset**. The Select Dataset dialog box opens.

Copyright © 2024 All Rights Reserved

4. In the Select Dataset dialog box, navigate to the Intelligent Cube that you want to use as a dataset.
5. Select the Intelligent Cube, and click **OK**. The Intelligent Cube is added to the Dataset Objects pane.
6. To add another Intelligent Cube to the document, repeat the steps above.

For detailed information on using documents with multiple datasets, refer to the Document Creation Help.

Creating Visual Insight dashboards that connect to Intelligent Cubes

A Visual Insight (VI) dashboard is a customized, interactive display that you can use to explore your business data. For example, you can sort and rearrange data in an interactive grid, perform manipulations on the data to display only the information you are interested in, and display visual representations of the data in the VI dashboard to make the data easier to interpret.

The steps to create VI dashboards that connect to Intelligent Cubes are described below.

Prerequisites

- An Intelligent Cube has been created and published.
- You need the Define Intelligent Cube Report privilege. This privilege is part of OLAP Services.
- If you want to use multiple Intelligent Cubes as datasets for your dashboard, you must have the Execute Report that Uses Multiple Data Sources privilege.
- If you want to use multiple Intelligent Cubes as datasets for your dashboard, you must have the Import Table from Multiple Data Sources privilege.

Copyright © 2024 All Rights Reserved

To create a Visual Insight dashboard that connects to Intelligent Cubes

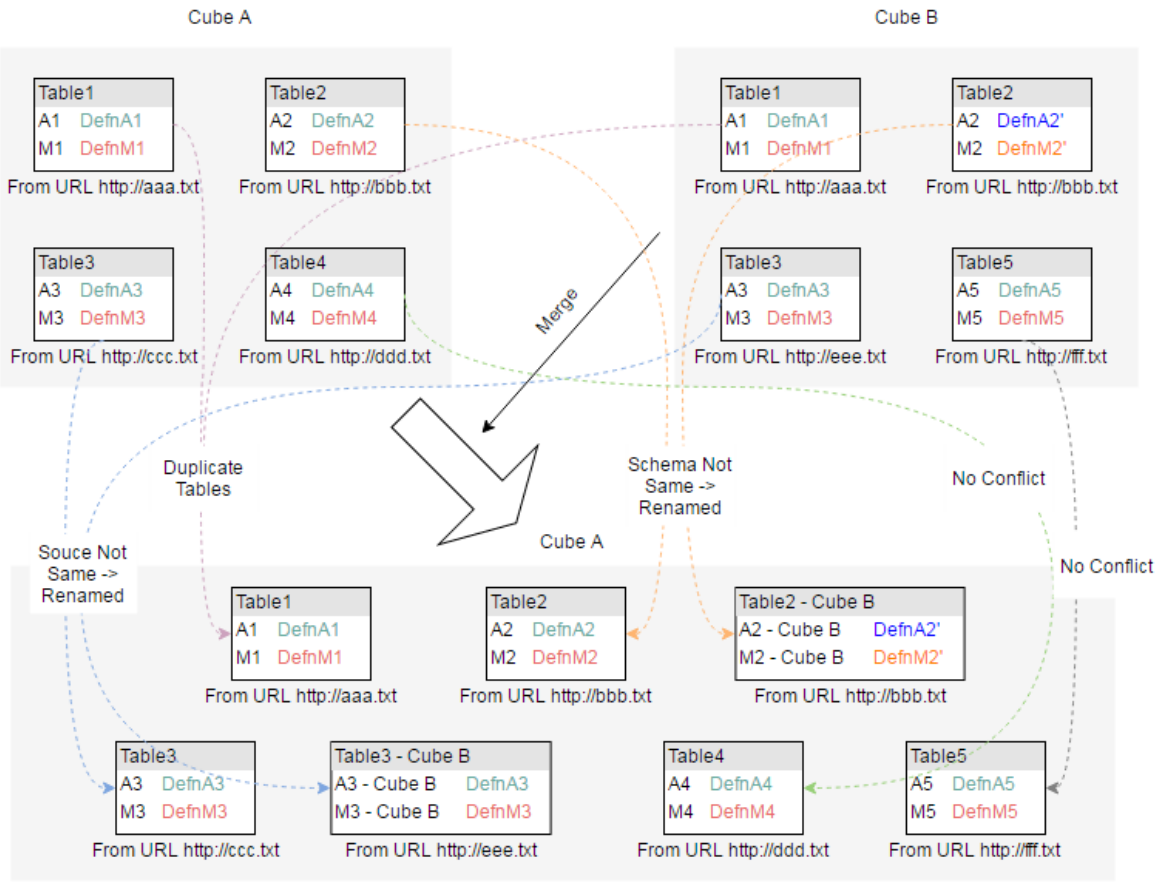
1. In Web, click **New Dashboard**. The Select Dataset dialog box opens.
2. In the Select Dataset dialog box, navigate to the Intelligent Cube that you want to use for the dashboard.
3. Select the Intelligent Cube, and click **Next**. A new VI dashboard opens, with the Intelligent Cube added to the Dataset Objects pane.
4. To add another Intelligent Cube to the dashboard's datasets, in the toolbar, click **Add Dataset**.

   For steps to create visualizations and analyze data in VI dashboards, see the MicroStrategy Web Help.

Merging Intelligent Cubes

MicroStrategy Web allows you to merge two or more Intelligent Cubes. This enables the development of datasets via data importing, instead of creating and modifying each new dataset from scratch. This option allows users to collaborate and develop an application schema over time by merging smaller, independent datasets into one Intelligent Cube. The attribute IDs for attributes in the cubes will remain intact so that any dashboards or reports that were previously linked to the cubes will still execute correctly. The following diagram shows how two cubes are merged.

Copyright © 2024 All Rights Reserved



For descriptions of scenarios involving duplicate data between cubes, see *Handling duplicate data when merging Intelligent Cubes, page 69*.

Prerequisites

- User must have full control access rights to the cubes being merged.
- None of the cubes being merged are exclusive.
- Cubes must support the same access type. (For example, In-Memory only cubes cannot merge with Direct Data Access only cubes.)

Merging cubes in MicroStrategy Web

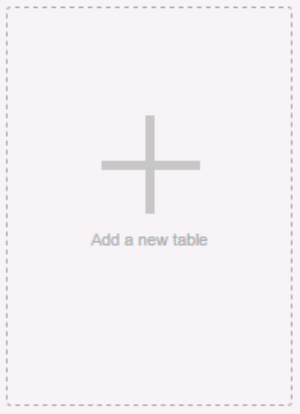
For simplicity, the following steps refer to Cube A and Cube B, where Cube B will be merged into Cube A.

Copyright © 2024 All Rights Reserved

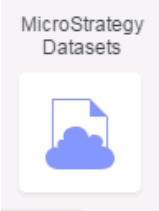
1. From the MicroStrategy Web Home page navigate to the Intelligent Cube (Cube A) you want to merge data into.

   ⚠ Cube A must be selected here to maintain its attribute IDs after the merge. If you select Cube B, any dashboards or reports linked to Cube A before the merge may not function correctly.

2. Right-click on the cube and choose **Edit**.
3. In the **Preview** window select **Add a new table**.



4. From the **Connect to Your Data** window, choose **MicroStrategy Datasets**.



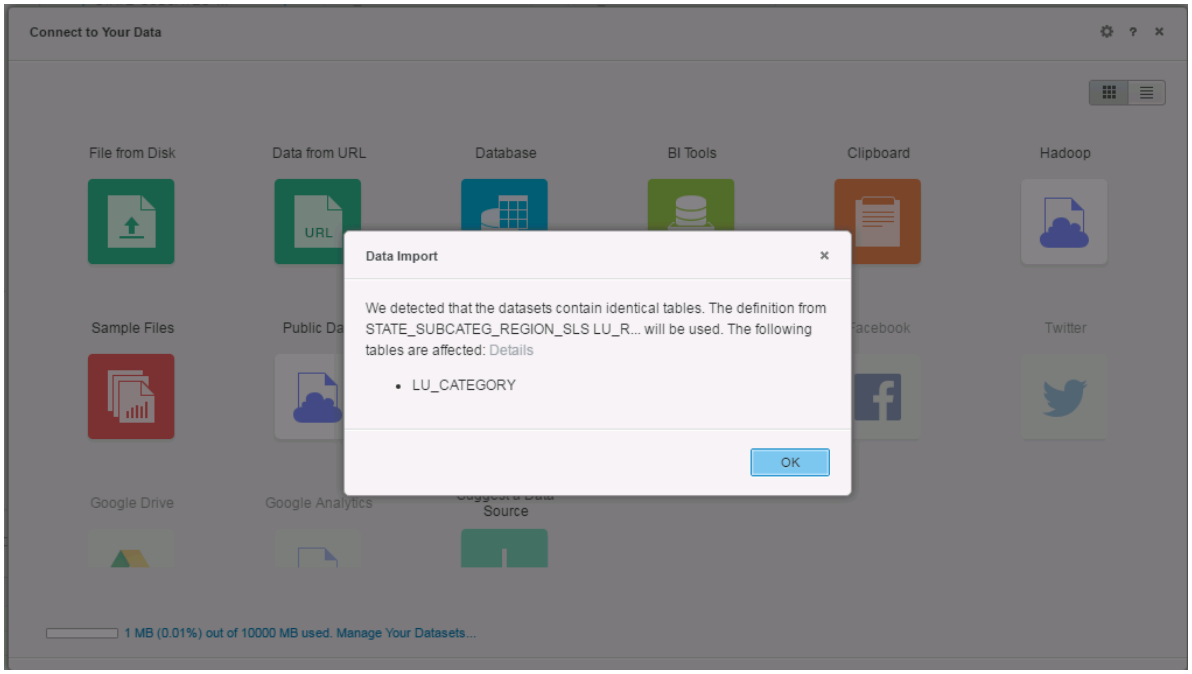
5. Navigate to the Intelligent Cube (Cube B) you want to add. Select the Cube and click **OK**.
   - To add another cube, repeat Steps 4 and 5.
   - To save the updated Cube A, click **Save Progress**.
6. If you are finished adding to Cube A, click **Update Dataset**.

Copyright © 2024 All Rights Reserved

7. In the **Data Access Mode** window choose how you access the new cube by selecting **Connect Live** or **Import as an In-memory Dataset**.
8. The Start your analysis window opens. You can choose from **Create Dashboard**, **Create Document**, or **Create Report**.

Handling duplicate data when merging Intelligent Cubes

In some cases two Intelligent Cubes being merged will contain tables, attributes, or metrics that are complete or partial duplicates. MicroStrategy will detect and resolve these anomalies during the merge to preserve the original attribute IDs in the newly merged cube. A warning message like the one pictured below will notify you of the duplicate data and how it will be resolved.



Schema comparisons for duplicate tables

- If the schema for the duplicate tables from Cube A and Cube B is the same, the table from Cube A will be picked and the duplicate from Cube B

Copyright © 2024 All Rights Reserved

will be skipped. The attribute IDs from Cube A will be retained.

- If the schema is different on both of the duplicate tables, Cube A will map as many attributes from the duplicate table as possible. These attributes will retain the IDs from Cube A and the remaining attributes will retain their IDs from Cube B.
- Columns are paired using column names
- Derived columns are skipped.
- Matched pairs of columns are compared using their mapping information, including object type, object name, data type, geo roles, etc.
- The schema will be treated as different if missing or redundant columns are found.

Source comparisons for duplicate tables

- If the data source table structure and schema for two identical tables is the same, they will be considered duplicates, and the table from Cube B will be skipped.
- If the data source for two identical tables is different, both tables will be merged. The name of the table from Cube B will be changed to "table name - Cube B".

Attribute and metric comparisons

- Attributes can be manually linked between two cubes before the merge.
- For attributes and metrics, if the name is the same and the data type are compatible, they will be linked automatically.
- If attributes or metrics have the same name, but are not data type compatible, the attribute or metric being merged will be renamed "attribute/metric name - Cube B".

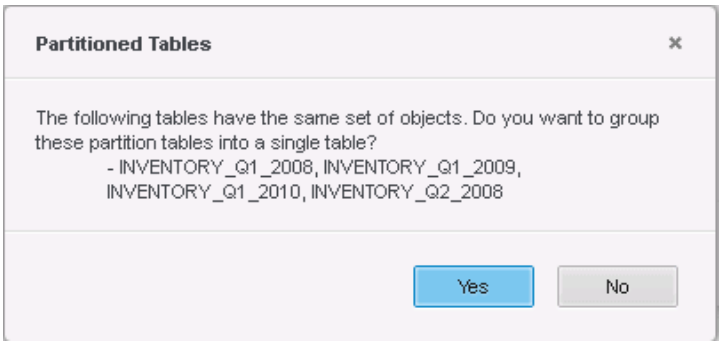
Copyright © 2024 All Rights Reserved

Multiform attributes follow the same rules as above. When multiform and single form attributes are compared, the data types for the ID of the multiform attribute and the name of the single form attribute are checked for compatibility.

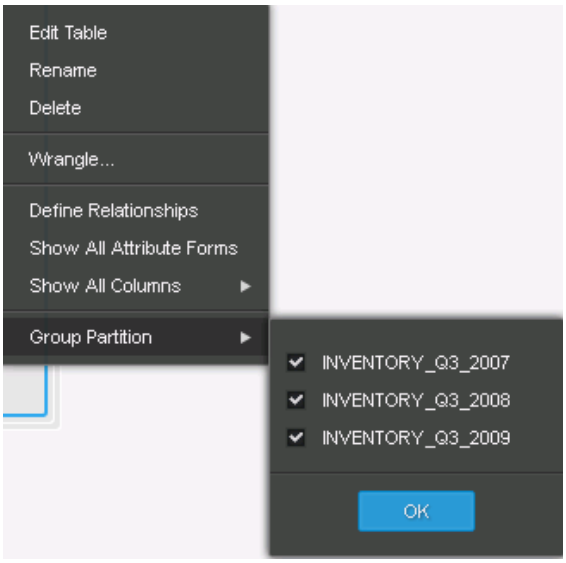
- GeoRole related attributes
  - If Cube A and Cube B have duplicate attributes, where one is missing the GeoRole, they will be linked and the GeoRole added to the attribute without one.
  - If Cube A and Cube B have duplicate attributes with different GeoRole values, the attribute from Cube B will be renamed.

Partitioned Tables

- If Cube A and Cube B both have two partition tables, they will merge successfully. To group the tables into a single table, select **Yes** in the reminder window. You can also group them together manually on the table menu.



Copyright © 2024 All Rights Reserved



- If all the sub tables are the same, the newly added duplicate database table will be removed.
- If the tables in Cube B are not a subset of the group tables in Cube A, but they can be grouped, you will see the pop up window for the partition option. Click **Yes** to remove duplicate tables. Click **No** to keep the tables separate. The tables can be grouped manually on the table menu.

Analyzing data using standard OLAP Services features

Reports that connect to Intelligent Cubes make use of OLAP Services features to provide a broad range of analysis capabilities. All OLAP Services analysis is executed against the Intelligent Cube. The following reporting features are described in this section:

- *Run-time reporting with prompts, page 73*
- *Relational analysis with drilling, page 74*
- *Troubleshooting reports connected to Intelligent Cubes, page 77*

For additional information on the different OLAP Services features available for reports that connect to Intelligent Cubes, see the following sections:

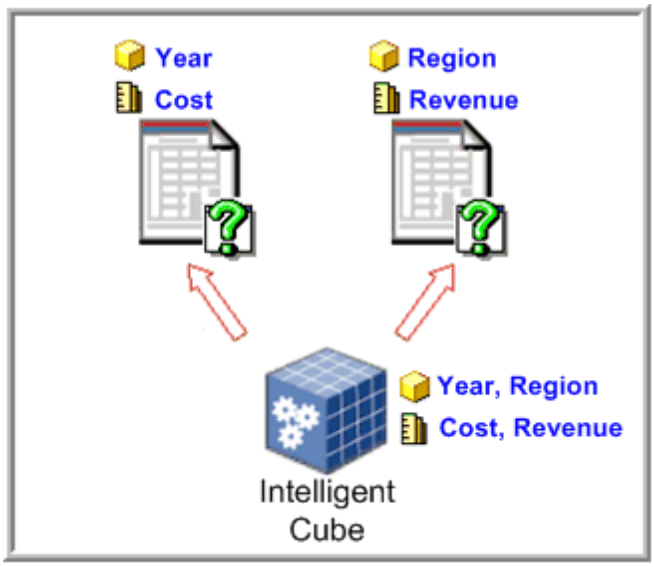
Copyright © 2024 All Rights Reserved

- *Chapter 5, Derived Elements*
- *Chapter 9, Dynamic Aggregation*
- *Chapter 6, Derived Metrics*
- *Chapter 7, View Filters*

Run-time reporting with prompts

In addition to including base attributes, metrics, and other objects to be displayed on your reports, you can also determine how much of the report to display at run time. Prompts allow users to choose which objects and filtering criteria to apply to a report during report execution.

Prompts serve the same purpose in any report, including reports that connect to Intelligent Cubes. However, instead of modifying SQL at report run time, prompts allow reports to select data within the Intelligent Cube, as illustrated below.



The image above shows standard run-time reporting with prompts, while using OLAP Services to execute against the Intelligent Cube rather than against the data warehouse. The performance of your business intelligence application is improved by reducing execution against your data warehouse and maintaining only a single Intelligent Cube for multiple prompted reports.

Copyright © 2024 All Rights Reserved

Prompts on reports that connect to Intelligent Cubes can only access data that is available within the Intelligent Cube. These restrictions are applied automatically when creating prompts. For example, the attributes Year and Region and the metrics Cost and Revenue are included in the Intelligent Cube shown in the image above. If you create an object prompt in your report that connects to this Intelligent Cube, then you can only create prompts based on one of Year, Region, Cost, and Revenue.

You cannot use prompts that include objects or data that are not part of the Intelligent Cube, or prompts that use hierarchies. If you try to use such prompts, an error message is displayed.

Prompts in reports that access Intelligent Cubes can use the complete ROLAP schema of a project. However, if a prompt retrieves data from outside the Intelligent Cube, re-execution against the data warehouse is necessary.

Relational analysis with drilling

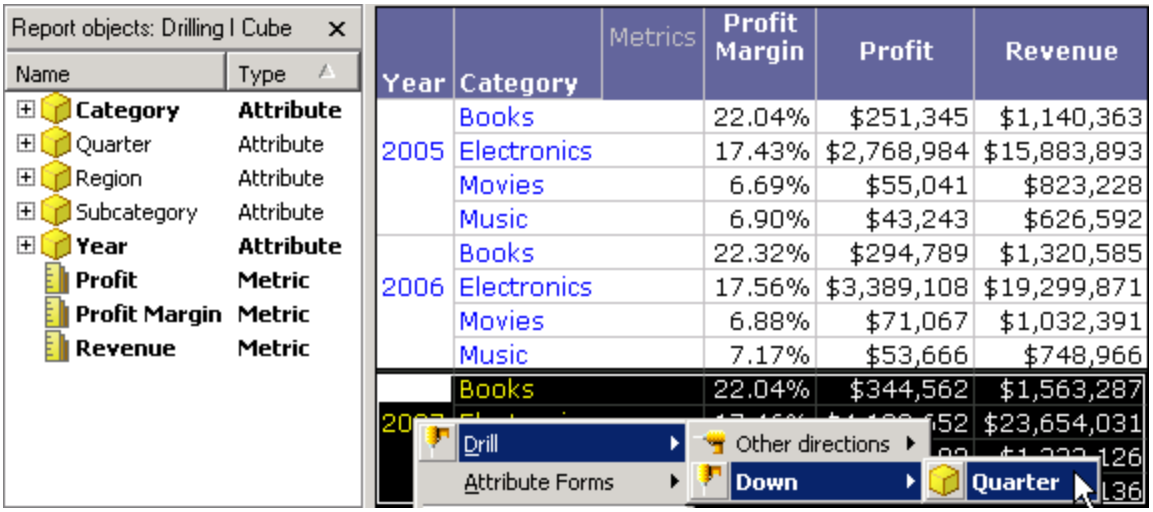
As with any MicroStrategy report, you can drill on reports that connect to Intelligent Cubes to analyze data at different logical levels.

A report connected to an Intelligent Cube can drill within the data available in the Intelligent Cube it is connected to. This means that you can drill from an attribute on the report grid to an attribute that is not on the report grid, but available in the Report Objects pane. If the attribute is not available in the Report Objects pane, it is not an available drilling option by default. However, Intelligent Cubes can be defined to allow drilling outside of an Intelligent Cube to the full relational data warehouse.

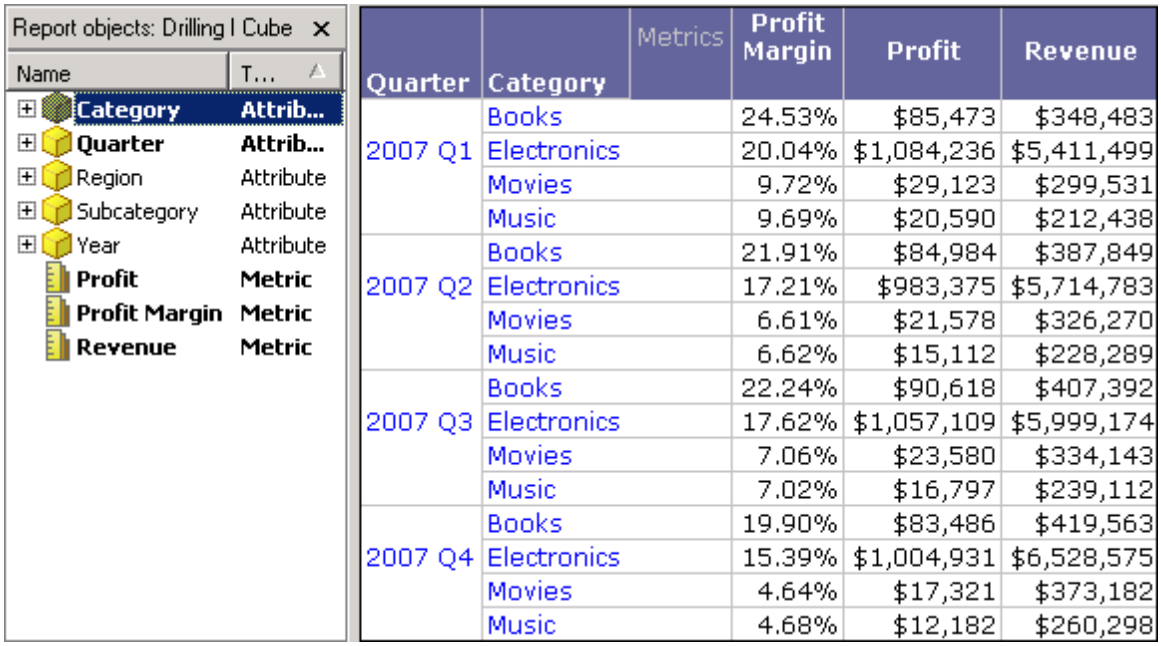
For example, your report includes the attribute Year. After analyzing data at the Year level, you want to analyze data for each quarter. You can drill down from Year to the attribute Quarter to view and analyze data at the new logical level. This drilling action is performed within an Intelligent Cube. Following this example scenario, you want to drill from year 2007 to quarters

Copyright © 2024 All Rights Reserved

for that year. You have a report connected to an Intelligent Cube that is defined as shown below:



Notice that Quarter is not on the report, but it is included in the Report Objects pane on the left as it is a part of the Intelligent Cube that the report is connected to. As shown in the report above, you right-click the 2007 attribute element for Year and drill down to Quarter. The drilled-to report is shown below.

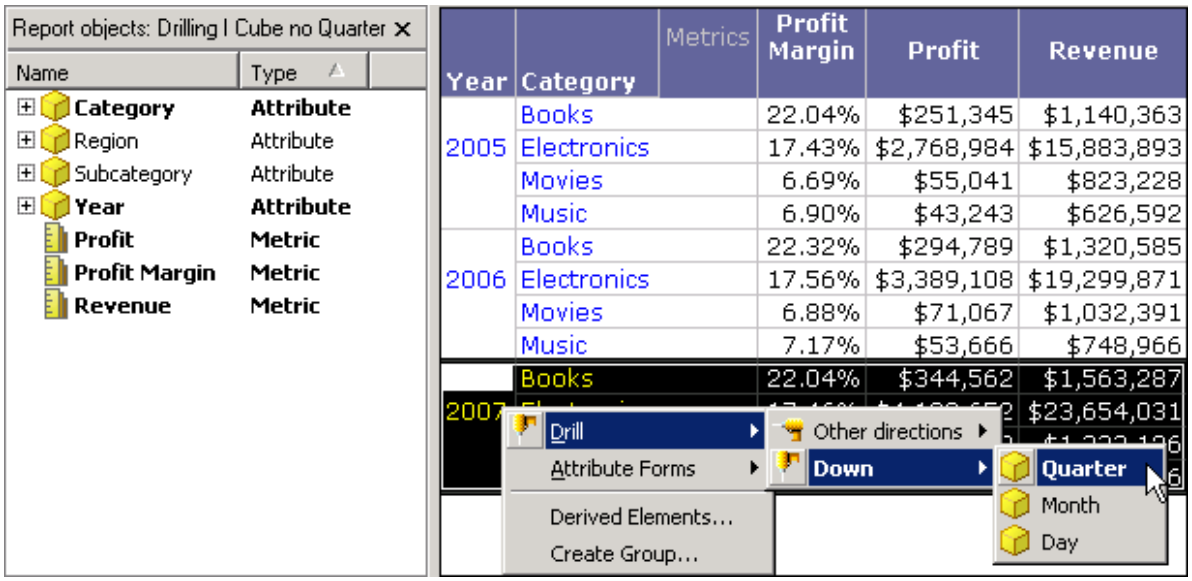


Copyright © 2024 All Rights Reserved

This drilled-to report is executed within and connected to the same Intelligent Cube as the original report. This is verifiable by looking at the Report Objects pane, which shows that the report objects are being returned from the same Intelligent Cube (named Drilling I Cube) as the original report. This provides relational analysis without having to execute the report against the data warehouse.

In the scenario above, drilling is performed within the Intelligent Cube, which is achievable through any report connected to an Intelligent Cube. However, if the Intelligent Cube is defined to allow drilling outside it (see *Enabling ROLAP drilling for reports accessing Intelligent Cubes, page 36*), you can also drill to any object not included in the Intelligent Cube. While drilling outside of an Intelligent Cube requires execution against the data warehouse, it provides access to the full ROLAP schema of the project outside of the Intelligent Cube.

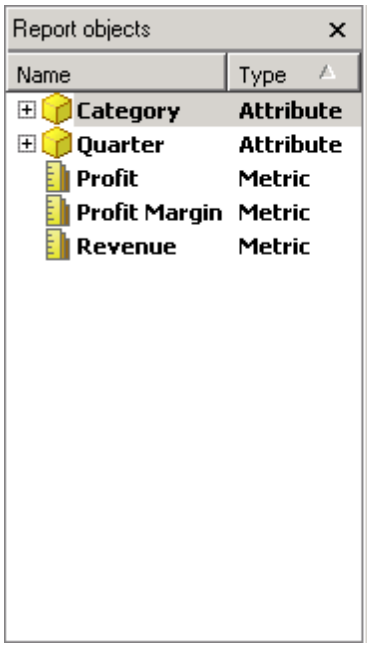
In the next example, the same scenario of drilling from Year to Quarter is used, except that the Intelligent Cube does not contain the Quarter attribute. As shown in the report below, you right-click the 2007 attribute element for Year and drill down to Quarter.



Notice in the report shown above that all the attributes in the Time hierarchy are available drilling options even though they are not all included in the

Copyright © 2024 All Rights Reserved

Intelligent Cube. These attributes are available drilling options because the Intelligent Cube is defined to enable drilling outside of the Intelligent Cube. As shown in the report above, you right-click the 2007 attribute element for Year and drill down to Quarter. The drilled-to report is shown below.



Quarter	Category	Metrics	Profit Margin	Profit	Revenue
2007 Q1	Books		24.53%	$85,473	$348,483
	Electronics		20.04%	$1,084,237	$5,411,499
	Movies		9.72%	$29,123	$299,531
	Music		9.69%	$20,590	$212,438
2007 Q2	Books		21.91%	$84,984	$387,849
	Electronics		17.21%	$983,376	$5,714,783
	Movies		6.61%	$21,578	$326,270
	Music		6.62%	$15,112	$228,289
2007 Q3	Books		22.24%	$90,618	$407,392
	Electronics		17.62%	$1,057,109	$5,999,174
	Movies		7.06%	$23,580	$334,143
	Music		7.02%	$16,797	$239,112
2007 Q4	Books		19.90%	$83,486	$419,563
	Electronics		15.39%	$1,004,932	$6,528,576
	Movies		4.64%	$17,321	$373,182
	Music		4.68%	$12,182	$260,298

This drilled-to report is executed against the data warehouse, and it allows you to access data outside of the Intelligent Cube for further relational analysis. Notice also that all report objects that were not on the report grid are now removed from the Report Objects pane, because this new, drilled-to report is not connected to the Intelligent Cube.

You should consider the execution time requirements for a report before drilling outside of an Intelligent Cube.

For basics on how to drill on data, see the Basic Reporting Help. For information on creating drill maps, which are used to enable drilling techniques, see the *Drill Maps* section in the Advanced Reporting Help.

Troubleshooting reports connected to Intelligent Cubes

While reporting on Intelligent Cubes, a few scenarios can produce unexpected results. To troubleshoot these issues, review the following

Copyright © 2024 All Rights Reserved

sections.

No data is returned for the report

If no data is returned for a report that is connected to an Intelligent Cube, this can be caused by the following scenarios:

- The view filter for the report is too restrictive (see *Chapter 7, View Filters*). Modify or remove the view filter to attempt to return data for the report. If removing the view filter does not allow the report to return data, the cause may be due to your security filter, as described below.
- Your security filter does not allow you to see the data available on the Intelligent Cube.

Resolving security filter resolution

If security filters are used to maintain data access security, this can cause data to not be displayed for a report connected to an Intelligent Cube.

While this may be due to normal data access security, the data available on the Intelligent Cube may prevent you from viewing all possible data in a data source (see *Security filter resolution for reports connected to Intelligent Cubes, page 22*.) To verify whether there is additional data that is not being returned, you can create or view a report that connects directly to a data source rather than an Intelligent Cube. You can also contact the designers of the Intelligent Cube to review whether additional data can be added to the Intelligent Cube so that it supports your report.

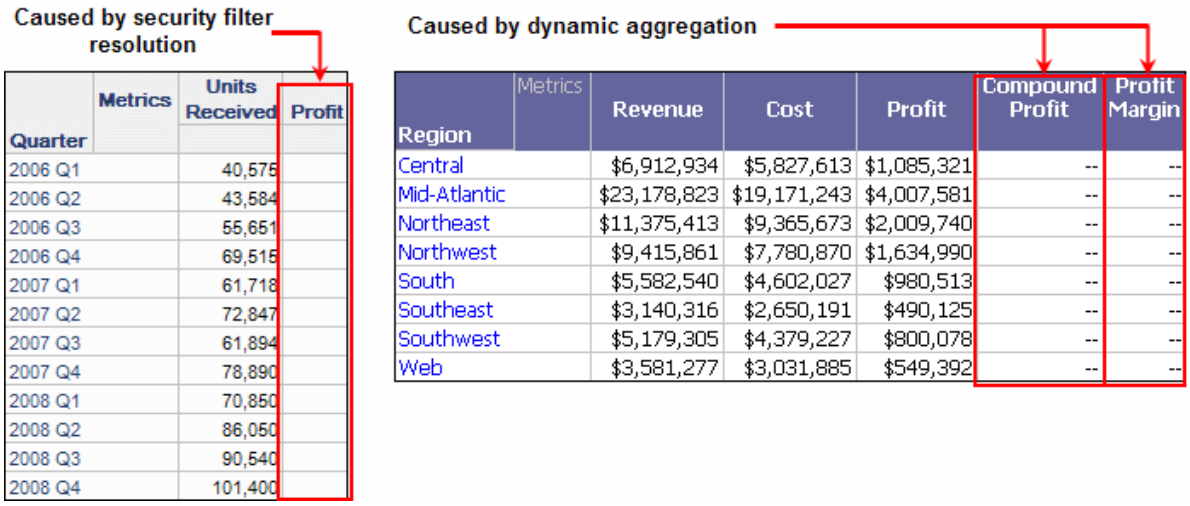
No data is returned for a metric or metrics on the report

If no data is returned for a metric or metrics on a report connected to an Intelligent Cube, this can be caused by either dynamic sourcing or your security filter.

When this is caused by your security filter resolution, no data is displayed for the metric or metrics. However, when this is caused by dynamic

Copyright © 2024 All Rights Reserved

aggregation, null values are displayed for the metric rather than no information at all. The image below shows the difference between security filter resolution and dynamic aggregation as the cause for metrics not displaying any data.



Quarter	Metrics	Units Received	Profit
2006 Q1		40,575	
2006 Q2		43,584	
2006 Q3		55,651	
2006 Q4		69,515	
2007 Q1		61,718	
2007 Q2		72,847	
2007 Q3		61,894	
2007 Q4		78,890	
2008 Q1		70,850	
2008 Q2		86,050	
2008 Q3		90,540	
2008 Q4		101,400	

Region	Metrics	Revenue	Cost	Profit	Compound Profit	Profit Margin
Central		$6,912,934	$5,827,613	$1,085,321	--	--
Mid-Atlantic		$23,178,823	$19,171,243	$4,007,581	--	--
Northeast		$11,375,413	$9,365,673	$2,009,740	--	--
Northwest		$9,415,861	$7,780,870	$1,634,990	--	--
South		$5,582,540	$4,602,027	$980,513	--	--
Southeast		$3,140,316	$2,650,191	$490,125	--	--
Southwest		$5,179,305	$4,379,227	$800,078	--	--
Web		$3,581,277	$3,031,885	$549,392	--	--

By default, null values are represented by dashes (--) on reports. For information on changing the display of null values, see *Changing the display of null values, page 340*.

If dynamic aggregation is the cause for null values being displayed, resolutions for this issue are described in *Metrics that are not dynamically aggregated by default, page 330*.

The report fails due to the unavailability of Intelligent Cubes

An unavailable Intelligent Cube can cause your reports connected to this Intelligent Cube to fail. This is because the report depends on the Intelligent Cube to provide the data for the report.

The scenarios listed below cause reports to return error messages when Intelligent Cubes are unavailable:

- The Intelligent Cube is not published. This scenario can occur when an Intelligent Cube is removed to:

Copyright © 2024 All Rights Reserved

- Update it with new data available in the data warehouse.
- Release its resources for more frequently used Intelligent Cubes.
- Provide space for a new Intelligent Cube because Intelligent Cube storage limits have been exceeded.

Contact your administrator to determine why an Intelligent Cube is no longer published and whether there are plans to publish it again.

- The Intelligent Cube is currently offline. In this scenario, the Intelligent Cube is published but has been made unavailable for reporting. Contact the administrator of Intelligent Cubes to determine why the Intelligent Cube has been made unavailable, as well as if and when the Intelligent Cube will be made available again.
- The Intelligent Cube is being published, but the publishing process is not complete. This scenario can occur when an Intelligent Cube is first being published. If you wait for the Intelligent Cube to publish, you should then be able to run your report without experiencing this error.

Reporting on Intelligent Cubes with dynamic sourcing

You can manually connect a report to an Intelligent Cube to perform all reporting and analysis within the shared in-memory copy of data, as described in *Reporting and analyzing data with Intelligent Cubes, page 59*. Even if you don't manually connect your reports to an Intelligent Cube, the reports you create can also access Intelligent Cube data rather than querying the data warehouse automatically. Enabling this automatic link between reports and Intelligent Cubes is referred to as dynamic sourcing.

Dynamic sourcing extends the accessibility of Intelligent Cubes by allowing regular reports to access published Intelligent Cubes, as long as the Intelligent Cubes can satisfy the requirements of the report. For a detailed explanation of dynamic sourcing and procedures to perform the project-wide configurations necessary to enable and support dynamic sourcing, see *Chapter 9, Dynamic Aggregation*.

Copyright © 2024 All Rights Reserved

While reporting on Intelligent Cubes, there are a few scenarios that can produce unexpected results. To troubleshoot these issues, see *Troubleshooting reports connected to Intelligent Cubes, page 77*

Supporting report execution through dynamic sourcing

When creating reports, you can enable and increase your chances that your report utilizes dynamic sourcing by following the recommendations below.

First, you must enable dynamic sourcing for your report. This can add some overhead to the execution of your report to check whether there are any Intelligent Cubes that meet the data requirements of your report. This overhead usually has no effect on performance. If an Intelligent Cube can be used, the report results can be returned quickly from the in-memory copy of data rather than querying the data warehouse. To enable dynamic sourcing for reports, see *Enabling or disabling dynamic sourcing for reports, page 273*.

The simpler your report is, the more likely it is that an Intelligent Cube satisfies its data requirements. This does not mean that you should create reports with minimal data simply to take advantage of dynamic sourcing. However, the considerations listed below can help you create a report that meets your reporting requirements and is well-suited for dynamic sourcing:

- Avoid the use of features that prevent the use of dynamic sourcing, which are described in *Features that prevent the use of dynamic sourcing, page 261*.
- Only add objects that are relevant to the analysis required for the report. When creating a report, you should determine whether certain attributes, metrics, advanced filtering techniques, and so on are required for the report or are extraneous to the analysis requirements of a report.

Copyright © 2024 All Rights Reserved

Overview: Large, in-memory datasets in MicroStrategy

MicroStrategy allows you to load large datasets into your Intelligence Server's memory as Intelligent Cubes, and divide the Intelligent Cubes into multiple segments. These segments, called partitions, are processed simultaneously, distributed across the processor cores of your Intelligence Server.

By storing your data in your Intelligence Server's memory and processing the data using all the server's processor cores, you can analyze large and complex datasets with very fast response times.

You divide your dataset based on an attribute, called a partition attribute. When you choose a partition attribute, all the tables that contain that attribute are partitioned.

Benefits of using partitioned datasets

The benefits of storing your data in your Intelligence Server's memory and processing the data using all the server's processor cores partitioning your dataset include:

- Improving the response time for your reports, documents, and dashboards.
- Reducing the load on your data warehouse, because your dataset is stored in Intelligence Server's memory instead of the data warehouse.
- Analyzing large volumes of data in real time, since your datasets can contain up to 2 billion rows of data for every partition on the Intelligence Server.
- Using the standard features of the MicroStrategy platform, such derived metrics, custom groups, consolidations, and so on, on your dataset.

Copyright © 2024 All Rights Reserved

Creating an application that uses a partitioned dataset

To create an application based on a partitioned dataset, you need to perform the following high-level tasks:

- Evaluate the needs of your application and whether your application is suitable for partitioned datasets.
- Prepare your dataset and determine the number of partitions that you require.
- Ensure that you have the necessary hardware to support the partitions of your dataset.
- Design a logical data model of your dataset, similar to designing a project schema for MicroStrategy. The logical model is a representation of your dataset as experienced by your users. A logical data model depicts the flow and structure of your data, providing a way of organizing data so that it can be analyzed from different perspectives.

  For steps and considerations for creating a logical data model, see the Project Design Help.
- Select the attribute that will divide your dataset (the partition attribute). All the tables that contain the partition attribute are split into partitions. The tables are analyzed in parallel by the processor cores of your Intelligence Server machine.
- Create the dataset using Web.
- Use simple base metrics when you create the dataset, and then create derived metrics for more complex metrics calculations after you import the dataset. For steps to create derived metrics, see the MicroStrategy Web Help.

Copyright © 2024 All Rights Reserved

- Design your documents and dashboards using MicroStrategy Web. For steps to design documents and dashboards, see the MicroStrategy Web Help.
- Maintain and update your dataset.

Evaluating your application

Features that make an application suitable for a partitioned dataset

An application that meets at least one of the following criteria may be a suitable candidate for a partitioned dataset:

- Your documents and dashboards are centrally managed and allow your users to analyze data from different perspectives.
- All the data that you need for the application can be loaded in a single dataset.
- All tables that have more than two billion rows of data can be split based on the same attribute.
- Your base Key Performance Indicators (KPIs) are calculated using basic aggregation functions such as Sum, Average, Minimum, Maximum, Count, and so on. The KPIs can be calculated individually for each partition and then combined.

  Once you have created your dataset, you can create derived metrics that use any of the standard MicroStrategy functions.

  For a full list of the most efficient functions to use in partitioned datasets, see *Appendix A, Efficient Functions for Partitioned Datasets*.
- The dataset for your application needs to be incrementally updated on a schedule.
- The dataset for your application is less than or equal to two terabytes (TB) in size.

Copyright © 2024 All Rights Reserved

- Your data is structured, and an Extract, Transform, Load (ETL) process has been performed on it.

Features that make an application unsuitable for partitioned datasets

An application that meets at least one of the following criteria may not be suitable for a partitioned dataset:

- Your dataset needs to support self-service analyses, where your users can create their own reports, documents, or dashboards.
- All of the data for the application cannot be loaded in a single dataset.
- Your application allows users to add or update data in your warehouse by using Transaction Services.
- All tables that have more than two billion rows of data cannot be partitioned based on the same attribute.
- The calculations for your KPIs require the entire dataset. For example, KPIs that use functions such as First, Last, Standard Deviation, OLAP functions, and so on require the entire dataset.

  For a full list of the most efficient functions to use in partitioned datasets, see *Appendix A, Efficient Functions for Partitioned Datasets*.
- Your data is unstructured, and include data sources other than RDBMS or flat files.
- Your dataset needs to be updated in real time.

Preparing your MicroStrategy environment and dataset

Before you create your partitioned dataset, you need to ensure that your Intelligence Server machine meets the hardware requirements for loading a partitioned dataset into its memory. Additionally, you must select an appropriate partition attribute that maximizes the performance of your reports, documents, and dashboards.

Copyright © 2024 All Rights Reserved

Requirements for your Intelligence Server machine

To determine whether your Intelligence Server can support your in-memory dataset, note the following:

- The number of partitions you divide your dataset into must be less than or equal to the number of processor cores on your Intelligence Server machine.
- The maximum size of a partition is 2 billion rows.
- Ensure that your Intelligence Server machine has enough memory to handle the data.

Requirements for the metrics on your partitioned dataset

Ensure that the metrics that you need to use can be calculated independently for each partition, then combined.

Requirements for the partition attribute

The partition attribute is the attribute that will be used to divide your dataset. The partition attribute must meet the following requirements:

- The data type of the partition attribute must be one of the following:
  - Integer
  - BigDecimal
  - Text
  - Date
- The partition attribute should be present on as many fact tables as possible, especially your largest fact tables. This requirement ensures that the calculations that you need to perform take full advantage of MicroStrategy's parallel processing capabilities.

Copyright © 2024 All Rights Reserved

- The partition attribute is not used to filter your dataset. This ensures that when users analyze data in your application, the maximum number of partitions are involved in the calculations, which leads to faster response times.
- The partition attribute is on any tables that are larger than two billion rows.

Creating a partitioned dataset

To create a partitioned dataset, import tables from your data warehouse using Web, as described in the steps below.

To create a partitioned dataset

1. In Web, log in to the project to import your dataset into.
2. On the home page, click **Add External Data**. The Connect to your Data dialog box opens.
3. Click **Database**. The Select Import Options dialog box opens.
4. Select **Pick Tables** and click **Next**. The Import from Tables dialog box opens.
5. From the Database Connections panel on the left, select the database connection that contains the data to import. A list of the database tables in the selected database is displayed in the Available Tables panel.
6. If your database supports multiple namespaces, you can display only the database tables for a specific namespace. To do this, select the namespace from the drop-down list. To search for a namespace, type the namespace in the field. The choices in the drop-down list are filtered as you type.
7. To expand the name of a table and view a list of columns in the table, click the **Expand** icon next to the table name. Each column in the table

Copyright © 2024 All Rights Reserved

is displayed, along with its data type. You can filter the list of database tables by typing the name of a table in the search field.

8. To partition your data, click **Prepare Data**. The Preview dialog box opens, displaying the attributes and metrics for your dataset, and the data tables that they are based on.
9. If you want to create search indexes to improve the performance of search-based selectors, you must define relationships between attributes to specify how the attributes are connected.

   In a relationship between Year and Quarter, Year is the parent attribute and Quarter is the child.

   Hover the cursor over the name of a table, click the **Menu** > **Define Relationships**. Click **Add a New Relation**, select the **Parent Attribute**, **Child Attribute**, and **Relationship**. For detailed steps to define relationships between attributes, see the MicroStrategy Web Help.
10. Click **All Objects View**. The All Objects View dialog box opens, showing all the attributes and metrics in your dataset.
11. From the **Partition Attribute** drop-down list, select the attribute to use to divide your dataset. For help in evaluating your dataset for an appropriate attribute, see *Requirements for the partition attribute, page 86*.
12. In the **Number of Partitions** field, type the number of partitions to split the dataset into. The number of partitions should be less than or equal to the number of CPU cores on your Intelligence Server machine.
13. To improve the performance of search-based selectors, in the **Search Index** column, enable the check box for the appropriate attributes. You must have defined attribute relationships, as described above.

Copyright © 2024 All Rights Reserved

Turning on search indexes takes up additional space in your Intelligence Server's memory.

14. Click **OK**.

Editing and updating your dataset

Once you have imported your dataset, you can make modifications to it. For example, you can manually designate a data column as an attribute or a metric. You can also add or remove data from your dataset, depending on changes in your data source.

- For steps to edit your dataset, see the MicroStrategy Web Help. Some examples of editing your dataset are listed below.
  - Add more data from the same data source or a different one.
  - Remove a table from the dataset.
  - Change the data type of a column.
  - Assign a geo role to a data column, to generate additional geographical data and to allow for easier integration with map-based visualizations.
  - Designate a data column as an attribute or a metric.
- For steps to update the data in your dataset, such as when new data is available in your data source, see *Updating the data in your dataset, page 89*.

Updating the data in your dataset

You can specify the schedule for data updates for your dataset. This ensures that when new or updated data is available from the data source, your dataset and application are updated.

Copyright © 2024 All Rights Reserved

To update your dataset using a schedule

1. In Web, navigate to the Intelligent Cube for your dataset.
2. Right-click the dataset, and choose **Schedule**. The Schedule dialog box opens for your dataset, showing the tables in your dataset.
3. In the Data Source column, select the check boxes for the tables to update.
4. In the Refresh Policy column, choose one of the following options:
   - To replace all the data in your dataset with updated data from your data source, select **Replace existing data**. Select this option if your dataset contains a rolling set of data—for example, if your dataset always contains data for the last six months.
   - To update the existing data in your dataset with any updated data in your data source, select **Update existing data**. Select this option if the data in your data source is updated regularly, but your dataset requires a specific subset of the data.
   - To update the existing data and add any new data, select **Update existing data and add new data**. Select this option if your dataset is updated often, for example, a sales dataset that records new transactions and tracks updates to older transactions as applicable.
   - To add new data from your data source to your dataset, select **Add new data**. Data that is already in the dataset is not altered. Select this option if old data is not updated after it is saved to your data source.

     For complex updates, or to only update your dataset with a subset of the data in your data source, you can create a refresh filter. For example, you can update the tables in your dataset at different intervals, depending on when the data in your data source is updated.

Copyright © 2024 All Rights Reserved

5. Click **Show Advanced Update Options**. The Set Refresh Filter and Alternate Source columns are displayed.
6. Click **Set Refresh Filter**. The Select a filter dialog box opens.
7. Click **Add Condition**. A new attribute qualification is started, displaying a list of the attributes in your dataset.
8. In the **Based On** list, select the attribute to filter by.
9. Define the qualification by doing one of the following:

- To create a qualification based on attribute elements in a list (an attribute element is an individual characteristic of an attribute, such as USA or France for the Country attribute):
    a. Under Select, do one of the following:
        - To include data only for elements that you select, click **In**. For example, you can include data for the Books and Movies categories only.
        - To include data for all elements except those that you select, click **Not in**. For example, you can include data for all product categories except Books and Movies.
    b. Type the names of the attribute elements to select.
- To create a qualification based on attribute form values:
    a. Under Qualification, select the attribute form to base the qualification on. For example, you can filter based on the attribute element's ID form, one of its description forms, or the DATE form if the attribute is time-based.
    b. From the list of operators to the right, select a comparison operator, such as **Greater Than** or **Less Than**. Do one of the following:
        - To compare the attribute form to a specific value, type the value in the field.

Copyright © 2024 All Rights Reserved

    - To compare the attribute form to another attribute form, select the attribute that contains the second form. Select the second attribute form from the list.

- Click the **Apply** icon to create the qualification. The qualification is created and displayed in the Select a filter dialog box.
- Click **Save**.

  If the updated data for your tables is in a different data source than the original, you can configure a different data source for each table. Ensure that the table in the new data source contains columns with identical names to the columns in your original data source. If you do not need to specify a difference source, click **Finish**.
- If the Set Refresh Filter and Alternate Source columns are not displayed on the Schedule dialog box, click **Show Advanced Update Options**.
- In the Alternate Source column for the table to update, click **Set Source**. The Select Alternate Source dialog box opens.
- From the Database Sources panel on the left, select the database connection that contains the data to import. A list of the database tables in the selected database is displayed in the Available Tables panel.
- Type the name of the table in the search field. The list of tables is updated automatically as you type.
- Click and drag the name of the table from the Available Tables panel to the panel on the right.
- Click **Finish**.

Copyright © 2024 All Rights Reserved

Derived Elements

Copyright © 2024 All Rights Reserved

Defining Attribute Elements On The Fly

A derived element is a grouping of attribute elements on a report. Derived element groups are defined by a list, filter, or calculation. These groups provide a new view of report data for analysis and formatting purposes.

For example, the reports below contain Region, Category, and Profit. The report on the left does not display any derived elements. The report on the right displays derived elements defined using groups of Region attribute elements:

- East Coast: This derived element combines the data for the Mid-Atlantic, Northeast, and Southeast regions.
- West Coast: This derived element combines the data for the Northwest and Southwest regions.
- Central and South: This derived element combines the data for the Central and South regions.
- Web: This derived element displays the data for the Web attribute element, which is not included in any of the derived elements listed above.

Copyright © 2024 All Rights Reserved

Region	Category (Metrics)	Profit
Central	Books	$55,495
	Electronics	$935,123
	Movies	$16,347
	Music	$78,357
Mid-Atlantic	Books	$46,461
	Electronics	$3,933,032
	Movies	$14,351
	Music	$13,737
Northeast	Books	$455,964
	Electronics	$1,500,514
	Movies	$27,367
	Music	$25,895
Northwest	Books	$19,734
	Electronics	$1,604,110
	Movies	$5,639
	Music	$5,508
South	Books	$216,663
	Electronics	$739,064
	Movies	$12,905
	Music	$11,882
Southeast	Books	$25,609
	Electronics	$419,847
	Movies	$37,704
	Music	$6,965
Southwest	Books	$41,576
	Electronics	$686,421
	Movies	$60,787
	Music	$11,294
Web	Books	$29,195
	Electronics	$469,635
	Movies	$42,611
	Music	$7,952

Region	Category (Metrics)	Profit
East Coast	Books	$528,034
	Electronics	$5,853,393
	Movies	$79,423
	Music	$46,597
West Coast	Books	$61,310
	Electronics	$2,290,530
	Movies	$66,426
	Music	$16,803
Central and South	Books	$272,158
	Electronics	$1,674,186
	Movies	$29,251
	Music	$90,238
Web	Books	$29,195
	Electronics	$469,635
	Movies	$42,611
	Music	$7,952

You can do more than just simple combinations of attribute elements with derived elements. For example, after you have defined the East Coast derived element, you can determine the East Coast region's percent contribution to profit, as shown in the last row of the report below.

Copyright © 2024 All Rights Reserved

Region	Category	Metrics	Profit
East Coast	Books		$528,034
	Electronics		$5,853,393
	Movies		$79,423
	Music		$46,597
West Coast	Books		$61,310
	Electronics		$2,290,530
	Movies		$66,426
	Music		$16,803
Central and South	Books		$272,158
	Electronics		$1,674,186
	Movies		$29,251
	Music		$90,238
Web	Books		$29,195
	Electronics		$469,635
	Movies		$42,611
	Music		$7,952
East Coast's % Contribution	Books		59.28%
	Electronics		56.90%
	Movies		36.48%
	Music		28.84%

This demonstrates only a fraction of the analysis and formatting capabilities of derived elements. With derived elements you can also create custom sort orders for attribute elements, use aggregation functions such as Average to combine attribute elements, and perform other analysis and formatting tasks.

Derived elements are evaluated on the report dataset without regenerating or re-executing SQL.

The following sections cover derived element concepts, functionality, and procedures:

- *Types of derived elements, page 97*
- *Creating derived elements, page 106*
- *Defining derived element functionality and formatting, page 151*
- *Interaction with other reporting features, page 164*

Copyright © 2024 All Rights Reserved

Types of derived elements

The different types of derived elements provide a wide range of reporting and analysis features. The following sections describe each type of derived element and how they can be used for reporting and analysis requirements:

- *Group derived element, page 97*
- *Filter derived element, page 99*
- *Calculation derived element, page 102*
- *All Other derived element, page 105*

Group derived element

A Group derived element is a combination of attribute elements into a single derived element. All attribute element data is added together to form the metric values for the new Group derived element.

For example, you have a report with Region, Category, and Profit displayed on a report. The report shown below includes the following derived elements defined using groups of Region attribute elements:

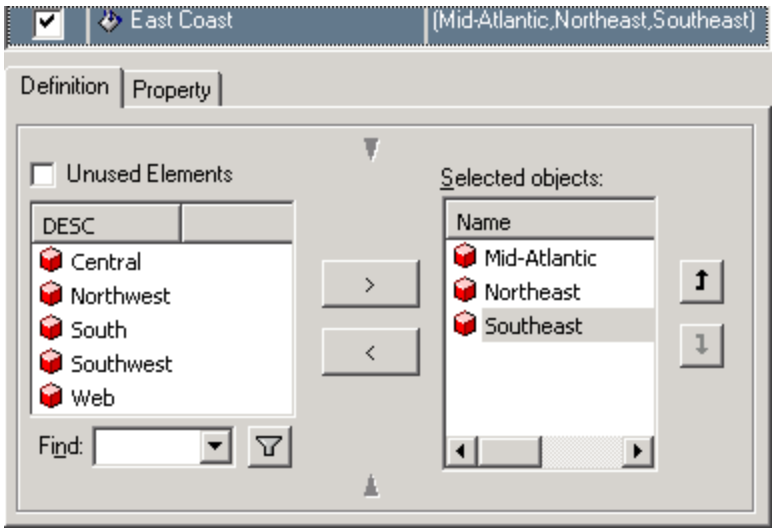
- East Coast: Groups the Mid-Atlantic, Northeast, and Southeast attribute elements.
- West Coast: Groups the Northwest and Southwest attribute elements.
- Central and South: Groups the Central and South attribute elements.

Copyright © 2024 All Rights Reserved

Region	Category	Metrics	Profit
East Coast	Books		$528,034
	Electronics		$5,853,393
	Movies		$79,423
	Music		$46,597
West Coast	Books		$61,310
	Electronics		$2,290,530
	Movies		$66,426
	Music		$16,803
Central and South	Books		$272,158
	Electronics		$1,674,186
	Movies		$29,251
	Music		$90,238
Web	Books		$29,195
	Electronics		$469,635
	Movies		$42,611
	Music		$7,952

A procedure to create the report shown above is provided in *Grouping attribute elements to create a derived element, page 130*.

Group derived elements are created by selecting attribute elements to include in each derived element. The image below shows how the East Coast Group derived element is created in the Derived Elements Editor:



Group derived elements can only combine attribute elements, they cannot combine other derived elements. If you want to create a derived element

Copyright © 2024 All Rights Reserved

that is a combination of other derived elements, you must use a Calculation derived element (see *Calculation derived element, page 102*).

You can quickly create Group derived elements using right-click options (see *Creating quick groups, page 109*), or you can use the Derived Elements Editor to access the full functionality of derived elements (see *Using the Derived Elements Editor, page 124*).

You can also use this type of derived element to display the attribute elements in a different order. This enables you to do more advanced attribute element sorting than simple ascending or descending sorts. For information on using Group derived elements to sort the display of attribute elements on a report, see *Creating quick sorts, page 120*.

Filter derived element

A Filter derived element uses a filter qualification to determine the combination of attribute elements for a derived element.

For example, in a report with Region and Category attributes and a Profit metric, you can filter the regions on the report into various geographical groups based on the region names.

The report shown below includes the following derived elements defined using filters for Region attribute elements:

- Southern Regions: Returns attribute elements whose name begins with South.
- Northern Regions: Returns attribute elements whose name begins with North.

Copyright © 2024 All Rights Reserved

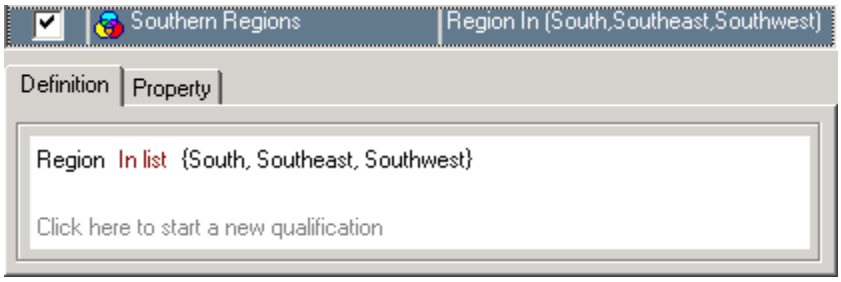
Region	Category	Metrics	Profit
Southern Regions	Books		$283,848
	Electronics		$1,845,331
	Movies		$111,396
	Music		$30,141
Northern Regions	Books		$475,697
	Electronics		$3,104,624
	Movies		$33,006
	Music		$31,404
Central	Books		$55,495
	Electronics		$935,123
	Movies		$16,347
	Music		$78,357
Mid-Atlantic	Books		$46,461
	Electronics		$3,933,032
	Movies		$14,351
	Music		$13,737
Web	Books		$29,195
	Electronics		$469,635
	Movies		$42,611
	Music		$7,952

A procedure to create the report shown above is provided in *Filtering attribute elements to create a derived element, page 135*.

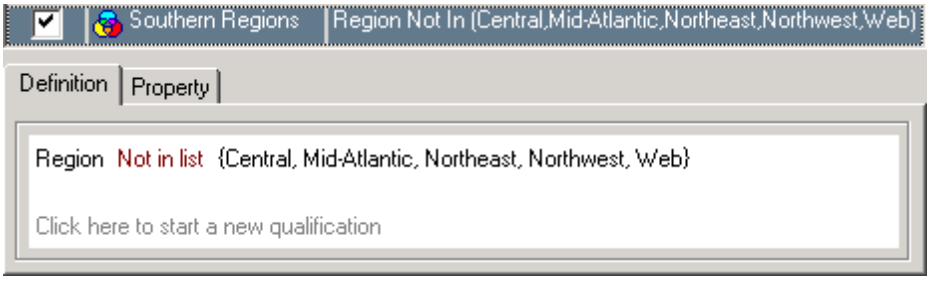
Filter derived elements are created by filtering attribute elements to include in each derived element. There are two methods to create Filter derived elements:

- Create a Filter derived element using a filter qualification on a list of attribute elements. This includes using the In list and Not in List operators.
  - In list: A filter qualification using In list returns data for all the attribute elements you select. An In list filter qualification that returns all the southern regions is shown below.

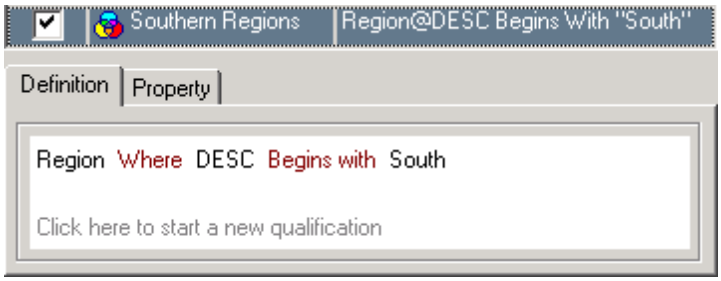
Copyright © 2024 All Rights Reserved



- Not in List: A filter qualification using Not in List returns data for all the attribute elements you do not select, for a given attribute. A Not in List filter qualification that returns all the southern regions is shown below.



- Create a Filter derived element using a filter qualification on attribute forms. This enables you to use various logical and mathematical operators to create filter qualifications on attribute forms to return data. An attribute form qualification using a Begins with operator that returns all southern regions is shown below.



You can use any of the following operators in attribute form qualifications, which are described in detail in *Appendix A, Logical and Mathematical Operators for Filtering* in the Advanced Reporting Help:

Copyright © 2024 All Rights Reserved

Operator Type	Operator
Comparison Operators Comparison operators compare values. The values can be numbers, text strings, or expressions.	Exactly
	Different from
	Greater than
	Less than
	Greater than or equal to
	Less than or equal to
	Between
Pattern Operators Pattern operators allow text strings to be compared. Pattern operators are case-sensitive.	Like
	Not Like
	Contains
	Does not contain
	Begins with
	Does not begin with
	Ends with
	Does not end with

To create Filter derived elements, you must use the Derived Elements Editor (see *Using the Derived Elements Editor, page 124*).

Calculation derived element

A Calculation derived element uses operators and functions to combine attribute elements and derived elements into calculations that define a single derived element.

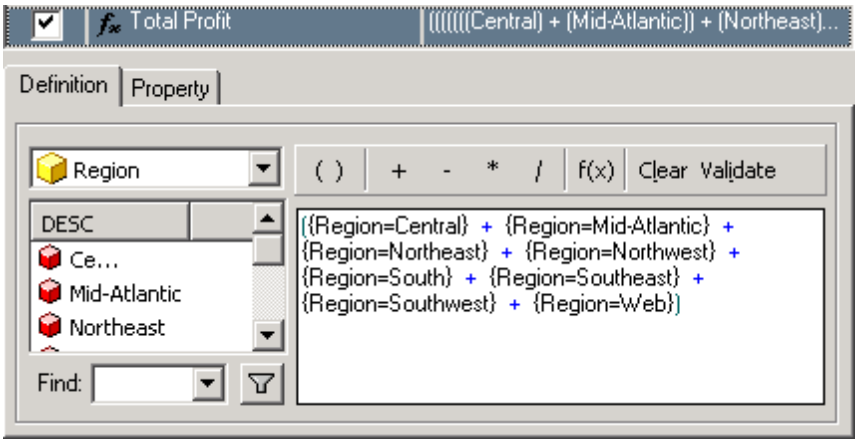
Copyright © 2024 All Rights Reserved

For example, in a report with Region and Category attributes and a Profit metric, you can combine the regions on the report into various groups for profit analysis, as shown below.

Region	Category / Metrics	Profit
Total Profit	Books	$890,696
	Electronics	$10,287,744
	Movies	$217,710
	Music	$161,590
Average Profit	Books	$111,337
	Electronics	$1,285,968
	Movies	$27,214
	Music	$20,199
Greatest Regional Profit	Books	$455,964
	Electronics	$3,933,032
	Movies	$60,787
	Music	$78,357
Greatest Regional Profit % Contribution	Books	51.19%
	Electronics	38.23%
	Movies	27.92%
	Music	48.49%

A procedure to create the report shown above is provided in *Using calculations to create derived elements, page 140*.

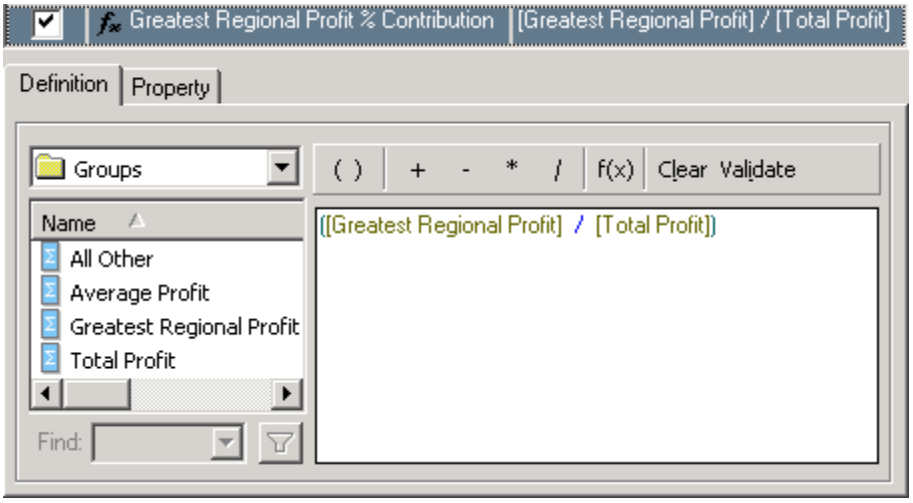
Calculation derived elements are created by defining expressions with valid combinations of operators, functions, attribute elements, and derived elements. An example of a valid expression is shown below.



Copyright © 2024 All Rights Reserved

You can include the following when you create a Calculation derived element expression:

- Attribute elements: You can include attribute elements in your expression by selecting the attribute from the drop-down list, selecting attribute elements, and dragging and dropping them into the expression area.
- Derived elements: You can include other derived elements in your expression by selecting derived elements from the Groups drop-down list, and dragging and dropping them into the expression area. The example below shows a Calculation derived element created by performing a division of two other derived elements.



- Operators: You can include ( ), +, -, *, and /, which are all available on the toolbar.
- Functions: You can include Average, Greatest, and Least by clicking **f(x)** on the toolbar and completing the Insert Function Wizard. For information on these functions, see *Creating quick calculations, page 113*. For steps to use the Insert Function Wizard, click Help in the wizard.
- Clear: You can clear the expression to start creating a new expression.
- Validate: You can check your expression to see if its syntax is valid. Any errors in syntax are highlighted in red.

Copyright © 2024 All Rights Reserved

To create Calculation derived elements, you can either quickly create derived elements with right-click options (see *Creating quick calculations, page 113*), or you can use the Derived Elements Editor to access the full functionality of derived elements (see *Using the Derived Elements Editor, page 124*).

All Other derived element

The All Other derived element collects all attribute elements that are not included in other derived elements, and displays them on the report as individual attribute elements by default. This derived element is created automatically when the first derived element is created for an attribute. An All Other derived element can only be deleted by deleting all derived elements.

Because it is created automatically, the All Other derived element is created and implemented without having to take note of it at all. For example, recall the report created with a Filter derived element (see *Filter derived element, page 99*) that included Southern Regions and Northern Regions derived elements as shown below.

Region	Category (Metrics)	Profit
Southern Regions	Books	$283,848
	Electronics	$1,845,331
	Movies	$111,396
	Music	$30,141
Northern Regions	Books	$475,697
	Electronics	$3,104,624
	Movies	$33,006
	Music	$31,404
Central	Books	$55,495
	Electronics	$935,123
	Movies	$16,347
	Music	$78,357
Mid-Atlantic	Books	$46,461
	Electronics	$3,933,032
	Movies	$14,351
	Music	$13,737
Web	Books	$29,195
	Electronics	$469,635
	Movies	$42,611
	Music	$7,952

Copyright © 2024 All Rights Reserved

Central, Mid-Atlantic, and Web are all attribute elements that are not included in any derived elements. The All Other derived element gathers these remaining attribute elements, and displays them on a report as individual attribute elements.

You can also include all of the attribute elements that are part of the All Other derived element as one consolidated element on a report (see *Displaying derived elements or their attribute elements, page 154*).

While this is the most common way the All Other derived element is used, you can define attribute elements included in other derived elements to also be included as part of the All Other derived element. For information on including attribute elements in the All Other derived element, see *Displaying derived elements and their attribute elements simultaneously, page 157*.

Creating derived elements

Derived elements require static report results so that they can be evaluated without regenerating or re-executing SQL. Derived elements can be created in the following types of reports and documents:

- Standard reports
- Intelligent Cube reports: An Intelligent Cube report is a report that is connected to and retrieves its data from an Intelligent Cube.
- Grid/Graphs in Report Services documents: A Grid/Graph is a control placed in a document that displays information in the same way a MicroStrategy report does. In other words, a Grid/Graph is a report that has been added to a section of a Report Services document.

If a report containing derived elements is included as a dataset of a Report Services document, derived elements are displayed for any associated attributes added to the details or grouping sections. However, you cannot create or modify derived elements for attributes added to the grouping or details sections.

Copyright © 2024 All Rights Reserved

You can create derived elements with the following methods, which are all described in this section:

Depending on the type of report you are creating a derived element on, you can use either MicroStrategy Developer or Web. To see which interfaces you can use, refer to the tables under each of the following methods.

- *Quickly creating groups, calculations, and sorts, page 107*: While reviewing the data on a report or document, you can quickly group attribute elements into derived elements for further analysis of your data.
- *Using the Derived Elements Editor, page 124*: Using the Derived Elements Editor provides the full set of derived elements functionality when creating derived elements.
- *Creating and using stand-alone derived elements, page 146*: You can create stand-alone derived elements that can be used in multiple reports and Grid/Graphs. You use the Derived Elements Editor to create stand-alone derived elements, but you do not have to create them from within a report or Grid/Graph.

Quickly creating groups, calculations, and sorts

While reviewing the results on a report or document, you can quickly group attribute elements into derived elements for further analysis of your data.

You can quickly create derived elements based on attribute elements and other derived elements in reports and Grid/Graphs using right-click options. You can create the following types of quick group derived elements:

Review the table following the list of quick group derived elements for a list of when you can use these quick group options to create derived elements.

- *Creating quick groups, page 109*: Creates a simple group of attribute elements.

Copyright © 2024 All Rights Reserved

- *Creating quick calculations, page 113*: Creates a calculation on the attribute elements or derived elements (or a combination of both).
- *Creating quick sorts, page 120*: Creates a derived element that sorts the attribute elements on the report or document in any order you want. This option is only available if no derived elements are defined for the attribute on the report or document.

These quick group options to create derived elements are quick and easy ways to create derived elements. However, creating derived elements with these quick group techniques is only available with the configurations listed in the table below:

Report Object Type	View/Mode	MicroStrategy Developer/Web	Derived Element Quick Groups		
			Quick Group	Quick Sort	Quick Calculation
Standard report	Grid View	Web	Yes	Yes	Yes
	Grid and Graph View	Web	Yes	Yes	Yes
	Graph View	Web	No	No	No
Intelligent Cube report	Grid View	Developer and Web	Yes	Yes	Yes
	Grid and Graph View	Developer and Web	Yes	Yes	Yes
	Graph View	Developer and Web	No	No	No

Copyright © 2024 All Rights Reserved

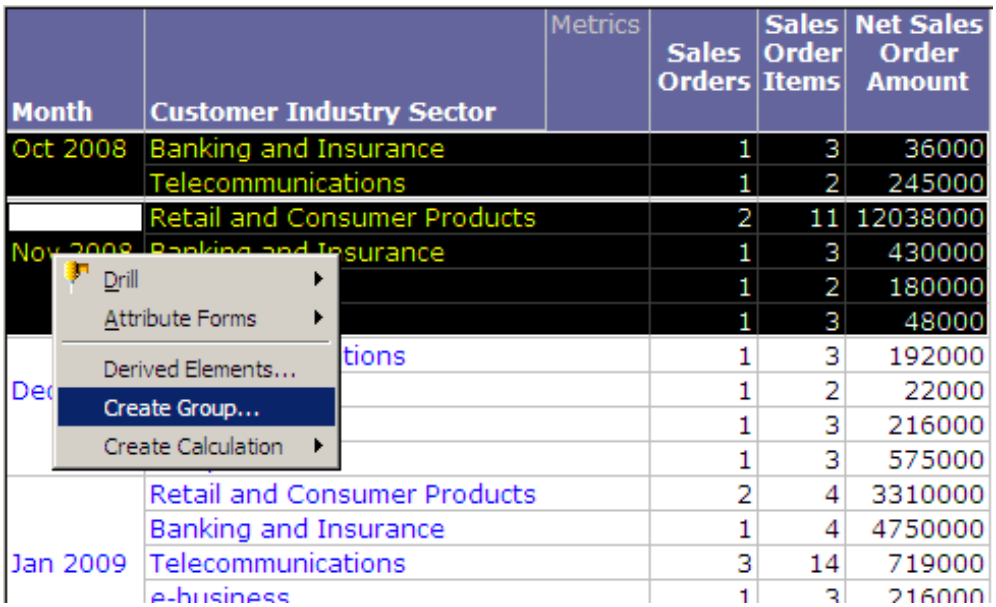
Report Object Type	View/Mode	MicroStrategy Developer/Web	Derived Element Quick Groups		
			Quick Group	Quick Sort	Quick Calculation
Grid/Graph in a Report Services document	Design View	Developer	No	Yes	No
	Design Mode	Web	No	No	No
	Interactive Mode (Grid/Graph must be displayed as a Grid or a Grid and Graph)	Web	Yes	No	Yes
	Editable Mode (Grid/Graph must be displayed as a Grid or a Grid and Graph)	Web	Yes	No	Yes

Creating quick groups

Using right-click options, you can group attribute elements into a single derived element. The derived element created by this action aggregates and displays all the data for the attribute elements selected, into one set of data. For information on Group derived elements, see *Group derived element, page 97*.

For example, you have a report that displays the number of sales orders, sales order items, and sales order amounts for customer industry sectors. Your report displays this information over a number of months. To get a more summarized view of the data, you decide to group months into seasons. You group October and November 2008 into a single derived element called Fall 08, by selecting the attribute elements, right-clicking the selection, and selecting **Create Group**. This process is shown below.

Copyright © 2024 All Rights Reserved



You can then type a name for the group, such as Fall 08, and then click **OK**. A derived element is created for the Fall 08 group, which displays and aggregates the data for October and November 2008.

You can also group attributes elements to create derived elements for the winter, spring and summer months. A report with a derived element for each season group is shown below.

Copyright © 2024 All Rights Reserved

Month	Customer Industry Sector	Metrics	Sales Orders	Sales Order Items	Net Sales Order Amount
Fall 08	Retail and Consumer Products		2	11	12038000
	Banking and Insurance		2	6	466000
	Telecommunications		1	2	245000
	e-business		1	2	180000
	Others		1	3	48000
Winter 08/09	Retail and Consumer Products		3	7	5820000
	Banking and Insurance		2	6	4773000
	Telecommunications		4	17	911000
	e-business		2	5	238000
	Others		3	9	3142400
	Computers		6	16	3261500
Spring 09	Retail and Consumer Products		4	11	3087000
	Banking and Insurance		4	10	4308500
	Telecommunications		2	5	4942000
	Others		1	4	650000
	Computers		2	11	1509000
Summer 09	Retail and Consumer Products		5	20	14785000
	Banking and Insurance		5	12	2386000
	Telecommunications		6	14	2938000
	e-business		1	5	90000
	Others		3	10	2626400
	Computers		2	3	425000

As shown in the report above, data for each customer industry sector is aggregated for the three months of each group. For example, Retail and Consumer Products had one sales order in June 2009, three sales orders in July 2009, and one sales order in August 2009. This data is aggregated to a total of five sales orders for the Summer 09 time period.

Creating a derived element on Intelligent Cube reports with a quick group

The steps below show you how to create a quick group on a report connected to an Intelligent Cube.

Prerequisites

- The report on which you create the derived element is connected to an active Intelligent Cube.

Copyright © 2024 All Rights Reserved

- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To create a derived element with a quick group

1. In MicroStrategy Developer or Web, log in to a project that contains Intelligent Cube reports.
2. Navigate to and run the Intelligent Cube report. View the report in either Grid View or Grid and Graph View.
3. In the grid display of the report, hold down the **CTRL** key and select multiple attribute elements within the same attribute.

   ⚠ Do not select derived elements for the attribute, as you cannot create quick groups based on derived elements. To group derived elements, you must use the Derived Elements Editor.

4. Right-click your selection and select **Group**. The Create Group dialog box opens.
5. Type a name for the derived element, and click **OK**.

The group is created as a derived element and displayed on the report. You can modify the derived element using the Derived Elements Editor (see *Using the Derived Elements Editor, page 124*).

Creating a derived element on standard reports with a quick group

You can create quick groups on standard reports in View Mode in MicroStrategy Web. For steps to create quick groups on standard reports in MicroStrategy Web, see the *MicroStrategy Web Help*.

Copyright © 2024 All Rights Reserved

Creating a derived element on Grid/Graphs with a quick group

You can create quick groups on Grid/Graphs in documents in Interactive or Editable Mode in MicroStrategy Web. For steps to create quick groups on Grid/Graphs in documents in MicroStrategy Web, see the *MicroStrategy Web Help*.

Creating quick calculations

Using right-click options, you can group attribute elements and derived elements into a single derived element using the following quick calculations. Quick calculations enable you to view various types of data and analysis on the same report.

- If you select exactly two attribute elements, you can choose from all of the calculations listed below.
- If you select more than two attribute elements, Subtract and Divide are not available calculations as they can only accept two operands. If you want to create a subtraction or division including more than two attribute elements, you must use the Derived Elements Editor.

- The Greatest and Least calculations are best used when your report has only one metric. These calculations operate on each metric individually, so if your report has more than one metric, the values for the Greatest derived element will not necessarily correspond to the same attribute element.

- **Sum**: Performs an addition of two or more attribute elements, derived elements, or a combination of both.
- **Subtract**: Performs a subtraction of two attribute elements, derived elements, or a combination of both. The element that is at the higher position on the report is the first operand of the subtraction, and the element at the lower position is the second operand of the subtraction.

Copyright © 2024 All Rights Reserved

- **Average**: Performs an average of two or more attribute elements, derived elements, or a combination of both.
- **Divide**: Performs a division of two attribute elements, derived elements, or a combination of both. The element that is at the higher position is the first operand of the division, and the element at the lower position is the second operand of the division.
- **Greatest**: Calculates and displays the greatest value of each metric for two or more attribute elements, derived elements, or a combination of both.

  The two tables below show an example of a creating a greatest calculation on the books and electronics categories.

Category	Revenue	Profit
Books	$1,000	$500
Electronics	$2,000	$300
…	…	…

Category	Revenue	Profit
Greatest of Books and Electronics	$2,000	$500
…	…	…

- **Least**: Calculates and displays the least value of each metric for two or more attribute elements, derived elements, or a combination of both.

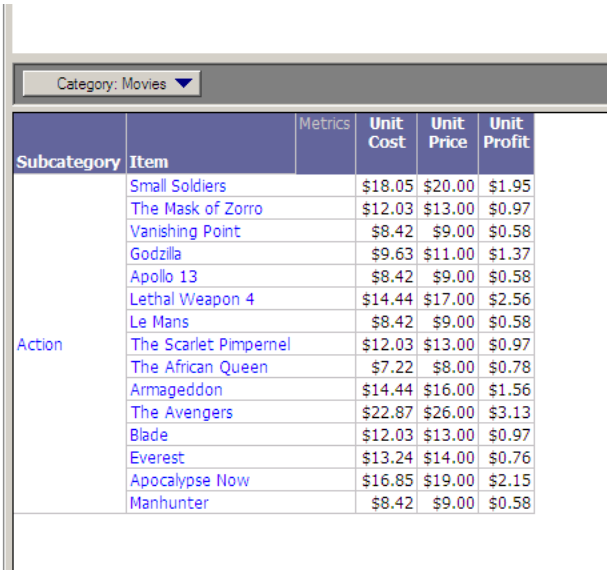
  Using the same example used to illustrate the Greatest calculation listed above, the table below shows an example of a creating a least calculation on the books and electronics categories.

Copyright © 2024 All Rights Reserved

Category	Revenue	Profit
Least of Books and Electronics	$1,000	$300
…	…	…

A quick calculation is created as a Calculation derived element. For information on Calculation derived elements, see *Calculation derived element, page 102*.

To illustrate quick calculations, consider the following example. You have a report that displays the unit price, cost, and profit for all items sold in the Action movies subcategory, as shown below.



Category: Movies

Subcategory	Item	Metrics	Unit Cost	Unit Price	Unit Profit
Action	Small Soldiers		$18.05	$20.00	$1.95
	The Mask of Zorro		$12.03	$13.00	$0.97
	Vanishing Point		$8.42	$9.00	$0.58
	Godzilla		$9.63	$11.00	$1.37
	Apollo 13		$8.42	$9.00	$0.58
	Lethal Weapon 4		$14.44	$17.00	$2.56
	Le Mans		$8.42	$9.00	$0.58
	The Scarlet Pimpernel		$12.03	$13.00	$0.97
	The African Queen		$7.22	$8.00	$0.78
	Armageddon		$14.44	$16.00	$1.56
	The Avengers		$22.87	$26.00	$3.13
	Blade		$12.03	$13.00	$0.97
	Everest		$13.24	$14.00	$0.76
	Apocalypse Now		$16.85	$19.00	$2.15
	Manhunter		$8.42	$9.00	$0.58

You decide to do further analysis based on the unit cost of the various items listed. To provide this analysis you begin to put the items into various groups that perform an average of their data. You create the first group by selecting Vanishing Point, Godzilla, Apollo 13, Le Mans, The African Queen, and Manhunter, then right-clicking the selection, pointing to **Create Calculation**, and selecting **Average**. This process is shown below.

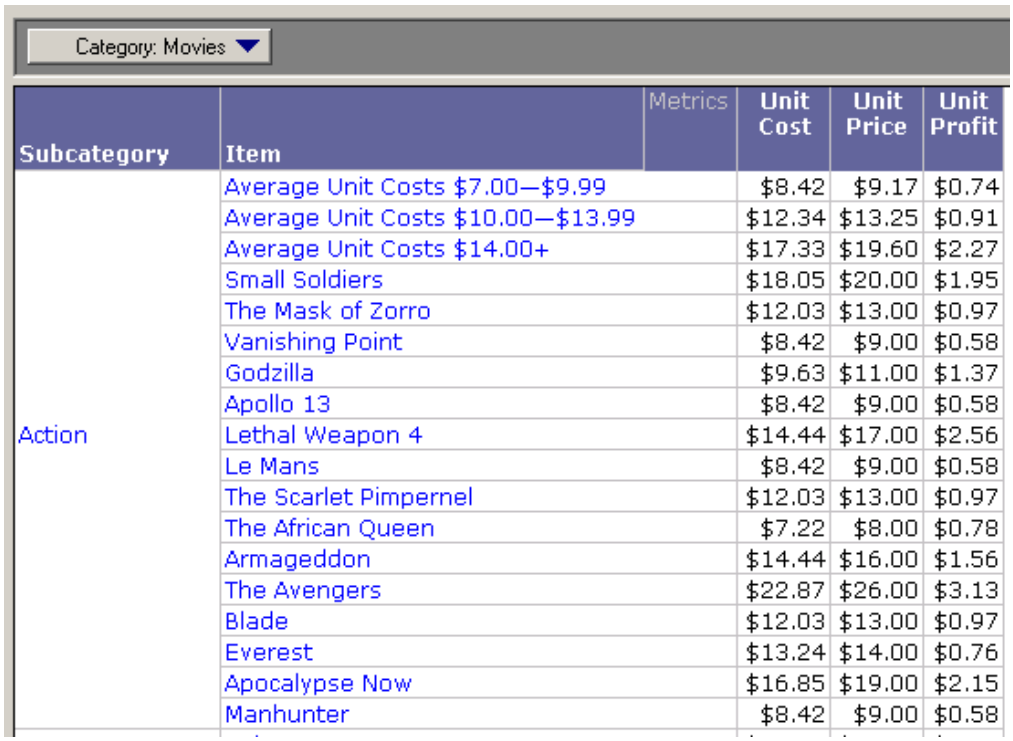
Copyright © 2024 All Rights Reserved

You then type a name for the group, such as Average Unit Costs $7.00-$9.99, and then click **OK**. A derived element is created for the Average Unit Costs $7.00-$9.99 group, which displays an average of the data for Vanishing Point, Godzilla, Apollo 13, Le Mans, The African Queen, and Manhunter.

When you create the quick calculation, the resulting derived element appears at the top of the report, and its attribute elements remain below it. To hide these elements, use the Derived Elements Editor. To show or hide attribute elements using the Derived Elements Editor, see *Displaying derived elements and their attribute elements simultaneously, page 157*.

You can then group the rest of the items into two other groups, one for Average Unit Costs $10.00-$13.99 and one for Unit Costs $14.00+. The resulting report is shown below.

Copyright © 2024 All Rights Reserved

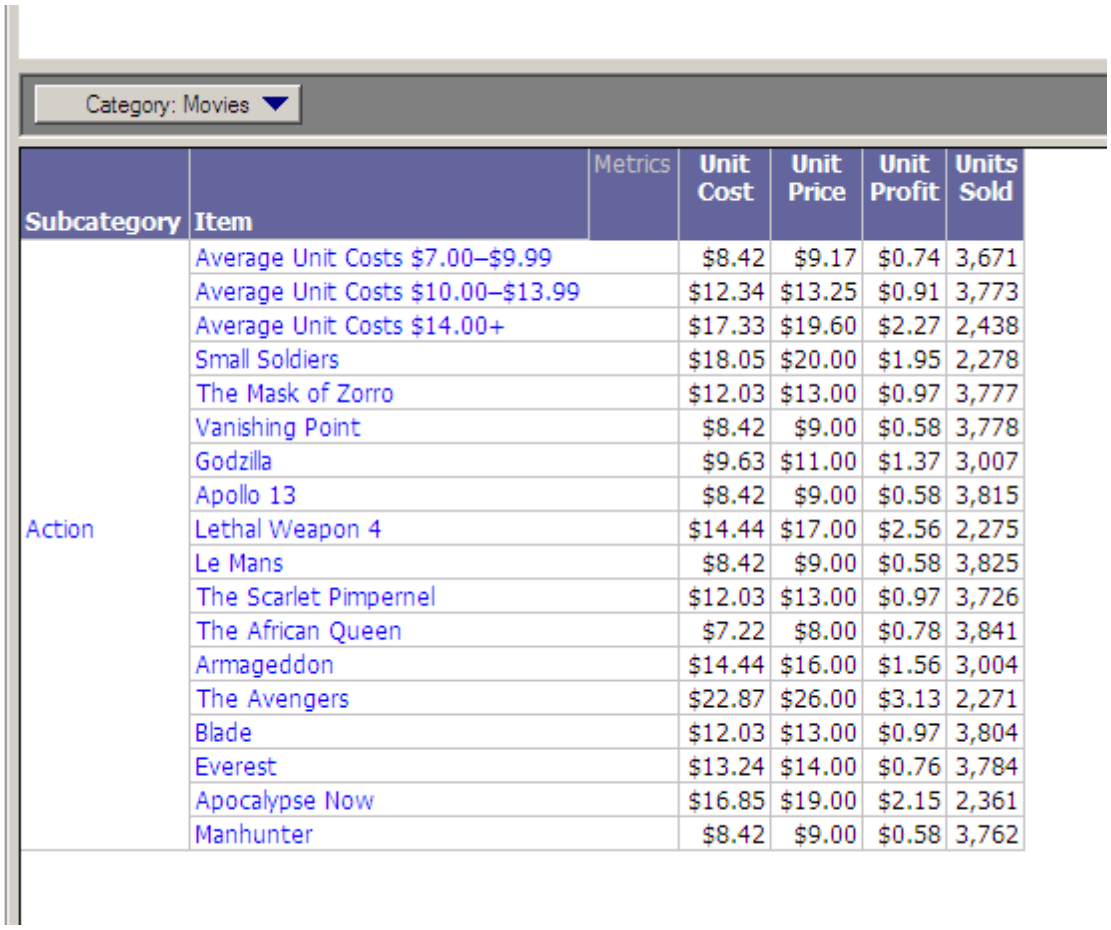


Category: Movies

Subcategory	Item	Metrics	Unit Cost	Unit Price	Unit Profit
	Average Unit Costs $7.00—$9.99		$8.42	$9.17	$0.74
	Average Unit Costs $10.00—$13.99		$12.34	$13.25	$0.91
	Average Unit Costs $14.00+		$17.33	$19.60	$2.27
	Small Soldiers		$18.05	$20.00	$1.95
	The Mask of Zorro		$12.03	$13.00	$0.97
	Vanishing Point		$8.42	$9.00	$0.58
	Godzilla		$9.63	$11.00	$1.37
	Apollo 13		$8.42	$9.00	$0.58
Action	Lethal Weapon 4		$14.44	$17.00	$2.56
	Le Mans		$8.42	$9.00	$0.58
	The Scarlet Pimpernel		$12.03	$13.00	$0.97
	The African Queen		$7.22	$8.00	$0.78
	Armageddon		$14.44	$16.00	$1.56
	The Avengers		$22.87	$26.00	$3.13
	Blade		$12.03	$13.00	$0.97
	Everest		$13.24	$14.00	$0.76
	Apocalypse Now		$16.85	$19.00	$2.15
	Manhunter		$8.42	$9.00	$0.58

You can now view your data at a new summarized level. The report shows a relatively sizable average unit profit for items with unit costs greater than $14.00. You can continue your analysis to see whether the higher prices are affecting the average number of items sold, as shown below.

Copyright © 2024 All Rights Reserved



Category: Movies

Subcategory	Item	Metrics	Unit Cost	Unit Price	Unit Profit	Units Sold
	Average Unit Costs $7.00–$9.99		$8.42	$9.17	$0.74	3,671
	Average Unit Costs $10.00–$13.99		$12.34	$13.25	$0.91	3,773
	Average Unit Costs $14.00+		$17.33	$19.60	$2.27	2,438
	Small Soldiers		$18.05	$20.00	$1.95	2,278
	The Mask of Zorro		$12.03	$13.00	$0.97	3,777
	Vanishing Point		$8.42	$9.00	$0.58	3,778
	Godzilla		$9.63	$11.00	$1.37	3,007
	Apollo 13		$8.42	$9.00	$0.58	3,815
Action	Lethal Weapon 4		$14.44	$17.00	$2.56	2,275
	Le Mans		$8.42	$9.00	$0.58	3,825
	The Scarlet Pimpernel		$12.03	$13.00	$0.97	3,726
	The African Queen		$7.22	$8.00	$0.78	3,841
	Armageddon		$14.44	$16.00	$1.56	3,004
	The Avengers		$22.87	$26.00	$3.13	2,271
	Blade		$12.03	$13.00	$0.97	3,804
	Everest		$13.24	$14.00	$0.76	3,784
	Apocalypse Now		$16.85	$19.00	$2.15	2,361
	Manhunter		$8.42	$9.00	$0.58	3,762

As shown in the report above, the higher prices have no negative effect on the number of items sold for the items you expect a higher profit margin on. This type of analysis can lead you to update your pricing guidelines to maximize profits for items of varying values.

Creating a derived element on a report with a quick calculation

The steps below describe how to create a quick calculation on a report connected to an Intelligent Cube.

Prerequisites

- The report on which you create the derived element is connected to an active Intelligent Cube.

Copyright © 2024 All Rights Reserved

- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To create a derived element with a quick calculation

1. In MicroStrategy Developer or Web, log in to a project that contains reports connected to active Intelligent Cubes.
2. Navigate to and run the Intelligent Cube report. View the report in either Grid View or Grid and Graph View.
3. In the grid display of the report, press the **CTRL** key and select multiple attribute elements within the same attribute.
4. Right-click your selection, point to **Create Calculation**, and then select one of the following calculations.

   - If you selected exactly two attribute elements, you can choose from all of the calculations listed below.

   - If you select more than two attribute elements, Subtract and Divide are not available calculations as they can only accept two operands.
   - If you want to create a subtraction or division including more than two attribute elements, you must use the Derived Elements Editor.

   - **Sum**
   - **Subtract**
   - **Average**
   - **Greatest**
   - **Least**
   - **Divide**

Copyright © 2024 All Rights Reserved

The Defining Group dialog box opens.

5. Type a name for the derived element and click **OK**.

The calculation is created as a derived element and displayed on the report. You can modify the derived element using the Derived Elements Editor (see *Using the Derived Elements Editor, page 124*).

Creating a derived element on standard reports with a quick calculation

You can create quick calculations on standard reports in View Mode in MicroStrategy Web. For steps to create quick calculations on standard reports in MicroStrategy Web, see the MicroStrategy Web Help.

Creating a derived element on Grid/Graphs with a quick calculation

You can create quick calculations on Grid/Graphs in documents in Interactive or Editable Mode in MicroStrategy Web. For steps to create quick calculations on Grid/Graphs in documents in MicroStrategy Web, see the MicroStrategy Web Help.

Creating quick sorts

Using right-click options, you can group attribute elements into a single derived element that sorts the attribute elements in any order you choose. This enables you to do more advanced attribute element sorting than simple ascending or descending sorts.

For example, you have a report that lists the vendors with the ten largest open accounts payable amounts, as shown below.

Copyright © 2024 All Rights Reserved

Vendor	Metrics	Open Payable Amount	Gross Payable Amount
Best Insurance		$231,162	$4,827,296
J.D. Soft and Company		$182,003	$845,007
Grocers Market		$168,114	$1,592,643
Care Pharmacy		$147,501	$2,114,454
Highcard Corporation		$139,892	$1,855,994
Jupiter Networks		$96,014	$3,997,077
Fiber Optics Networks		$91,865	$2,415,128
MEICO Insurance		$84,886	$358,385
Digivision Systems Corp.		$84,323	$4,105,835
Money Market Credit Corp		$79,177	$207,078

After reviewing the list of vendors with the largest open payable amounts from highest to lowest, you can sort the vendors into any order that meets your requirements. You can achieve this by right-clicking the Vendor attribute, pointing to **Sort**, and then selecting **List (custom)**. The Derived Elements Editor opens with all of the attribute elements listed as selected objects.

A quick sort is created as a Group derived element. For information on Group derived elements, see *Group derived element, page 97*.

You can move the elements up and down in the Selected objects list to re-order them on the report. When you are ready, click **OK** to accept the changes and return to the report. Notice the new order of Vendor attribute elements in the report shown below.

Vendor	Metrics	Open Payable Amount	Gross Payable Amount
Fiber Optics Networks		$91,865	$2,415,128
Jupiter Networks		$96,014	$3,997,077
Best Insurance		$231,162	$4,827,296
MEICO Insurance		$84,886	$358,385
Care Pharmacy		$147,501	$2,114,454
Grocers Market		$168,114	$1,592,643
J.D. Soft and Company		$182,003	$845,007
Highcard Corporation		$139,892	$1,855,994
Digivision Systems Corp.		$84,323	$4,105,835
Money Market Credit Corp		$79,177	$207,078

Copyright © 2024 All Rights Reserved

Creating a derived element on Intelligent Cube reports with a quick sort

The steps below show you how to create a quick sort on a report connected to an Intelligent Cube.

Prerequisites

- The report on which you create the derived element is connected to an active Intelligent Cube.
- The attribute you select to create a quick sort for cannot have any existing derived elements defined on it for the report.
- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To create a derived element with a quick sort

1. In MicroStrategy Developer or Web, log in to a project that contains reports connected to active Intelligent Cubes.
2. Navigate to and run the Intelligent Cube report. View the report in either Grid View or Grid and Graph View.
3. Right-click an attribute in the grid display, point to **Sort**, and then select **Custom**. The Derived Elements Editor opens with all of the attribute elements listed as selected objects.

   ⚠ If **Custom** is not an option in the Sort options, this means that the attribute has a derived element defined for it on the report. You must remove any derived elements defined for the attribute before you can create a quick sort to re-order the attribute elements.

4. In the **Selected objects** list, move the attribute elements to the order you want them displayed on the report.
5. Once the attribute elements are ordered appropriately, click **OK**. The Derived Elements Editor closes and you are returned to the report.

Copyright © 2024 All Rights Reserved

The report displays the attribute elements in the new order. You can modify the derived element using the Derived Elements Editor (see *Using the Derived Elements Editor, page 124*).

Creating a derived element on standard reports with a quick sort

You can create quick sorts on standard reports in View Mode in MicroStrategy Web. For steps to create quick sorts on standard reports in MicroStrategy Web, see the MicroStrategy Web Help.

Creating a derived element on Grid/Graphs with a quick sort

The steps below show you how to create a quick sort on a Grid/Graph included in a Report Services document from MicroStrategy Developer.

You can create quick sorts on Grid/Graphs in documents in Interactive or Editable Mode in MicroStrategy Web. For steps to create quick sorts on Grid/Graphs in MicroStrategy Web, see the MicroStrategy Web Help.

Prerequisites

- The Report Services document with a Grid/Graph included in one of the document sections, on which you create the derived element.
- The attribute you select to create a quick sort for cannot have any existing derived elements defined on it for the report.
- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To create a derived element with a quick sort

1. In MicroStrategy Developer, log in to a project that contains Grid/Graphs in Report Services documents.

Copyright © 2024 All Rights Reserved

2. Open the document in Design View and expand the document section that contains the Grid/Graph.
3. Right-click the Grid/Graph and select **Edit Grid**.
4. Right-click an attribute, point to **Sort**, and then select **List (custom)**. The Derived Elements Editor opens with all of the attribute elements listed as selected objects.

   If **List (custom)** is not an option in the Sort options, this means that the attribute has a derived element defined for it on the source report. You must remove any derived elements defined for the attribute before you can create a quick sort to re-order the attribute elements.

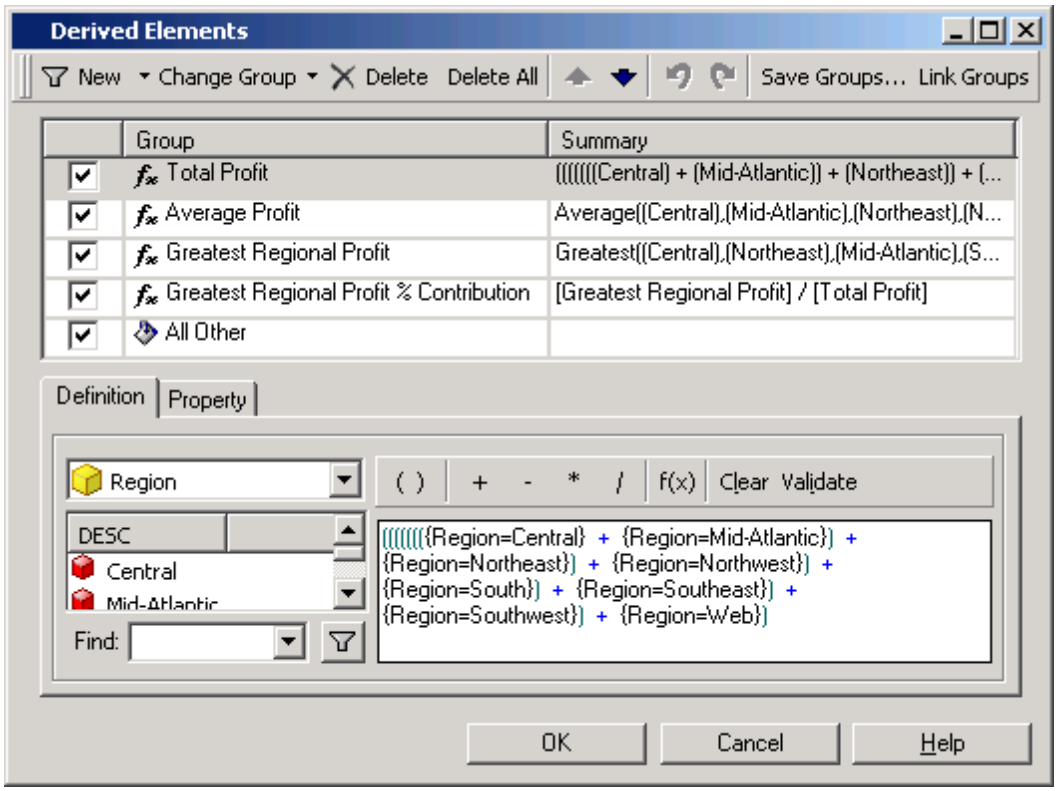
5. In the **Selected objects** list, re-order the attribute elements as you want to view them on the report.
6. Once the attribute elements are ordered appropriately, click **OK**.

The document displays the attribute elements in the new order. You can modify the derived element using the Derived Elements Editor (see *Using the Derived Elements Editor, page 124*).

Using the Derived Elements Editor

While the quick groups features described in *Quickly creating groups, calculations, and sorts, page 107* enable you to quickly create derived elements using right-click menu options while reviewing report results, the Derived Elements Editor provides the full set of derived elements functionality when creating derived elements. The Derived Elements Editor is shown in the image below.

Copyright © 2024 All Rights Reserved



For example, if you use right-click menu options create a quick calculation on a report connected to an Intelligent Cube, the expression can only include one type of function or operand such as +, /, and Average. However, in the Derived Elements Editor, you can create expressions with a valid combination of different functions. You can create a derived element from the Derived Elements Editor with an expression of the following form:

`(AttributeElement1 + AttributeElement2) / Sum (AllAttributeElements)`

You can modify derived elements in the following ways using the Derived Elements Editor:

- *Applying derived element values to subtotals, page 151*
- *Displaying derived elements or their attribute elements, page 154*
- *Displaying derived elements and their attribute elements simultaneously, page 157*

Copyright © 2024 All Rights Reserved

- *Formatting derived elements, page 160*
- *Creating and using stand-alone derived elements, page 146*
- Creating Filter derived elements with Not in List and Where filter qualifications
- Creating advanced Calculation derived elements
- Deleting derived elements

Sample report

The report shown below is used in the procedures and examples that follow for creating derived elements with the Derived Elements Editor.

Copyright © 2024 All Rights Reserved

Region	Category	Metrics	Profit
Central	Books		$55,495
	Electronics		$935,123
	Movies		$16,347
	Music		$78,357
Mid-Atlantic	Books		$46,461
	Electronics		$3,933,032
	Movies		$14,351
	Music		$13,737
Northeast	Books		$455,964
	Electronics		$1,500,514
	Movies		$27,367
	Music		$25,895
Northwest	Books		$19,734
	Electronics		$1,604,110
	Movies		$5,639
	Music		$5,508
South	Books		$216,663
	Electronics		$739,064
	Movies		$12,905
	Music		$11,882
Southeast	Books		$25,609
	Electronics		$419,847
	Movies		$37,704
	Music		$6,965
Southwest	Books		$41,576
	Electronics		$686,421
	Movies		$60,787
	Music		$11,294
Web	Books		$29,195
	Electronics		$469,635
	Movies		$42,611
	Music		$7,952

Accessing the Derived Elements Editor

Using the Derived Elements Editor, you can create derived elements with groups, filters, and calculations. These derived elements can be created from the Derived Elements Editor in MicroStrategy Developer and Web in the reporting objects listed below:

Copyright © 2024 All Rights Reserved

- A standard report.
- A report which is connected to an active Intelligent Cube. The table below lists the views you can access the Derived Elements Editor from, in MicroStrategy Developer and Web. To access the Derived Elements Editor from reports, see *To access the Derived Elements Editor in reports connected to an active Intelligent Cube, page 129*.
- A Grid/Graph in a Report Services document. The table below lists the modes you can access the Derived Elements Editor from, in MicroStrategy Developer and Web.

Report Object Type	View/Mode	MicroStrategy Developer/Web	Can Access Derived Element Editor
Standard report	Grid View	Web	Yes
Report connected to an active Intelligent Cube	Grid View	Developer and Web	Yes
	Grid and Graph View	Developer and Web	Yes
	Design View	Developer	Yes
	Design View	Web	No
Grid/Graph in a Report Services document	Design View	Developer	Yes
	Design Mode	Web	No
	Interactive Mode (Grid/Graph must be displayed as a Grid or a Grid and Graph)	Web	Yes
	Editable Mode (Grid/Graph must be displayed as a Grid or a Grid and Graph)	Web	Yes

Copyright © 2024 All Rights Reserved

You can also create stand-alone derived elements by accessing the Derived Elements Editor from outside reports or Grid/Graphs. Stand-alone derived elements can be used by multiple reports and Grid/Graphs. For information on using a derived element in multiple reports and accessing the Derived Elements Editor to create stand-alone derived elements, see *Creating and using stand-alone derived elements, page 146*.

To access the Derived Elements Editor in reports connected to an active Intelligent Cube

1. In MicroStrategy Developer or Web, log in to a project that contains reports connected to active Intelligent Cubes.
2. Open a report and run it. View it in either Grid View, Grid and Graph View, or Design View.
3. In the grid display of the report, right-click the attribute you want to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.

To access the Derived Elements Editor in standard reports

1. In MicroStrategy Web, log in to a project that contains the report.
2. Open the report in View Mode, in either Grid View or Grid and Graph View.
3. In the grid display of the report, right-click the attribute you want to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.

To access the Derived Elements Editor in Grid/Graphs

1. In MicroStrategy Developer or Web, log in to a project that contains Grid/Graphs in Report Services documents.

Copyright © 2024 All Rights Reserved

2. Open the document. Accessing the Derived Elements Editor depends on what mode you are viewing the document in. Follow the steps below depending on which view or mode you are using:
    - Design View in MicroStrategy Developer only:
        - Expand the document section that contains the Grid/Graph.
        - Right-click the Grid/Graph and select **Edit Grid**.
        - Right-click the attribute to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.
    - Interactive Mode or Editable Mode in MicroStrategy Web only:
        - In the grid display of the Grid/Graph, right-click the attribute to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.

Grouping attribute elements to create a derived element

When viewing a report you can combine attribute elements into a single group of aggregated data. This combined set of attribute elements is called a Group derived element. This derived element enables you to view and analyze the attribute elements' data as a single, distinct group.

For example, in a report with Region and Category attributes and a Profit metric, you can combine the regions on the report into various geographical groups. The final report you create with attribute elements grouped based on geography is shown below.

Copyright © 2024 All Rights Reserved

Region	Category	Metrics	Profit
East Coast	Books		$528,034
	Electronics		$5,853,393
	Movies		$79,423
	Music		$46,597
West Coast	Books		$61,310
	Electronics		$2,290,530
	Movies		$66,426
	Music		$16,803
Central and South	Books		$272,158
	Electronics		$1,674,186
	Movies		$29,251
	Music		$90,238
Web	Books		$29,195
	Electronics		$469,635
	Movies		$42,611
	Music		$7,952

The steps below show you how to create a Group derived element on a report, as well as specific instructions to create the sample report shown above.

Prerequisites

- The report on which you create the derived element is connected to an active Intelligent Cube, or the Report Services document with a Grid/Graph included in one of the document sections, on which you create the derived element.
- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To create a Group derived element

While this procedure creates only Group derived elements, you can create any combination of Group, Filter, and Calculation derived elements on a report or Grid/Graph.

Copyright © 2024 All Rights Reserved

1. Log in to a project in MicroStrategy Developer. For steps to use the Derived Elements Editor to create derived elements in MicroStrategy Web, see the MicroStrategy Web Help.
2. Open the report or Grid/Graph (the example scenario uses a report connected to an Intelligent Cube), as follows:
   - To access the Derived Elements Editor for a report connected to an Intelligent Cube:
     - Execute the report and run it in Grid View or Grid and Graph View.
     - Right-click the attribute to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.

     For the example scenario, right-click the **Region** attribute, and click **Derived Elements**.
   - To access the Derived Elements Editor for a Grid/Graph in a Report Services document:
     a. Expand the document section that contains the Grid/Graph.
     b. Double-click the Grid/Graph to edit it.
     c. Right-click the attribute to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.

        For the example scenario, right-click the **Region** attribute, and click **Derived Elements**.
3. To create a new Group derived element, from the **New** drop-down list select **Group**.

   Two new derived elements are created, a blank Group derived element and an All Other derived element. The All Other derived element is a collection of all attribute elements that are not included in any of the

Copyright © 2024 All Rights Reserved

other derived elements for the attribute. For further explanation of the All Other derived element, see *All Other derived element, page 105*.

4. Select the new Group derived element. This displays the available attribute elements in the Definition tab.

   For the example scenario, the Region attribute elements are displayed.

5. From the left pane, select attribute elements to include in the derived element, and then click the right arrow (>) to add your selections to the Selected objects pane.

   For the example scenario, select the **Mid-Atlantic**, **Northeast**, and **Southeast** attribute elements.

6. To rename the Group derived element, from the **Change Group** drop-down list, select **Rename Group**. Type a name for the derived element.

   For the example scenario, rename the group as **East Coast**.

7. From the **Property** tab, you can make various modifications to the new derived element such as:

   - *Displaying derived elements or their attribute elements, page 154*
   - *Applying derived element values to subtotals, page 151*
   - *Displaying derived elements and their attribute elements simultaneously, page 157*

8. From the **Change Group** drop-down list, you can format derived element headers and values. For information on these formatting techniques, see *Formatting derived elements, page 160*.

9. You can change the order in which the derived elements are displayed on the report using the up ( ) and down ( ) arrows.

10. You can continue to create more derived elements, or you can click **OK** to close the Derived Elements Editor and return to the report. The steps below continue the example scenario.

Copyright © 2024 All Rights Reserved

11. From the **New** drop-down list, select **Group**. A blank group is created.
12. Select the new derived element. This displays the available attribute elements in the Definition tab.
13. From the left pane, select the **Northeast** and **Southeast** attribute elements, and then click the right arrow (>) to add your selections to the Selected objects pane.
14. From the **Change Element** drop-down list, select **Rename Group**. Type **West Coast** to rename the Group derived element.
15. From the **New** drop-down list, select **Group**. A blank group is created.
16. Select the new derived element. This displays the available attribute elements in the Definition tab.
17. From the left pane, select the **Central** and **South** attribute elements, and then click the right arrow (>) to add your selections to the Selected objects pane.
18. From the **Change Group** drop-down list, select **Rename Group**. Type **Central and South** to rename the derived element group.
19. You can save your derived element for the report or Grid/Graph, or save the derived element as a stand-alone object that can be used by multiple reports and Grid/Graphs:
    - To save the derived element for the report or Grid/Graph, click **OK**. The Derived Elements Editor closes and you are returned to the report or document.
    - To save the derived element as a stand-alone object that can be used by multiple reports and Grid/Graphs, click **Save Groups**. Choose a location to save the derived element to, type a name, and click **Save**. Click **OK**.

      Stand-alone derived elements can only be modified by editing the stand-alone object; you cannot modify them from within reports or

Copyright © 2024 All Rights Reserved

Grid/Graphs. For information stand-alone derived elements, see *Creating and using stand-alone derived elements, page 146*.

If you used the steps above to create the sample report, the report is displayed with the regions grouped into East Coast, West Coast, Central and South, and Web.

Filtering attribute elements to create a derived element

While viewing a report, you can combine attribute elements into a single group of aggregated data using filter qualifications. This group of attribute elements is called a Filter derived element. This derived element enables you to view and analyze the attribute elements' data as a single distinct group by utilizing various filter qualifications.

For example, in a report with Region and Category attributes and a Profit metric, you can filter the regions on the report into various geographical groups based on the region names. The final report you create with Filter derived elements is shown below.

Region	Category (Metrics)	Profit
Southern Regions	Books	$283,848
	Electronics	$1,845,331
	Movies	$111,396
	Music	$30,141
Northern Regions	Books	$475,697
	Electronics	$3,104,624
	Movies	$33,006
	Music	$31,404
Central	Books	$55,495
	Electronics	$935,123
	Movies	$16,347
	Music	$78,357
Mid-Atlantic	Books	$46,461
	Electronics	$3,933,032
	Movies	$14,351
	Music	$13,737
Web	Books	$29,195
	Electronics	$469,635
	Movies	$42,611
	Music	$7,952

Copyright © 2024 All Rights Reserved

The steps below show you how to create a Filter derived element on a report, as well as specific instructions to create the report shown above.

Prerequisites

- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To create a Filter derived element

While this procedure creates only Filter derived elements, you can create any combination of Group, Filter, and Calculation derived elements on a report or Grid/Graph.

1. Log in to a project in MicroStrategy Developer. For steps to use the Derived Elements Editor to create derived elements in MicroStrategy Web, see the MicroStrategy Web Help.
2. Open the report or Grid/Graph (the example scenario uses a report connected to an Intelligent Cube), as follows:
   - To access the Derived Elements Editor for a report connected to an Intelligent Cube:
     a. Execute the report. Then view it in either Grid View or Grid and Graph View.
     b. Right-click the attribute to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.

        For the example scenario, right-click the **Region** attribute, and click **Derived Elements**.
   - To access the Derived Elements Editor for a Grid/Graph in a Report Services document:

Copyright © 2024 All Rights Reserved

a. Expand the document section that contains the Grid/Graph.

b. Double-click the Grid/Graph to edit it.

c. Right-click the attribute to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.

   For the example scenario, right-click the **Region** attribute, and click **Derived Elements**.

3. To create a new Filter derived element, from the **New** drop-down list, select **Filter**.

   Two new derived elements are created, a blank Filter derived element and an All Other derived element. The All Other derived element is a collection of all attribute elements that are not included in any of the other derived elements for the attribute. For further explanation of the All Other group, see *All Other derived element, page 105*.

4. Select the new Filter derived element.

5. From the **Definition** tab, click **Click here to start a new qualification**.

6. Click **Field**, and then select an attribute.

   For the example scenario, select **Region**.

7. Click **Operator**, and then select one of the following operators to create a filter qualification (for the example scenario, select **Where**):

   - **In list**: Returns attribute data for the list of attribute elements you select. Click **Value**, and then select the attribute elements to return data for.
   - **Not in List**: Returns attribute data for the list of attribute elements that are not in the list of attribute elements you select. Click **Value**, and then select the attribute elements to exclude data for.

Copyright © 2024 All Rights Reserved

- **Where**: Returns attribute data based on a filter qualification of an attribute form. Proceed to the next step to select an attribute form and complete the filter qualification.

8. For filter qualifications that use the operator Where, new Field, Operator, and Value fields appear. Follow the steps below to complete the filter qualification:

   a. Click **Field**, and then select an attribute form. For the example scenario, select **DESC**.

   b. Click **Operator**, and then select the operator for the filter qualification on the attribute form. For the example scenario, select **Begins with**.

   c. Click **Value**, and use one of the options to enter in the required value. For the example scenario, select **Type a value**, and then type **South**.

9. To rename the derived element group, from the **Change Group** drop-down list, and select **Rename Group**. Type a name for the Filter derived element.

   For the example scenario, rename the derived element as **Southern Regions**.

10. From the **Property** tab, you can make various modifications to the new derived element such as:

    - *Displaying derived elements or their attribute elements, page 154*
    - *Applying derived element values to subtotals, page 151*
    - *Displaying derived elements and their attribute elements simultaneously, page 157*

11. From the **Change Group** drop-down list, you can format derived element headers and values. For information on these formatting techniques, see *Formatting derived elements, page 160*.

Copyright © 2024 All Rights Reserved

12. You can change the order in which the derived elements are displayed on the report using the up ( ) and down ( ) arrows.
13. You can continue to create more derived elements, or you can click **OK** to close the Derived Elements Editor and return to the report. The steps below continue the example scenario.
14. From the **New** drop-down list, select **Filter**. A blank Filter derived element is created.
15. Select the new derived element.
16. From the **Definition** tab, click **Click here to start a new qualification**.
17. Click **Field**, and then select **Region**.
18. Click **Operator**, and then select **Where**.
19. Click **Field**, and then select the **DESC** attribute form.
20. Click **Operator**, and then select **Begins with**.
21. Click **Value**, and then select **Type a value**, and then type **North**.
22. To rename the derived element group, from the **Change Group** drop-down list, and select **Rename Group**. Type **Northern Regions** to rename the derived element.
23. You can save your derived element for the report or Grid/Graph, or save the derived element as a stand-alone object that can be used by multiple reports and Grid/Graphs:
    - To save the derived element for the report or Grid/Graph, click **OK**.
    - To save the derived element as a stand-alone object that can be used by multiple reports and Grid/Graphs, click **Save Groups**. Choose a location to save the derived element to, type a name, and click **Save**. Click **OK**.

Copyright © 2024 All Rights Reserved

Stand-alone derived elements can only be modified by editing the stand-alone object; you cannot modify them from within reports or Grid/Graphs. For information on using derived elements, see *Creating and using stand-alone derived elements, page 146*.

If you used the steps above to create the sample report, the report is displayed with the regions grouped into Southern Regions and Northern Regions, along with the Central, Mid-Atlantic, and Web regions.

Using calculations to create derived elements

When viewing a report you can combine attribute elements into a single set of aggregated data using calculations. This combined set of attribute elements is called a Calculation derived element. This derived element enables you to view and analyze the attribute elements' data as a single, distinct group by utilizing various calculation functions and operators.

For example, in a report with Region and Category attributes and a Profit metric, you can combine the regions on the report into various groups for profit analysis. The final report you create with Calculation derived elements is shown below.

Region	Category / Metrics	Profit
Total Profit	Books	$890,696
	Electronics	$10,287,744
	Movies	$217,710
	Music	$161,590
Average Profit	Books	$111,337
	Electronics	$1,285,968
	Movies	$27,214
	Music	$20,199
Greatest Regional Profit	Books	$455,964
	Electronics	$3,933,032
	Movies	$60,787
	Music	$78,357
Greatest Regional Profit % Contribution	Books	51.19%
	Electronics	38.23%
	Movies	27.92%
	Music	48.49%

Copyright © 2024 All Rights Reserved

The steps below show you how to create a Calculation derived element on a report, as well as specific instructions to create the sample report shown above.

Prerequisites

- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To create a Calculation derived element

While this procedure creates only Calculation derived elements, you can create any combination of Group, Filter, and Calculation derived elements on a report or Grid/Graph.

1. Log in to a project in MicroStrategy Developer. For steps to use the Derived Elements Editor to create derived elements in MicroStrategy Web, see the MicroStrategy Web Help.
2. Open the report or Grid/Graph (the example scenario uses a report connected to an Intelligent Cube), as follows:
   - To access the Derived Elements Editor for a report connected to an Intelligent Cube:
     a. Execute the report. The view it in either Grid View or Grid and Graph View.
     b. Right-click the attribute to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.

        For the example scenario, right click the **Region** attribute, and click **Derived Elements**.
   - To access the Derived Elements Editor for a Grid/Graph in a Report Services document:

Copyright © 2024 All Rights Reserved

a. Expand the document section that contains the Grid/Graph.

b. Double-click the Grid/Graph to edit it.

c. Right-click the attribute to create or modify derived elements for, and click **Derived Elements**. The Derived Elements Editor opens.

   For the example scenario, right click the **Region** attribute, and click **Derived Elements**.

3. To create a new Calculation derived element, from the **New** drop-down list, select **Calculation**.

   Two new derived elements are created, a blank Calculation derived element and an All Other derived element. The All Other derived element is a collection of all attribute elements that are not included in any of the other derived elements for the attribute. For further explanation of the All Other derived element, see *All Other derived element, page 105*.

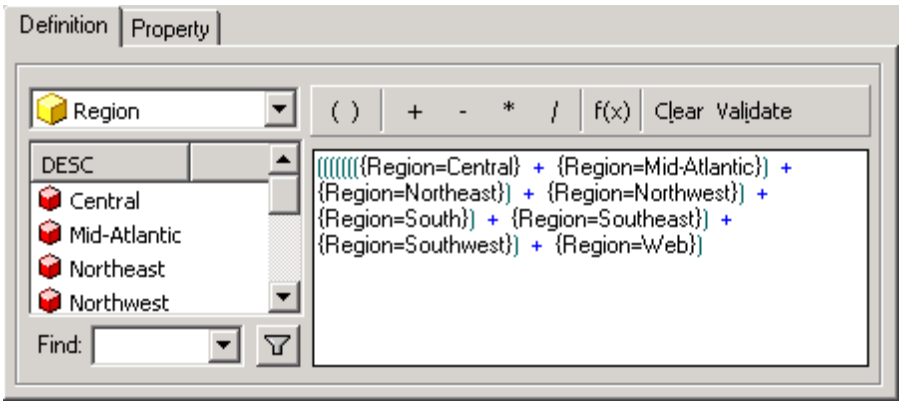
4. Select the new Calculation derived element. This displays the available attribute elements in the Definition tab.

   For the example scenario, the Region attribute elements are displayed.

5. From the left pane, select attribute elements to include in the Calculation derived element, and add them to the expression area on the right (shown below). You can also use the toolbar above the expression to include operators and functions, as well as validate or clear the expression.

   For the example scenario, select all the attribute elements available and drag the entire selection to the expression area. This adds all the attribute elements to the expression area, and automatically use the addition operator to combine the attribute elements, as shown below.

Copyright © 2024 All Rights Reserved



6. To rename the derived element, from the **Change Group** drop-down list, select **Rename Group**. Type a name for the derived element.

   For the example scenario, rename the group as **Total Profit**.

7. From the **Property** tab, you can make various modifications to the new derived element, such as:

   - *Displaying derived elements or their attribute elements, page 154*
   - *Applying derived element values to subtotals, page 151*
   - *Displaying derived elements and their attribute elements simultaneously, page 157*

8. From the **Change Group** drop-down list, you can format derived element headers and values. For information on these formatting techniques, see *Formatting derived elements, page 160*.

9. You can change the order in which the derived elements are displayed on the report using the up ( ) and down ( ) arrows.

10. You can continue to create more derived elements, or you can click **OK** to close the Derived Elements Editor and return to the report. The steps below continue the example scenario.

11. From the **New** drop-down list, select **Calculation**. A blank Calculation derived element is created.

12. Select the new derived element.

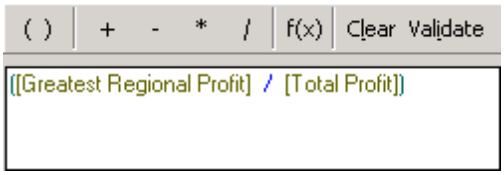
Copyright © 2024 All Rights Reserved

13. On the **Definition** tab, click **f(x)**. The Insert Function Wizard opens.
14. Select the **Average** function, and click **Next**. The Arguments page opens.
15. Click **...** (browse) next to Argument 1, and in the Select an Object dialog box, select **Central**. Then click **OK**. Repeat this step for each successive argument, selecting the following regions:
    - Mid-Atlantic
    - Northeast
    - Northwest
    - South
    - Southeast
    - Southwest
    - Web
16. Click **Finish**.
17. From the **Change Group** drop-down list, select **Rename Group**. Type **Average Profit** to rename the derived element.
18. From the **New** drop-down list, select **Calculation**. A blank Calculation derived element is created.
19. Select the new derived element.
20. On the **Definition** tab, click **f(x)**. The Insert Function Wizard opens.
21. Select the **Greatest** function, and click **Next**. The Arguments page opens.
22. Click **...** (browse) next to Argument 1, and in the Select an Object dialog box, select **Central**. Then click **OK**. Repeat this step for each successive argument, selecting the following regions:

Copyright © 2024 All Rights Reserved

- Mid-Atlantic
- Northeast
- Northwest
- South
- Southeast
- Southwest
- Web

23. From the **Change Group** drop-down list, select **Rename Group**. Type **Greatest Regional Profit** to rename the derived element.
24. From the **New** drop-down list, select **Calculation**. A blank Calculation derived element is created.
25. Select the new derived element.
26. On the **Definition** tab, from the drop-down list select **Groups**.
27. From the left pane, select **Greatest Regional Profit** and drag and drop it into the expression area on the right.
28. Click **/**.
29. From the left pane, select **Total Profit** and drag and drop it into the expression area on the right. Your final expression should appear as shown below:



30. From the **Change Group** drop-down list, select **Rename Group**. Type **Greatest Regional Profit % Contribution** to rename the derived element.

Copyright © 2024 All Rights Reserved

31. You can save your derived element for the report or Grid/Graph, or save the derived element as a stand-alone object that can be used by multiple reports and Grid/Graphs:

    - To save the derived element for the report or Grid/Graph, click **OK**. The Derived Elements Editor closes and you are returned to the report or document.
    - To save the derived element as a stand-alone object that can be used by multiple reports and Grid/Graphs, click **Save Groups**. Choose a location to save the derived element to, type a name, and click **Save**. Click **OK**. The Derived Elements Editor closes and you are returned to the report or document.

      Stand-alone derived elements can only be modified by editing the stand-alone object; you cannot modify them from within reports or Grid/Graphs. For information on sharing derived elements, see *Creating and using stand-alone derived elements, page 146*.

If you used the steps above to create the sample report, the report is displayed with the regions grouped into Total Profit, Average Profit, Greatest Regional Profit, and Greatest Regional Profit % Contribution.

Creating and using stand-alone derived elements

A stand-alone derived element can be connected to matching attributes in reports connected to Intelligent Cubes and Grid/Graphs. This enables you to use a derived element in multiple reports and Grid/Graphs.

To use a derived element in multiple reports and Grid/Graphs, you must create a stand-alone derived element. There are two methods with to create a stand-alone derived element:

- Create a derived element from within a report or Grid/Graph and then save it as a stand-alone object. You can use the Derived Elements Editor to create and save the derived element as a stand-alone derived element.

Copyright © 2024 All Rights Reserved

You can use this method in MicroStrategy Developer and Web. For information on creating a derived element within a report or Grid/Graph using the Derived Elements Editor, see *Using the Derived Elements Editor, page 124*.

Once you save a derived element as a stand-alone derived element, it can no longer be modified from within the report or Grid/Graph.

- Create a stand-alone derived element outside of any report or Grid/Graph. You can use this method only in MicroStrategy Developer.

Once created, a stand-alone derived element has the following functionality:

- All Group, Calculation, Filter, and All Other derived elements are saved as part of the derived element. You cannot select a subset of the derived elements; you must save and share the entire collection of derived elements.
- A stand-alone derived element can only be connected to the attribute that was used to define the derived element.
- The stand-alone derived element itself can be modified, but you cannot modify it from within a report or Grid/Graph. Any modifications for the derived element are applied to the derived element in all of the reports and Grid/Graphs that it is used in.
- The stand-alone derived element can only be deleted if it is not used in any report or Grid/Graph. A list of reports and Grid/Graphs that use the derived element is displayed when you attempt to delete a stand-alone derived element.

Creating a stand-alone derived element

The steps below show you how to create a stand-alone derived element.

Prerequisites

Copyright © 2024 All Rights Reserved

- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To create a stand-alone derived element

1. In MicroStrategy Developer, log in to a project.
2. Choose **File** > **New** > **Derived Element**. The Select Attribute dialog box opens.
3. Browse to and select the attribute you want to base your derived element on, and then click **Open**. The Derived Element Editor opens.
4. You can create Group, Filter, and Calculation derived elements to define your stand-alone derived element. With the exception any steps to access or close the Derived Elements Editor, you can use the same processes and techniques described in the following sections to create derived elements for a stand-alone derived element:
   - *Grouping attribute elements to create a derived element, page 130*
   - *Filtering attribute elements to create a derived element, page 135*
   - *Using calculations to create derived elements, page 140*
5. Click **Save and Close**.
6. Type a name and click **Save**. The derived element is saved as a stand-alone object.

You can now use your derived element in reports connected to Intelligent Cubes and Grid/Graphs in Report Services documents.

Using a derived element in multiple reports or Grid/Graphs

You can use a stand-alone derived element in multiple reports and Grid/Graphs in documents. To use a stand-alone derived element, you can connect the derived element to an attribute in a report connected to an

Copyright © 2024 All Rights Reserved

Intelligent Cube or to an attribute in a Grid/Graph in a Report Services document. However, a derived element can only be connected to the attribute used to define the derived element.

For example, if a derived element is created on the Year attribute, this derived element can only be connected to the Year attribute in reports and Grid/Graphs. You cannot connect this derived element to any other attributes such as Quarter, Month, Category, Customer, and so on.

The steps below show you how to apply a stand-alone derived element to an attribute in a report or Grid/Graph.

To connect a stand-alone derived element to an attribute in a report or Grid/Graph

1. Log in to a project in MicroStrategy Developer or Web.
2. Open the report or Grid/Graph, as follows:

- To access the Derived Elements Editor for a report:
  - Execute the report. Then view it in either Grid View or Grid and Graph View.
  - Right-click the attribute to apply a stand-alone derived element to, and click **Derived Elements**. The Derived Elements Editor opens.

    For the example scenario, right-click the Region attribute, and click **Derived Elements**.
- To access the Derived Elements Editor for a Grid/Graph in a Report Services document:
  - Expand the document section that contains the Grid/Graph.
  - Double-click the Grid/Graph to edit it.
  - Right-click the attribute to apply a stand-alone derived element to, and click **Derived Elements**. The Derived Elements Editor opens.

Copyright © 2024 All Rights Reserved

For the example scenario, right-click the Region attribute, and click **Derived Elements**.

To connect a stand-alone derived element to an attribute

If an attribute already has a derived element defined for it in the report or Grid/Graph, connecting a stand-alone derived element overwrites the existing definition.

1. Click **Link Derived Elements**. The Select Derived Elements dialog box opens.
2. Browse to and select the derived element to connect to the attribute, and then click **Open**.
3. Click **OK**.

Editing stand-alone derived elements

Editing a stand-alone derived element updates the derived element in all reports and Grid/Graphs it is used in. Since these modifications are applied to multiple reporting objects, you can only update the stand-alone derived element itself, you cannot update it from a report or Grid/Graph that it is used in.

The steps below show you how to modify a stand-alone derived element.

To edit a stand-alone derived element

1. In MicroStrategy Developer, log in to a project.
2. Browse to and right-click a stand-alone derived element, and then select **Edit**. The Derived Element Editor opens.
3. You can create and modify Group, Filter, and Calculation derived elements to define your stand-alone derived element. With the exception of any steps to access or close the Derived Elements Editor,

Copyright © 2024 All Rights Reserved

you can use the same processes and techniques described in the following sections to create and modify derived elements for a stand-alone derived element:

- *Grouping attribute elements to create a derived element, page 130*
- *Filtering attribute elements to create a derived element, page 135*
- *Using calculations to create derived elements, page 140*

4. Click **Save and Close**.

Defining derived element functionality and formatting

Derived elements provide a wealth of reporting and analysis functionalities that can be defined to achieve your specific requirements. The sections listed below cover various derived elements functionalities and formatting techniques:

- *Applying derived element values to subtotals, page 151*
- *Displaying derived elements or their attribute elements, page 154*
- *Displaying derived elements and their attribute elements simultaneously, page 157*
- *Formatting derived elements, page 160*

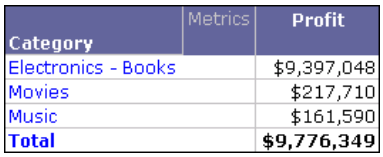
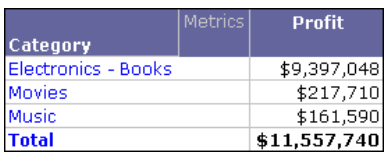
Applying derived element values to subtotals

You can define how each derived element value is applied to all subtotals included on a report or Grid/Graph. The simple report below is used as an example of how you can apply derived element values to subtotals.

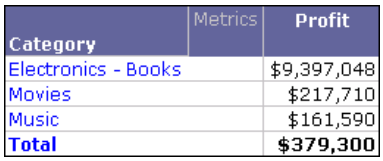
Category / Metrics	Profit
Books	$890,696
Electronics	$10,287,744
Movies	$217,710
Music	$161,590
Total	**$11,557,740**

Copyright © 2024 All Rights Reserved

You have the following options in the Derived Elements Editor for applying derived element values to subtotals. The examples below are all based on the basic report above.

Option Name	Subtotal Behavior	Example
Use this element when calculating subtotals	(default behavior) The derived element values are applied to all subtotals. This means that a derived element's aggregated data is applied to all subtotals for the report, rather than the values for the individual items that are part of a derived element.	In a report with Category and Profit, you create a derived element defined as Electronics - Books. Applying the derived element value to the subtotal means that subtotals uses the value resulting from the subtraction of the Electronics and Books attribute elements, as shown below.  Category / Metrics / Profit Electronics - Books $9,397,048 Movies $217,710 Music $161,590 **Total** **$9,776,349**
Use the individual items that make up this element when calculating subtotals	The values of the separate items that make up the derived element are applied to all subtotals. For Group and Filter derived elements, only attribute elements can be included as items of derived elements. Calculation derived elements can include attribute elements as well as other derived elements.	In a report with Category and Profit, you create a derived element defined as Electronics - Books. Applying the individual item values to the subtotal means that the subtotal uses the individual values of the Electronics and Books attribute elements, as shown below.  Category / Metrics / Profit Electronics - Books $9,397,048 Movies $217,710 Music $161,590 **Total** **$11,557,740** Notice that the subtotal in this

Copyright © 2024 All Rights Reserved

Option Name	Subtotal Behavior	Example
		report is the same value as the original report.
Do not use this element when calculating subtotals	This option excludes the derived element values and separate item values from all subtotals. You can use this option to avoid double counting when an attribute element is included in more than one derived element.	In a report with Category and Profit, you create a derived element defined as Electronics - Books. The derived element values as well as the individual attribute element values are excluded from the subtotal, as shown below.  Category / Metrics / Profit Electronics - Books $9,397,048 Movies $217,710 Music $161,590 **Total** **$379,300**

The steps below show you how to define subtotal behavior for derived elements.

Prerequisites

- A standard report, a report connected to an active Intelligent Cube, a Grid/Graph in a Report Services document, or a stand-alone derived element.
- This procedure also assumes you have created derived elements for the report, Grid/Graph, or stand-alone derived element.
- You have the Derived Elements Editor open. For information on accessing the Derived Elements Editor, see *Accessing the Derived Elements Editor , page 127*.

Copyright © 2024 All Rights Reserved

- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To define subtotal behavior for derived elements

1. In the Derived Elements Editor, select a derived element.
2. Select the **Property** tab.
3. From the **Subtotal behavior** drop-down list, select one of the following options:
   - **Use this element when calculating subtotals**
   - **Use the individual items that make up this element when calculating subtotals**
   - **Do not use this element when calculating subtotals**
4. Click **OK**.

Displaying derived elements or their attribute elements

Derived elements can be displayed on reports and Grid/Graphs as a single, consolidated element with aggregated data, or as the separate attribute elements that are items of the derived element. The simple report shown below is used to demonstrate how you can apply derived elements to subtotals.

Category	Metrics	Profit
Books		$890,696
Electronics		$10,287,744
Movies		$217,710
Music		$161,590

You have the following options for displaying derived elements:

Copyright © 2024 All Rights Reserved

- Consolidate items into one element (default behavior, except for the All Other derived element): The derived element's components are displayed as a single entry with the data of all of its items combined as determined by the derived element.

  For example, in a report with Category and Profit, you create a Group derived element that combines the Books and Electronics attribute elements. By keeping the default option, the derived element is displayed as a single entry on the report, as shown below.

Category	Metrics	Profit
Books and Electronics		$11,178,440
Movies		$217,710
Music		$161,590

- Keep individual items separate (default behavior for the All Other derived element): The attribute elements included as items of a derived element are displayed individually instead of the derived element.

  For example, in a report with Category and Profit, you create a Group derived element that combines the Books and Electronics attribute elements. You choose to keep the individual options separate, which displays the separate attribute elements in the report, as shown below.

Category	Metrics	Profit
Books and Electronics		$11,178,440
Movies		$217,710
Music		$161,590

Category	Metrics	Profit
Books		$890,696
Electronics		$10,287,744
Movies		$217,710
Music		$161,590

  This is the default for an All Other derived element because it displays all attribute elements (that were not included in any other derived elements) as separate attribute elements. This gives the appearance on the report that these attribute elements are not part of any derived element.

  This option is not recommended for Calculation derived elements. The purpose of Calculation derived elements is to use a calculation

Copyright © 2024 All Rights Reserved

expression to combine values of attribute elements or derived elements (or a combination of both). Using this option with a Calculation derived element means that these values are not combined using the calculation expression. This does not reflect the purpose of Calculation derived elements.

The steps below show you how to define whether derived elements are displayed as one consolidated entry, or all of the attribute elements that are items of the derived element are displayed individually.

Prerequisites

- A standard report, a report connected to an active Intelligent Cube, a Grid/Graph in a Report Services document, or a stand-alone derived element.
- This procedure also assumes you have created derived elements for the report, Grid/Graph, or stand-alone derived element.
- You have the Derived Elements Editor open. For information on accessing the Derived Elements Editor, see *Accessing the Derived Elements Editor , page 127*.
- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To display derived elements as consolidated or individual attribute elements

1. In the Derived Elements Editor, select a derived element.
2. Select the **Property** tab.
3. Select one of the following **Element behavior** options:

Copyright © 2024 All Rights Reserved

- **Consolidate items into one element**
- **Keep individual items separate**

4. Click **OK**.

Displaying derived elements and their attribute elements simultaneously

You can analyze attribute elements with derived elements, while also including the attribute element on the report. You can accomplish this by selecting which attribute elements are included in the All Other derived element.

By default, the All Other derived element only collects attribute elements that are not included in derived elements, and includes them on the report as individual attribute elements. For an introduction to the All Other derived element, see *All Other derived element, page 105*.

The simple report below is used as an example of how you can display derived elements and their attribute elements simultaneously.

Category	Metrics	Profit
Books		$890,696
Electronics		$10,287,744
Movies		$217,710
Music		$161,590

You have the following options in the Derived Elements Editor to display derived elements and their attribute elements simultaneously:

- Do not include individual items in the All Other element (default behavior): The attribute elements that are used to define the derived element are excluded from the All Other derived element. This means that the attribute elements are not displayed, only their combined data for the derived element is displayed.

Copyright © 2024 All Rights Reserved

For example, in a report with Category and Profit, you create a Group derived element that combines the Books and Electronics attribute elements. By selecting to keep the default option, the derived element is displayed on the report but the attribute elements that makes up these derived elements are not, as shown below.

Category	Metrics	Profit
Books and Electronics		$11,178,440
Movies		$217,710
Music		$161,590

- Include individual items in the All Other element: The attribute elements that are used to define the derived element are also included in the All Other derived element. This means that the attribute elements are displayed along with the derived elements they are a part of.

  For example, in a report with Category and Profit, you create a Group derived element that combines the Books and Electronics attribute elements. You select to include the Books attribute element and the Electronics attribute element in the All Other derived element. This displays the derived element, as well as the individual Books and Electronics attribute elements, as shown below.

Category	Metrics	Profit
Books and Electronics		$11,178,440
Books		$890,696
Electronics		$10,287,744
Movies		$217,710
Music		$161,590

  For an attribute element to be included in the All Other derived element, all derived elements that the attribute element is included in must be defined to include their attribute elements in the All Other derived element. This can cause unexpected behavior when an attribute element is included in multiple derived elements.

  For example, in a report with Category and Profit, you create a Group derived element that combines the Books and Electronics attribute

Copyright © 2024 All Rights Reserved

elements. You create a second Group derived element that combines Electronics and Movies. The report is shown below.

Category	Metrics	Profit
Books and Electronics		$11,178,440
Electronics and Movies		$10,505,454
Music		$161,590

You select to include the attribute elements of the Books and Electronics derived element in the All Other derived element. Notice that Books is displayed but Electronics is not displayed, as shown below.

Category	Metrics	Profit
Books and Electronics		$11,178,440
Electronics and Movies		$10,505,454
Books		$890,696
Music		$161,590

This is because the Electronics attribute element is also a part of the Electronics and Movies derived element, which is defined to exclude its attribute elements in the All Other calculation. To display the Electronics attribute element along with both derived elements, you must define both derived elements to include their attribute elements in the All Other derived element.

The steps below show how to define whether derived elements are displayed with their attribute elements simultaneously.

Prerequisites

- A standard report, a report connected to an active Intelligent Cube, a Grid/Graph in a Report Services document, or a stand-alone derived element.
- This procedure also assumes you have created derived elements for the report, Grid/Graph, or stand-alone derived element.

Copyright © 2024 All Rights Reserved

- You have the Derived Elements Editor open. For information on accessing the Derived Elements Editor, see *Accessing the Derived Elements Editor , page 127*.
- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

To include or exclude attribute elements in an All Other derived element

1. In the Derived Elements Editor, select a derived element.
2. Select the **Property** tab.
3. Select one of the following **Apply element to all other calculation** options:
   - **Do not include individual items in the All Other element**
   - **Include individual items in the All Other element**
4. Click **OK**.

Formatting derived elements

Formatting derived elements lets you highlight important or unique data as well as correctly reflect value and character formats for your derived elements. For example, you may require the values of a derived element to be displayed as a percentage.

Standard formatting of attributes and metrics lets you format all data for an attribute or metric. Formatting an attribute applies formatting to the attribute name and all of its attribute elements. Formatting a metric applies formatting to the metric name and all of its values across all attribute elements.

Formatting derived elements extends these features to allow you to format individual attribute elements as well format metric values associated with individual attribute elements.

Copyright © 2024 All Rights Reserved

For example, the report shown below has derived elements created for the Region attribute.

Region	Category / Metrics	Profit
Total Profit	Books	$890,696
	Electronics	$10,287,744
	Movies	$217,710
	Music	$161,590
Average Profit	Books	$111,337
	Electronics	$1,285,968
	Movies	$27,214
	Music	$20,199
Greatest Regional Profit	Books	$455,964
	Electronics	$3,933,032
	Movies	$60,787
	Music	$78,357
Greatest Regional Profit % Contribution	Books	51.19%
	Electronics	38.23%
	Movies	27.92%
	Music	48.49%

A procedure to create the report shown above is provided in *Using calculations to create derived elements, page 140*.

Notice that the derived element Greatest Regional Profit % Contribution uses derived element formatting to apply a percentage format to the profit values for the derived element. If you used metric formatting to apply a percentage format to profit values, the percentage format applies to all profit values across all derived elements. But this formats data incorrectly for the derived elements Total Profit, Average Profit, and Greatest Regional Profit.

In addition to applying formatting to metric values to correctly represent data for different derived elements, you can also format the names of each derived element individually. This gives you more flexibility than formatting attributes, which applies formatting to all the attribute elements. This is shown in the same sample report below with additional formatting on individual derived elements, and formatting on the Category attribute.

Copyright © 2024 All Rights Reserved

Region	Category (Metrics)	Profit
Total Profit	Books	$890,696
	Electronics	$10,287,744
	Movies	$217,710
	Music	$161,590
Average Profit	Books	$111,337
	Electronics	$1,285,968
	Movies	$27,214
	Music	$20,199
Greatest Regional Profit	Books	$455,964
	Electronics	$3,933,032
	Movies	$60,787
	Music	$78,357
Greatest Regional Profit % Contribution	Books	51.19%
	Electronics	38.23%
	Movies	27.92%
	Music	48.49%

Notice that each derived element name has a different format but the attribute elements for the Category attribute all share the same format.

Formatting derived elements is only available from the Derived Elements Editor. The procedure below describes how to format derived elements.

Prerequisites

- A standard report, a report connected to an active Intelligent Cube, a Grid/Graph in a Report Services document, or a stand-alone derived element.
- This procedure also assumes you have created derived elements for the report, Grid/Graph, or stand-alone derived element.
- You have the Derived Elements Editor open. For information on accessing the Derived Elements Editor, see *Accessing the Derived Elements Editor , page 127*.
- You need the Define Derived Elements (Developer) and/or the Web Define Derived Elements (Web) privileges. These privileges are part of OLAP Services.

Copyright © 2024 All Rights Reserved

To format derived elements

1. In the Derived Elements Editor, select a derived element.
2. From the toolbar, from the **Change Group** drop-down list, point to **Format**, and then select from the following options:

   - **Values**: Formatting the values of a derived element applies formatting to all metric values associated with a derived element. You can use this to apply value formats for derived elements that use formulas that return different types of data such as percentages, fractions, or other unique formats. In the report shown below, the derived element values for the various derived elements are highlighted with a solid red box, for the purpose of this example.

     Be aware that formatting a derived element's values only applies to the metric values associated with the derived element. For example, the first four profit values are associated with the Total Profit derived element.

   - **Headers**: Formatting the headers of a derived element applies formatting to the derived element name. You can use this to highlight or visually distinguish the various derived elements on the report.

     In the report shown below, the derived element headers for the various derived elements are highlighted with a dashed black box, for the purpose of this example. Be aware that formatting a derived element's header only applies to the derived element you are

Copyright © 2024 All Rights Reserved

formatting, not all derived elements for an attribute.

Region	Category (Metrics)	Profit
Total Profit	Books	$890,696
	Electronics	$10,287,744
	Movies	$217,710
	Music	$161,590
Average Profit	Books	$111,337
	Electronics	$1,285,968
	Movies	$27,214
	Music	$20,199
Greatest Regional Profit	Books	$455,964
	Electronics	$3,933,032
	Movies	$60,787
	Music	$78,357
Greatest Regional Profit % Contribution	Books	51.19%
	Electronics	38.23%
	Movies	27.92%
	Music	48.49%

The Format Cells dialog box opens.

3. Apply number, alignment, font, border, and background formatting for the derived element values or headers. For steps to use the Format Cells dialog box, click the **Help** on the Format Cells dialog box.
4. When you are finished formatting derived elements headers or values, click **OK**. You are returned to the Derived Elements Editor.
5. Click **OK**.

Interaction with other reporting features

The derived elements feature is just one of the many reporting and analysis features available with MicroStrategy. How you use derived elements in combination with other reporting and analysis features can affect the behavior and results of your reports and documents. For information on how derived elements interact with other MicroStrategy features, see the following sections:

Copyright © 2024 All Rights Reserved

- *Derived element interactions with view filters, page 165*
- *Derived element interactions with derived metrics, page 165*
- *Derived element interactions with page-by, page 168*
- *Derived element interactions with thresholds, page 170*
- *Derived elements and drilling on reports, page 172*
- *Derived element interactions with subtotals, page 174*
- *Derived elements in Report Services documents with multiple datasets, page 174*
- *Derived elements in Report Services documents with multiple datasets, page 174*

Derived element interactions with view filters

View filters restrict the amount of data displayed on the report, providing you with a different view of the data. For information on the full set of view filter functionality, see *Chapter 7, View Filters*.

View filters restrict the same data which is used by derived elements to analyze and display data to meet your reporting needs. Since both of these OLAP Services features use the same data, it is important to understand how the two features can be used together to return desired report results. For information on view filter effects on derived elements and other reporting features, see *View filter effects on reporting features, page 239*.

Derived element interactions with derived metrics

Most derived metrics are represented correctly when used with derived elements. However, some derived metrics can be calculated incorrectly when using derived elements, due to the order in which derived elements and derived metrics are calculated.

For example, a derived metric that uses a division, such as `Profit/Revenue`, is calculated incorrectly when using derived elements. This is because, by default, derived metrics are evaluated before derived

Copyright © 2024 All Rights Reserved

elements. You must change the evaluation order to evaluate derived elements before this type of derived metric.

For example, the report shown below includes the Region attribute with East Coast, West Coast, and Central and South derived elements defined on it. The metrics are Revenue, Profit, and Profit Margin.

Region	Metrics	Revenue	Profit	Profit Margin
East Coast		$37,694,553	$6,507,446	50.56%
West Coast		$14,595,166	$2,435,069	32.81%
Central and South		$12,495,474	$2,065,833	33.26%
Web		$3,581,277	$549,392	15.34%

Profit Margin is a derived metric, with the definition `Profit/Revenue`. Notice that the values for Profit Margin are much higher than they should be for the derived elements, but the value for the Web attribute element is correct. This is because Profit Margin is calculated for the individual attribute elements, before the attribute elements are combined to display the derived elements.

By evaluating the derived elements before the Profit Margin derived metric, the correct Profit Margin values are displayed for all derived elements and attribute elements.

Region	Metrics	Revenue	Profit	Profit Margin
East Coast		$37,694,553	$6,507,446	17.26%
West Coast		$14,595,166	$2,435,069	16.68%
Central and South		$12,495,474	$2,065,833	16.53%
Web		$3,581,277	$549,392	15.34%

To modify the evaluation order of derived elements in reports

1. In Developer, log in to a project that includes a report with derived elements and derived metrics.

Copyright © 2024 All Rights Reserved

2. Browse to and right-click a report, and select **Run**. The report is executed in the Report Editor.
3. Choose **Data** > **Report Data Options**. The Report Data Options dialog box opens.
4. Expand the **Calculations** category, and then select **Evaluation Order**.
5. Clear the **Use default evaluation order** check box. This displays all objects on the report that can have their evaluation order modified.
6. In the **Evaluation Order** column, select the evaluation orders for each object in the report.

   Objects are evaluated from lowest evaluation order number to highest. However, if you define an object to use Default as its evaluation order, it may be evaluated before an object with an evaluation order of `1`.

   For example, if you define a derived element with an evaluation order of `1` and a derived metric with an evaluation order of Default, the derived metric is evaluated first. This is because derived metrics are evaluated before derived elements by default. To evaluate a derived element before a derived metric, define the derived metric to have a higher evaluation order than the derived element.
7. Click **OK**. The changes to the evaluation order are displayed in the report. To view the evaluation order of the various objects on the report, display the report in SQL View.

To modify the evaluation order of derived elements in Grid/Graphs in Report Services documents

1. In Developer, log in to a project that includes a report with derived elements.
2. Browse to and right-click a document, and select **Edit**. The document opens in the Document Editor.

Copyright © 2024 All Rights Reserved

3. Double-click the Grid/Graph that contains the derived element to enter Edit mode.
4. Choose **Data** > **Grid Data Options**. The Grid Data Options dialog box opens.
5. Expand the **Calculations** category, and then select **Evaluation Order**.
6. Clear the **Use default evaluation order** check box. This displays all objects on the report that can have their evaluation order modified.
7. In the **Evaluation Order** column, select the evaluation orders for each object in the report.

   Objects are evaluated from lowest evaluation order number to highest. However, if you define an object to use Default as its evaluation order, it may be evaluated before an object with an evaluation order of `1`.

   For example, if you define a derived element with an evaluation order of `1` and a derived metric with an evaluation order of Default, the derived metric is evaluated first. This is because derived metrics are evaluated before derived elements by default. To evaluate a derived element before a derived metric, define the derived metric to have a higher evaluation order than the derived element.
8. Click **OK**.
9. To view the updated Grid/Graph, run the document.

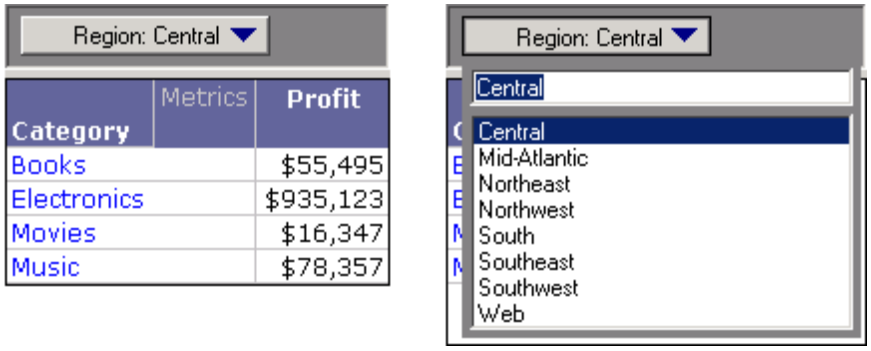
Derived element interactions with page-by

To group data into subsets, you can use the page-by feature. The subsets you separate your business data into are called pages, and you then page your way through the report, viewing one data subset at a time. Page-by makes viewing a large report easier than scrolling through long lists of data.

Attributes are one of the most common objects included in the page-by area of a report. When an attribute is included in the page-by area, you can select which attribute element to view data for.

Copyright © 2024 All Rights Reserved

For example, the report below on the left includes the attributes Region and Category, along with the Profit metric. The Region attribute is included in the page-by area, and Central is selected by default. The report below on the right displays all the attribute elements available in the page-by area.



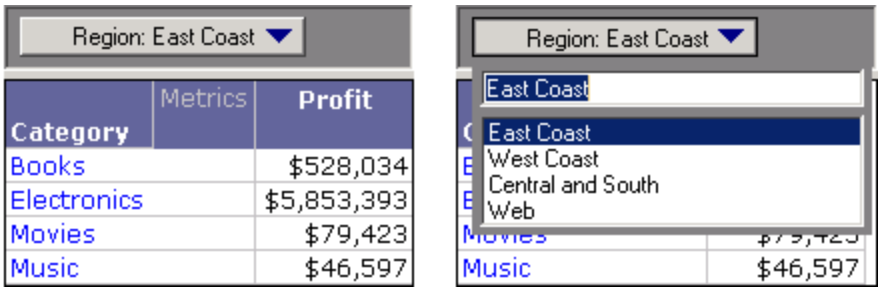
If you create derived elements for an attribute included in the page-by area, the derived elements are available from the page-by field to display their associated data.

For example, the report below shows East Coast, West Coast, and Central and South derived elements based on the Region attribute.

Region	Category	Metrics	Profit
East Coast	Books		$528,034
	Electronics		$5,853,393
	Movies		$79,423
	Music		$46,597
West Coast	Books		$61,310
	Electronics		$2,290,530
	Movies		$66,426
	Music		$16,803
Central and South	Books		$272,158
	Electronics		$1,674,186
	Movies		$29,251
	Music		$90,238
Web	Books		$29,195
	Electronics		$469,635
	Movies		$42,611
	Music		$7,952

Copyright © 2024 All Rights Reserved

If Region is moved to the page-by area, the derived elements are available for selection from the page-by field, along with the Web attribute element, as shown below.



You must move attributes from the page-by area to the grid of the report to create or modify derived elements for the attribute. You can then move the attribute back to the page-by area once all derived element modifications are complete.

Page-by is covered in greater detail in the Basic Reporting Help.

Derived element interactions with thresholds

Thresholds highlight particular data in a report by displaying special cell formats, symbols, images, or replacement text.

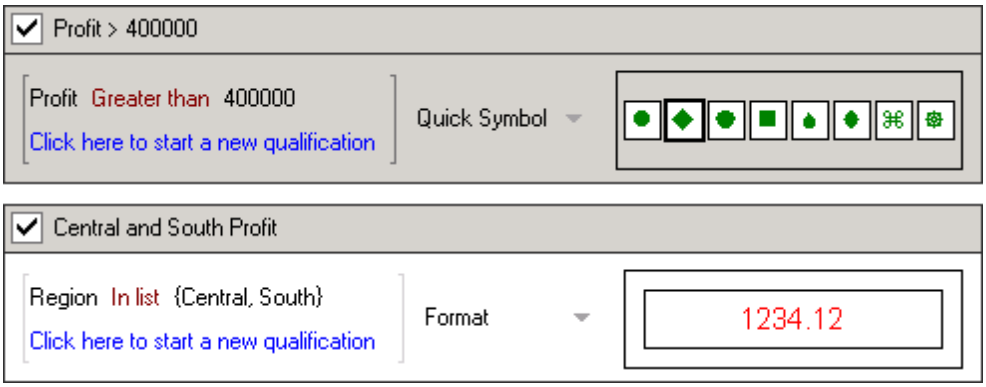
Thresholds are created using conditions on attributes or metrics. Thresholds created using conditions on individual attributes are not displayed for derived element data. However, the more commonly used thresholds created using conditions on metrics are applied to derived element data.

For example, the report below shows East Coast, West Coast, and Central and South derived elements based on the Region attribute.

Copyright © 2024 All Rights Reserved

Region	Category	Metrics	Profit
East Coast	Books		$528,034
	Electronics		$5,853,393
	Movies		$79,423
	Music		$46,597
West Coast	Books		$61,310
	Electronics		$2,290,530
	Movies		$66,426
	Music		$16,803
Central and South	Books		$272,158
	Electronics		$1,674,186
	Movies		$29,251
	Music		$90,238
Web	Books		$29,195
	Electronics		$469,635
	Movies		$42,611
	Music		$7,952

You create two thresholds, the first defined on a metric and the second defined on an attribute, as shown in the following image.



When the report is executed, only the Profit > 400000 threshold is displayed, as shown below.

Copyright © 2024 All Rights Reserved

Region	Category (Metrics)	Profit
East Coast	Books	◆
	Electronics	◆
	Movies	$79,423
	Music	$46,597
West Coast	Books	$61,310
	Electronics	◆
	Movies	$66,426
	Music	$16,803
Central and South	Books	$272,158
	Electronics	◆
	Movies	$29,251
	Music	$90,238
Web	Books	$29,195
	Electronics	◆
	Movies	$42,611
	Music	$7,952

The Profit > 400000 threshold uses the derived element data to determine when to display the defined formatting.

However, the formatting for the Central and South threshold is not displayed. This is due to the fact that the separate Central and South attribute elements are not on the report, and are instead replaced with a Central and South derived element. While you cannot display this type of threshold formatting for derived elements, you can still apply formatting to the data for each derived element, as described in *Formatting derived elements, page 160*.

Thresholds are covered in greater detail in the Basic Reporting Help.

Derived elements and drilling on reports

Drilling allows you to view displayed report data at levels other than that returned in the original grid or graph report. You can investigate the data in your report quickly and easily with the help of drilling. It allows you to execute another report based on the original report to get more detailed or supplemental information.

When you drill on attribute elements in a report, the resulting report restricts the results to data only for the attribute elements used when drilling. For

Copyright © 2024 All Rights Reserved

example, if you drill from the Year 2007 attribute element down to Quarter, the resulting report only includes quarters that are within 2007.

The same standard applies to drilling on derived elements. Drilling on a derived element restricts the resulting report to only data for the attribute elements used to define the derived element.

For example, the report below shows East Coast, West Coast, and Central and South derived elements based on the Region attribute.

Region	Category (Metrics)	Profit
East Coast	Books	$528,034
	Electronics	$5,853,393
	Movies	$79,423
	Music	$46,597
West Coast	Books	$61,310
	Electronics	$2,290,530
	Movies	$66,426
	Music	$16,803
Central and South	Books	$272,158
	Electronics	$1,674,186
	Movies	$29,251
	Music	$90,238
Web	Books	$29,195
	Electronics	$469,635
	Movies	$42,611
	Music	$7,952

If you drill down from the East Coast derived element to Call Center, the resulting report shown below returns data for Call Centers within the East Coast regions (Northeast, Mid-Atlantic, Southeast).

Copyright © 2024 All Rights Reserved

Call Center	Category (Metrics)	Profit
Atlanta	Books	$11,623
	Electronics	$197,187
	Movies	$17,766
	Music	$3,167
Washington, DC	Books	$32,282
	Electronics	$2,760,258
	Movies	$10,024
	Music	$9,638
Miami	Books	$13,985
	Electronics	$222,660
	Movies	$19,938
	Music	$3,797
Boston	Books	$80,969
	Electronics	$268,450
	Movies	$4,549
	Music	$4,356
New York	Books	$374,994
	Electronics	$1,232,064
	Movies	$22,818
	Music	$21,539
Charleston	Books	$14,179
	Electronics	$1,172,774
	Movies	$4,327
	Music	$4,099

Drilling is covered in greater detail in the Basic Reporting Help.

Derived element interactions with subtotals

You can define how derived elements are applied to subtotals, which is described in *Applying derived element values to subtotals, page 151*.

Derived elements in Report Services documents with multiple datasets

A Report Services document may contain multiple datasets. The dataset reports that you use may contain derived elements on the same attribute. If the attribute is added to a Grid/Graph in the document, derived elements from all reports are displayed on the Grid/Graph, as well as any elements that are not part of the derived elements, as described in the example below.

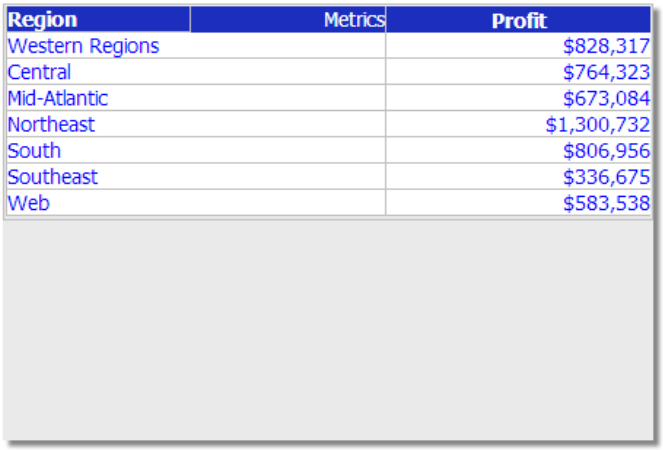
Copyright © 2024 All Rights Reserved

Example: derived elements in documents with multiple datasets

A document uses two reports as datasets, defined as follows:

- Dataset 1 is a Regional Profit report, containing the Region attribute and the Profit metric. The derived element Western Regions is defined on Region, with the following elements:
  - Northwest
  - Southwest

  The dataset is shown in the image below.

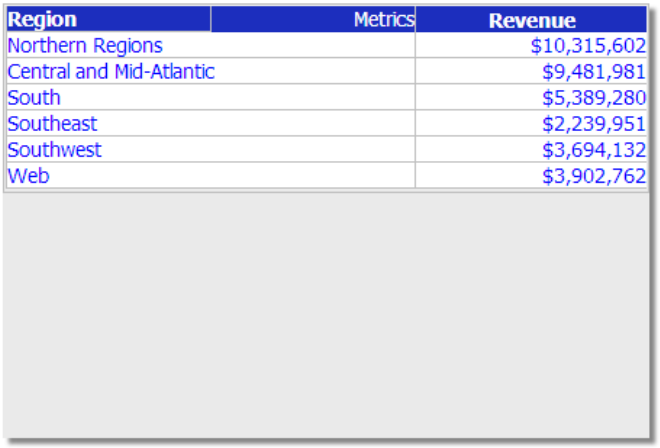


Region	Metrics	Profit
Western Regions		$828,317
Central		$764,323
Mid-Atlantic		$673,084
Northeast		$1,300,732
South		$806,956
Southeast		$336,675
Web		$583,538

- Dataset 2 is a Regional Revenue report, containing the Region attribute and the Revenue metric. The following derived elements are defined on Region:
  - Northern Regions, containing the Northwest and Northeast elements.
  - Central and Mid-Atlantic, containing the Central and Mid-Atlantic elements.

  The dataset is shown in the image below.

Copyright © 2024 All Rights Reserved



Region	Metrics	Revenue
Northern Regions		$10,315,602
Central and Mid-Atlantic		$9,481,981
South		$5,389,280
Southeast		$2,239,951
Southwest		$3,694,132
Web		$3,902,762

In the document, a Grid/Graph is created, and the following objects are added to it:

- Region
- Profit
- Revenue

When the document is executed, the data from both datasets is combined, and derived elements from both datasets are displayed on the combined Grid/Graph, as shown in the image below:

Region	Metrics	Profit	Revenue
Western Regions		$828,317	$5,455,319
Northern Regions		$1,567,717	$10,315,602
Central and Mid-Atlantic		$1,437,406	$9,481,981
South		$806,956	$5,389,280
Southeast		$336,675	$2,239,951
Web		$583,538	$3,902,762

For attribute elements that are not part of a derived element, such as South and Southeast, data from both datasets is combined and displayed.

If your datasets contain derived elements with the same name, and containing the same attribute elements, the data for the derived elements is combined and displayed in a single row.

For detailed instructions to use documents with multiple datasets, refer to the Document Creation Help.

Copyright © 2024 All Rights Reserved

Derived Metrics

Copyright © 2024 All Rights Reserved

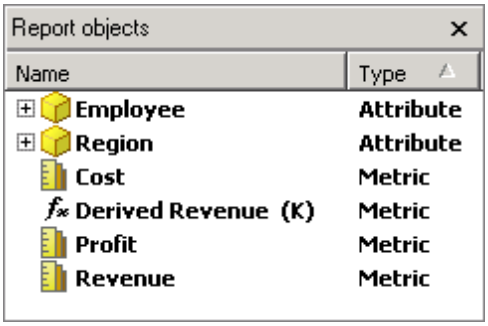
A derived metric is a calculation based on the data included in the report definition. You can use derived metrics to perform column math, for example, calculations on metrics included in the report definition, without regenerating or re-executing SQL against the data warehouse.

Derived metrics are easy to create and can be made on the fly when you are viewing a report. A typical case is to create a derived metric to perform calculations between columns of metric data. For example, a derived metrics can subtract the data of one metric by the data of another metric (Metric1 - Metric2) to obtain a new metric calculation.

To see how derived metrics can be used in reports, consider the **Derived Metric** report from the MicroStrategy Tutorial project. The report includes a derived metric that is defined as `Revenue/1000`, which returns the Revenue values in thousands of dollars, as shown below.

Region	Employee	Metrics	Revenue	Cost	Profit	Derived Revenue (K)
Central	Ellerkamp Nancy		$1,169,245	$987,540	$181,705	**$1,169**
	Gale Loren		$2,262,146	$1,907,140	$355,006	**$2,262**
	Torrison Mary		$2,364,993	$1,992,733	$372,260	**$2,365**
	Zemlicka George		$1,116,549	$940,199	$176,349	**$1,117**

Notice that in the Report Objects pane to the left of the report (and shown below), the derived metric is preceded by an ***fx*** symbol, instead of the usual metric symbol, meaning this is a new metric based on the existing metrics in the report.



Copyright © 2024 All Rights Reserved

Since derived metrics are created within a report, they can only be used for the report in which they are created. Derived metrics cannot be saved as individual objects in the project, and therefore cannot be applied to other reports in the project.

This section discusses the following topics on derived metrics:

- *Creating a derived metric, page 179*
- *Editing derived metrics, page 198*
- *Formatting derived metrics, page 198*
- *Deleting derived metrics, page 199*
- *View filter effects on derived metrics, page 199*
- *Derived element effects on derived metrics, page 200*

Creating a derived metric

Derived metrics can be created in the following:

- As part of the design or analysis of a report. This allows you to create these metrics while focusing on a given report.

  You can create a derived metric in any report. This includes reports connected to an Intelligent Cube, reports returning information directly from a data warehouse, Freeform SQL reports returning information from an Excel spreadsheet, and so on.

- In Report Services documents. You can define a derived metric based on one or more metrics from the document's datasets. For instructions to create derived metrics in Report Services documents, refer to the Document Creation Help.

- In Visual Insight dashboards. You can quickly and easily add simple derived metrics to a Visual Insight (VI) dashboard on the fly. For

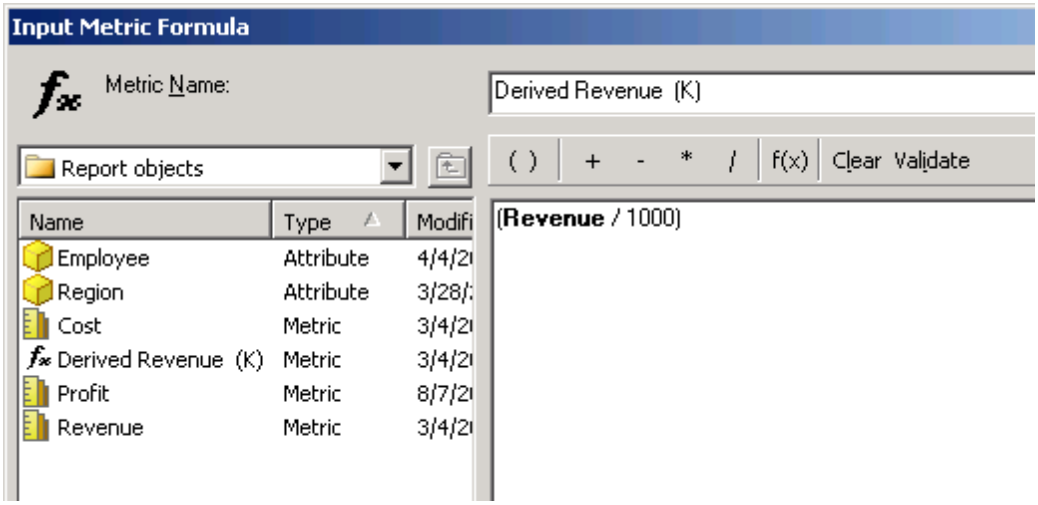
Copyright © 2024 All Rights Reserved

instructions to create derived metrics in VI dashboards, see the MicroStrategy Web Help.

Best practices

Follow the guidelines below when creating derived metrics:

- In reports, you can define derived metrics with objects in the Report Objects pane. The Report Objects are the components included in the report definition, even if they are not displayed on the report grid.
- A derived metric can be defined with the metrics in the report definition. The Input Metric Formula dialog box where you create derived metrics allows you to choose only from objects included in the report definition, as shown below.



- Attributes included in the report definition are also available to use in the definition of a derived metric. If you use an attribute as part of the metric definition, the metric calculation requires new report SQL to be executed against the data warehouse. This re-execution is not required for derived metrics that only use metrics in their definitions.
- You can use one or more functions or operators in the formula of the derived metric. Click the ***fx*** button to access available functions and operators.

Copyright © 2024 All Rights Reserved

- You can use numeric prompts in the formula of the derived metric, which allows users to determine part of the value of the metric. For example, if the value of your metric depends on the current tax rate, you can prompt users to type the current tax rate.

  For steps to create prompts, see the Basic Reporting Help.

- You can change the level at which a derived metric is calculated. For example, the derived metric `sum(M1) {Attribute1}` is calculated at the Attribute1 level. For information on metric levels, see the Advanced Reporting Help.

- Any user can modify a derived metric after report execution, since its formula is visible to all users. If a derived metric should not be modified by end users, create the metric in the Metric Editor and add it to the report as a normal metric.

- Transformation objects cannot be used with derived metrics because they require SQL to be re-executed against the data warehouse.

- View filters can filter the results of a derived metric. A view filter is an additional filter applied in memory to the report results to restrict the amount of data displayed on the report. For more information on view filters, see *Chapter 7, View Filters*.

- A derived metric is dependent on any report objects that are included in the derived metric's definition. Because of this dependency, you cannot remove an object from the report that is used in a derived metric definition.

  If you try to remove an object from the report, a message is displayed that indicates you cannot remove the object because it is being used by the derived metric. You can however move an object off the report grid so that it only appears in the Report Objects pane. This allows you to hide the object from the report grid and still support any derived metrics that are dependent on it.

Copyright © 2024 All Rights Reserved

- In Report Services documents that use multiple datasets, you can create derived metrics that use metrics from different datasets. If you do so, note the following:
  - The metrics you use may be calculated at different levels, depending on the definition of the datasets. For example, one dataset contains the Subcategory attribute and the Revenue metric, and another dataset contains the Category attribute and the Profit metric. If you create a derived metric based on Revenue and Profit, note that Revenue is calculated at the Subcategory level, and Profit is calculated at the Category level. This can be useful to create percent-to-total metrics.
  - If the derived metric is to be used in a data field, it is recommended that you use a calculated expression instead.

  For information and examples of using derived metrics in documents with multiple datasets, refer to the Document Creation Help.

You can create derived metrics with the following methods described in this section:

- *Quickly creating a derived metric in Web, page 183*: You can create a derived metric based on often-used functions, such as Average, by using the Insert Metric feature in Web.
- *Creating a derived metric using the Input Metric Formula dialog box, page 184*: You can create any type of derived metric by defining derived metric expressions using the Input Metric Formula dialog box.
- *Using rank and percent-to-total metric analysis , page 189*: You can quickly create derived metrics that display the percent in relation to a selected total of each item affected by the metric or display a ranking number to the metric values for a given attribute. These can be quickly created using shortcut metrics.

Copyright © 2024 All Rights Reserved

Quickly creating a derived metric in Web

You can quickly create a derived metric based on an attribute or metric, using the Insert Metric function in MicroStrategy Web. For example, Customer is an attribute, and you want to create a metric that counts the number of distinct customers.

You can use the Insert Metric function to create a derived metric that uses one of the following functions:

- Average
- Count
- Maximum
- Minimum
- Sum

For steps to create a derived metric using any other function, see *Creating a derived metric using the Input Metric Formula dialog box, page 184*.

To quickly create a derived metric in Web

1. Open a report or document in MicroStrategy Web.
2. In the Report Objects or Dataset Objects pane, as applicable, right-click the object that you want to base the metric on. For example, if you want to create a metric that counts the number of customers, right-click **Customer**.
3. Select **Insert Metric**, and click the function that you want to use for the metric. The metric is created, and is shown in the Report Objects or Dataset Objects pane, as applicable.
4. Drag the metric on to the report or document, as applicable.
5. Click **Save**.

Copyright © 2024 All Rights Reserved

Creating a derived metric using the Input Metric Formula dialog box

The Input Metric Formula dialog box allows you to use functions, operators, and report objects to create derived metrics. You can create any type of derived metric by defining derived metric expressions using the Input Metric Formula dialog box. This includes shortcut metrics that are described in *Using rank and percent-to-total metric analysis , page 189*.

The procedure below describes the high-level steps for creating a derived metric with the Input Metric Formula dialog box. For an example of creating a derived metric, see *Example: Average profit per customer with transactions, page 186*.

Prerequisites

- You need the Create Derived Metrics (Developer) and/or the Web Create Derived Metrics and Derived Attributes (Web) privileges. These privileges are part of OLAP Services.

To create a derived metric with the Input Metric Formula dialog box

For details on each option for any of the steps below, click **Help**.

1. Open a report in MicroStrategy Developer or Web.

   If you are running MicroStrategy Developer on Windows for the first time, run it as an administrator.

   Right-click the program icon and select **Run as Administrator**.

   This is necessary in order to properly set the Windows registry keys. For more information, see KB43491.

2. Open the Input Metric Formula dialog box to create a new metric by performing one of the following steps:

Copyright © 2024 All Rights Reserved

- In Developer: Choose **Insert** > **New Metric**.
- In Web: Choose **Data** > **Insert New Metric**.

3. The pane on the left displays the Report Objects which shows the components (attributes, attribute forms, metrics, custom groups, consolidations, and so on) included in the report, even if the components are not displayed in the report grid.
   - In Developer: Double-click or drag-and-drop an object to use it to define the derived metric.
   - In Web: Select an object and click the arrow to use it to define the derived metric. This moves the object to the window on the right.
4. Add functions and operators by typing their syntax or characters. You can also click the ***fx*** button to open the Insert Function Wizard, which guides you through adding functions and operators.
5. Continue to add report components, functions, operators, constant values, and other valid metric formula objects to complete your formula.
6. To add the level at which to calculate the metric, enclose the metric formula in parentheses. Type the attribute name between curly braces `{}` after the metric formula. If the attribute name contains a space, enclose the name within brackets `[]`. For example, `([Unit Cost] * [Units Sold]) {[Customer Name]}`, where Unit Cost and Units Sold are metrics and Customer Name is an attribute. This is a valid expression.

   For more information on metric levels, see the Advanced Reporting Help.
7. After you have created the expression, you can determine whether the expression is valid by performing one of the following steps:
   - In Developer: Click **Validate**. An error is displayed if the expression is invalid.

Copyright © 2024 All Rights Reserved

- In Web: Click **Apply**. An error is displayed if the expression is invalid.

8. In the **Metric Name** (Developer) or **Name** (Web) field, enter a name for the new metric.

   When naming a MicroStrategy object, you must follow the naming convention rules for your particular database platform. Using a word reserved by your database platform can result in an error when the report is executed. Refer to your database documentation for a list of these database-reserved words.

9. Click **OK**. The Input Metric Formula dialog box closes and the derived metric is added to the report.

To format the values or headers of a derived metric, see *Formatting derived metrics, page 198*.

Example: Average profit per customer with transactions

In this example, a derived metric is created for the **Customer Profitability - By Tenure** report from the Customer Analysis Module. This report provides monthly trends in customer information based on tenure, as shown below.

Copyright © 2024 All Rights Reserved

Month	Tenure Ranges (Metrics)	Active Customers	Customers with Transactions	Revenue	Profit	Margin
Oct 2007	1 Month	38	30	$15,197.75	$988.91	6.51%
	2 Months	40	28	$6,181.77	$341.72	5.53%
	3 Months	37	30	$17,201.78	$1,292.69	7.51%
	4 Months	40	33	$14,241.71	$1,203.63	8.45%
	5 Months	38	27	$9,305.84	$795.54	8.55%
	> 5 Months	1,106	865	$273,794.53	$17,493.32	6.39%
Nov 2007	1 Month	40	27	$11,447.74	$900.94	7.87%
	2 Months	37	29	$8,785.75	$357.27	**4.07%**
	3 Months	39	28	$5,001.27	$291.32	5.82%
	4 Months	36	31	$5,801.76	$347.67	5.99%
	5 Months	39	32	$10,344.24	$505.92	**4.89%**
	> 5 Months	1,142	862	$279,610.09	$16,257.96	5.81%
Dec 2007	1 Month	40	35	$10,554.66	$581.89	5.51%
	2 Months	38	34	$17,786.73	$1,157.86	6.51%
	3 Months	37	23	$10,071.25	$775.26	7.70%
	4 Months	37	33	$20,653.22	$1,169.74	5.66%
	5 Months	36	30	$10,877.18	$540.68	**4.97%**
	> 5 Months	1,175	866	$310,550.97	$19,642.77	6.33%

A derived metric is created to display the average profit generated per customer with transactions. The procedure below describes how to create and format this derived metric for the report shown above.

To create an average profit per customer with transactions derived metric

1. In MicroStrategy Developer, log in to the Customer Analysis Module.
2. Browse to the **Customer Profitability - By Tenure** report, right-click the report, and then select **Run Report**. The **Choose from all elements of 'Quarter'** prompt opens.
3. Keep the default prompt answer of **2007 Q4** and click **Finish**. The Customer Profitability - By Tenure report opens in Grid View.
4. Choose **Insert** > **New Metric**. The Input Metric Formula dialog box opens.
5. In the **Metric Name** field, type **Average Profit Per Customer with Transactions**.

Copyright © 2024 All Rights Reserved

To create the expression for the derived metric

You create this expression using the following steps:

```
(Profit / [Customers with Transactions])
```

1. Drag-and-drop the **Profit** metric from the list of Report Objects in the left pane to the metric definition area on the right.
2. Click the division sign (/) on the operators and functions toolbar. The division sign should be inserted to the right of the Profit metric.
3. Drag-and-drop the **Customer with Transactions** metric from the list of Report Objects to the metric definition area on the right. It should be placed on the right side of the division sign (/).
4. Click **Validate**. A green check is displayed in the right-hand corner below the metric definition area, indicating that the metric definition is valid.
5. Click **OK** . The new metric column for the derived metric is displayed in the report.

To format the values of the new derived metric

The metric is included without any value formatting. Follow the steps below to format the metric values to be displayed with a currency format.

1. Right-click the **Average Profit per Customer with Transactions** derived metric, point to **Formatting**, and select **Average Profit per Customer with Transactions Values**. The Format Cells dialog box opens.
2. On the **Number** tab, from the Category pane, select **Currency**.
3. Keep all defaults for the Currency format and click **OK**. You are returned to the report which is shown below.

Copyright © 2024 All Rights Reserved

Month	Tenure Ranges (Metrics)	Active Customers	Customers with Transactions	Revenue	Profit	Margin	Average Profit Per Customer with Transactions
Oct 2007	1 Month	38	30	$15,197.75	$988.91	6.51%	$32.96
	2 Months	40	28	$6,181.77	$341.72	5.53%	$12.20
	3 Months	37	30	$17,201.78	$1,292.69	7.51%	$43.09
	4 Months	40	33	$14,241.71	$1,203.63	8.45%	$36.47
	5 Months	38	27	$9,305.84	$795.54	8.55%	$29.46
	> 5 Months	1,106	865	$273,794.53	$17,493.32	6.39%	$20.22
Nov 2007	1 Month	40	27	$11,447.74	$900.94	7.87%	$33.37
	2 Months	37	29	$8,785.75	$357.27	**4.07%**	$12.32
	3 Months	39	28	$5,001.27	$291.32	5.82%	$10.40
	4 Months	36	31	$5,801.76	$347.67	5.99%	$11.22
	5 Months	39	32	$10,344.24	$505.92	**4.89%**	$15.81
	> 5 Months	1,142	862	$279,610.09	$16,257.96	5.81%	$18.86
Dec 2007	1 Month	40	35	$10,554.66	$581.89	5.51%	$16.63
	2 Months	38	34	$17,786.73	$1,157.86	6.51%	$34.05
	3 Months	37	23	$10,071.25	$775.26	7.70%	$33.71
	4 Months	37	33	$20,653.22	$1,169.74	5.66%	$35.45
	5 Months	36	30	$10,877.18	$540.68	**4.97%**	$18.02
	> 5 Months	1,175	866	$310,550.97	$19,642.77	6.33%	$22.68

Using rank and percent-to-total metric analysis

You can create derived metrics that display the percent in relation to a selected total of each item affected by the metric or display a ranking number to the metric values for a given attribute.

These can be quickly created using shortcut metrics. Shortcut metrics are a set of quick metrics you can create on the fly for a given report, based on the metrics already in the report. Shortcut metrics include:

- Percent-to-total shortcut metrics, which display the percent in relation to a selected total of each item affected by the metric. To create this type of shortcut metric, see *Creating a percent-to-total shortcut metric, page 190*.
- Rank shortcut metrics, which apply a ranking number to the metric values for a given attribute. To create this type of shortcut metric, see *Creating a rank shortcut metric, page 193*.
- Transformation shortcut metrics, which apply offset values, such as "four months ago," to an attribute that you select.

All the shortcut metrics are derived metrics, except for transformation shortcut metrics. Transformation shortcut metrics must be calculated in SQL and are therefore only available to users that have the Modify the List of Report Objects (Use Object Browser) privilege. Because they must be

Copyright © 2024 All Rights Reserved

calculated, the report re-executes to display the new metric. Transformation metrics are not covered in this guide, see the Advanced Reporting Help. Reports are not re-executed when other shortcut metrics are created.

Creating a percent-to-total shortcut metric

A percent-to-total shortcut metric displays the percent in relation to a selected total of each item affected by the metric. Use a percent-to-total shortcut metric to show cell-level values as percents of an accumulated row or column total. The metric can also show a total by page, show a total for each value of the attribute, or show the grand total.

For example, a report contains Sales Region, Sales Representative, and the Revenue by Sales Representative metric as shown below.

Sales Region	Sales Representative	Metrics	Revenue by Sales Representative
United States	Alec Berg		$4,355,000
	Bonnie Zastrow		$3,130,000
	Carole Alaniz		$3,160,000
	Danny Leavitt		$2,915,000
	Erin Dull		$2,325,000
	Gerald Blubaugh		$3,310,000
Canada	Jori Oakes		$3,755,000
	Miko Kemna		$3,785,000
	Oral Beason		$4,035,000
	Palmer Buzzerio		$2,885,000
	Anany Harlan		$2,920,000
	Benoit Ward		$2,485,000
Europe	Constance Imes		$3,200,000
	Dome Bachmeier		$1,665,000
	Enzo Kellaway		$3,375,000
	Gordon Cutting		$2,100,000
	Josh Hadley		$2,115,000
	Anthony Vanderwerff		$2,240,000

Along with revenue totals for each sales representative, you can also highlight a sales representative's percent contribution to revenue for their sales region, as well as company-wide. You insert two percent-to-total shortcut metrics to display this information, as shown in the report below:

Copyright © 2024 All Rights Reserved

Sales Region	Sales Representative	Metrics	Revenue by Sales Representative	Percent to Total (Revenue by Sales Representative)	Percent to Grand Total (Revenue by Sales Representative)
United States	Alec Berg		$4,355,000	22.69%	8.10%
	Bonnie Zastrow		$3,130,000	16.31%	5.82%
	Carole Alaniz		$3,160,000	16.46%	5.88%
	Danny Leavitt		$2,915,000	15.19%	5.42%
	Erin Dull		$2,325,000	12.11%	4.33%
	Gerald Blubaugh		$3,310,000	17.24%	6.16%
Canada	Jori Oakes		$3,755,000	18.90%	6.99%
	Miko Kemna		$3,785,000	19.05%	7.04%
	Oral Beason		$4,035,000	20.31%	7.51%
	Palmer Buzzerio		$2,885,000	14.52%	5.37%
	Anany Harlan		$2,920,000	14.70%	5.43%
	Benoit Ward		$2,485,000	12.51%	4.62%
Europe	Constance Imes		$3,200,000	21.78%	5.95%
	Dome Bachmeier		$1,665,000	11.33%	3.10%
	Enzo Kellaway		$3,375,000	22.97%	6.28%
	Gordon Cutting		$2,100,000	14.29%	3.91%
	Josh Hadley		$2,115,000	14.39%	3.93%
	Anthony Vanderwerff		$2,240,000	15.24%	4.17%

This report provides analysis into sales representatives' performances at both the regional and company-wide level.

The following rules apply to percent-to-total shortcut metrics:

- Row and column percent-to-totals refer to the extreme-left and topmost positions, respectively.
- Page percent-to-totals affect all attributes on a page.
- Percent to All -> A1, where A1 is an attribute, indicates that the calculation is performed across all elements of that attribute. An example is percent to all stores.
- If a report does not contain attributes at a given percent-to-total level, the level is unavailable for that report.
- In some cases, two or more percent-to-total calculations at different logical levels yield the same result. For example, Percent-to-Page Total data can be the same as Percent-to-Grand Total data in a single-page report.

Copyright © 2024 All Rights Reserved

- The level of a percent-to-total shortcut metric remains constant once the metric has been calculated; subsequent manipulation of objects on the report does not affect it.

The steps below show you how to create a percent-to-total shortcut metric.

Prerequisites

- You need the Create Derived Metrics (Developer) and/or the Web Create Derived Metrics and Derived Attributes (Web) privileges. These privileges are part of OLAP Services.

To create a percent-to-total shortcut metric

1. Open a report in MicroStrategy Developer or Web.

   If you are running MicroStrategy Developer on Windows for the first time, run it as an administrator.

   Right-click the program icon and select **Run as Administrator**.

   This is necessary in order to properly set the Windows registry keys. For more information, see KB43491.

2. Right-click the metric to see data displayed as percents of a total, point to **Insert** (Developer) or **Insert Metric** (Web), point to **Percent to Total**, and then select the portions of the report for which percent-to-total data is to be calculated. Options are:

   - **Over Rows**: Displays values in each row of the report as percents of an attribute row total. Row percent-to-totals refer to the attribute in the extreme-left position. This option is best suited for reports that display attributes in the rows and metrics in the columns.
   - **Over Columns**: Displays values in each column as percents of an attribute column total. Column percent-to-totals refer to the attribute in the topmost position. This option is best suited for reports that display attributes in the columns and metrics in the rows.

Copyright © 2024 All Rights Reserved

- **Page Total**: Displays all values on a page as percents of that page's total. This option is available only on reports that include an object in the page-by area.
- **Grand Total**: Displays all values in a report as percents of the grand total for that report.
- **Total for each**: Displays all values pertaining to a given report component (an attribute, for example) as percents of the total accumulated for that component.

The report is updated showing your new shortcut metric that displays the percent-to-totals for the components selected. To edit a shortcut metric, see *Editing derived metrics, page 198*. To format the values or headers of a derived metric, see *Formatting derived metrics, page 198*.

Creating a rank shortcut metric

A rank shortcut metric applies a ranking number to the metric values for a given attribute. Use the rank function to show the relative position of a given cell content in relation to other values for a report component. The rank shortcut metric provides break-by options for each attribute on the report.

For example, a report contains Sales Region, Sales Representative, and the Revenue by Sales Representative metric as shown below.

Copyright © 2024 All Rights Reserved

Sales Region	Sales Representative	Metrics	Revenue by Sales Representative
United States	Alec Berg		$4,355,000
	Bonnie Zastrow		$3,130,000
	Carole Alaniz		$3,160,000
	Danny Leavitt		$2,915,000
	Erin Dull		$2,325,000
	Gerald Blubaugh		$3,310,000
Canada	Jori Oakes		$3,755,000
	Miko Kemna		$3,785,000
	Oral Beason		$4,035,000
	Palmer Buzzerio		$2,885,000
	Anany Harlan		$2,920,000
	Benoit Ward		$2,485,000
Europe	Constance Imes		$3,200,000
	Dome Bachmeier		$1,665,000
	Enzo Kellaway		$3,375,000
	Gordon Cutting		$2,100,000
	Josh Hadley		$2,115,000
	Anthony Vanderwerff		$2,240,000

Along with revenue totals for each sales representative, you can also highlight sales representatives' relative performances company-wide. You insert a rank shortcut metric to display this information, as shown in the report below:

Copyright © 2024 All Rights Reserved

Sales Region	Sales Representative	Metrics	Revenue by Sales Representative	Rank (Revenue by Sales Representative)
United States	Alec Berg		$4,355,000	1
	Gerald Blubaugh		$3,310,000	6
	Carole Alaniz		$3,160,000	8
	Bonnie Zastrow		$3,130,000	9
	Danny Leavitt		$2,915,000	11
	Erin Dull		$2,325,000	14
Canada	Oral Beason		$4,035,000	2
	Miko Kemna		$3,785,000	3
	Jori Oakes		$3,755,000	4
	Anany Harlan		$2,920,000	10
	Palmer Buzzerio		$2,885,000	12
	Benoit Ward		$2,485,000	13
Europe	Enzo Kellaway		$3,375,000	5
	Constance Imes		$3,200,000	7
	Anthony Vanderwerff		$2,240,000	15
	Josh Hadley		$2,115,000	16
	Gordon Cutting		$2,100,000	17
	Dome Bachmeier		$1,665,000	18

This report provides analysis into sales representatives' relative performances company-wide. If you use Sales Region as a break-by, employees would be ranked within their sales regions.

By default, a rank shortcut metric ranks values from low to high in ascending order. Therefore, the lowest value has a rank of 1 and the highest value has a rank equal to the total number of items being ranked within a given break-by grouping. You can change the metric to rank in descending order by editing the metric.

Refer to the procedure below for steps to create a rank shortcut metric.

Prerequisites

- You need the Create Derived Metrics (Developer) and/or the Web Create Derived Metrics and Derived Attributes (Web) privileges. These privileges are part of OLAP Services.

Copyright © 2024 All Rights Reserved

To create a rank shortcut metric

1. Open a report in MicroStrategy Developer or Web.
2. Right-click the metric to rank, point to **Insert** (Developer) or **Insert Metric** (Web), point to **Rank**, and then select the break by value from the list. You have the following options:
   - **Break by None**: No break by is used and the rank is calculated accounting for each separate item on the report.
   - **Break by** *Object*: A break by on ***Object*** is used to calculate the rank. For example, a report contains Region, Employee, and the Revenue metric. You select to **Break by Region**. Each employee is ranked within their region; that is, each region has a separate ranking.
3. If you are creating a rank shortcut metric in MicroStrategy Web, you must choose from the following sort orders:
   - **Ascending**: Values are ranked from least to greatest in ascending order. Therefore, the smallest value has a rank of 1 and the largest value has a rank equal to the total number of items being ranked within a given break-by grouping.
   - **Descending**: Values are ranked from greatest to least in descending order. Therefore, the largest value has a rank of 1 and the smallest value has a rank equal to the total number of items being ranked within a given break-by grouping.

   In Developer, by default, a rank shortcut metric ranks values from low to high in ascending order. You can change the metric to rank in descending order by editing the metric, as described in *To change the ranking order of a rank shortcut metric, page 197*.

The report is updated showing your new shortcut metric that displays the ranking for the components selected. To edit a shortcut metric, see *Editing derived metrics, page 198*. To format the values or headers of a derived metric, see *Formatting derived metrics, page 198*.

Copyright © 2024 All Rights Reserved

To change the ranking order of a rank shortcut metric

When creating rank shortcut metrics in Developer, a rank shortcut metric ranks values from low to high in ascending order. You can change the metric to rank in descending order by editing the metric, as described in the steps below.

Prerequisites

- You need the Create Derived Metrics (Developer) and/or the Web Create Derived Metrics and Derived Attributes (Web) privileges. These privileges are part of OLAP Services.

To change the ranking order of a rank shortcut metric

1. In MicroStrategy Developer, open a report that contains a rank shortcut metric.
2. Right-click the rank shortcut metric and select **Edit**. The Input Metric Formula dialog box opens and displays the metric formula.
3. Modify the ranking order by modifying the `ASC=`*ranking_order* parameter, as described below:
   - `ASC=True`: (Default) Values are ranked from least to greatest in ascending order. Therefore, the smallest value has a rank of 1 and the largest value has a rank equal to the total number of items being ranked within a given break-by grouping.
   - `ASC=False`: Values are ranked from greatest to least in descending order. Therefore, the largest value has a rank of 1 and the smallest value has a rank equal to the total number of items being ranked within a given break-by grouping.
4. Click **Validate** to confirm that the metric definition is valid for use by the MicroStrategy engine.
5. Click **OK** to return to the Report Editor or Report Viewer. Any rank ordering changes are displayed in the report.

Copyright © 2024 All Rights Reserved

Editing derived metrics

To change the definition of a derived metric after the metric is applied to a report, right-click the derived metric and select **Edit**. The Input Metric Formula dialog box is displayed. From this dialog box, you can modify the metric's definition as needed. For details on any option, click **Help**. Click **OK** after you finish, and the modified metric definition is calculated and displayed on the report instantly without having to re-execute the report against the data warehouse.

For an example using the Input Metric Formula dialog box, see *Creating a derived metric using the Input Metric Formula dialog box, page 184*.

Formatting derived metrics

The same formatting options for regular metrics are available for derived metrics.

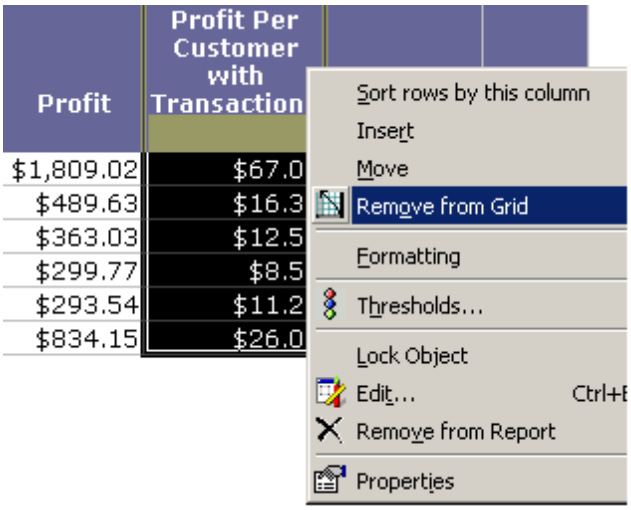
You can format the metric headers and values for a derived metric included on the report grid. To format a derived metric, right-click the derived metric on the grid, point to **Formatting**, and select the derived metrics headers or values. The Format Cells dialog box opens to modify the derived metric formatting. For details on each formatting option, click **Help**. For an example of creating and formatting a derived metric, see *Example: Average profit per customer with transactions, page 186*.

You can also use thresholds with derived metrics in the same ways you can use thresholds with standard metrics created with the Metric Editor. Thresholds are used to create conditional formatting for metric values. To open the Threshold dialog box for a derived metric, right-click the derived metric on the report grid and select **Thresholds**. For more information on thresholds, see the Basic Reporting Help.

Copyright © 2024 All Rights Reserved

Deleting derived metrics

Once a derived metric is created, it becomes part of the report and is saved with the other objects in the report. If you do not want the derived metric to be on the grid but want to keep it in the report definition, you can right-click it in the grid and select **Remove from Grid**. This action is shown in the image below. The metric is taken off the grid and is no longer bold in the Report Objects pane.



If you do not want the derived metric to be saved as part of the report, you can delete it from the report completely by right-clicking the derived metric and selecting **Remove from Report**.

If the derived metric is included in the definition of another derived metric on the report, an error message is displayed. To delete the derived metric from the report, you must remove or modify any derived metrics that are dependent on the derived metric you are attempting to delete.

View filter effects on derived metrics

Derived metrics calculate their values using the results in a report. Since view filters restrict the report results, view filters can have an effect on derived metrics. For information on view filter effects on derived metrics and

Copyright © 2024 All Rights Reserved

other reporting features, see *View filter effects on reporting features, page 239*.

Derived element effects on derived metrics

Most derived metrics are represented correctly by derived elements. However, note the following:

- In reports, a derived metric that uses division, such as `Profit/Revenue`, is initially calculated incorrectly when using derived elements. This is because, by default, derived metrics are evaluated before derived elements. You must change the evaluation order to evaluate derived elements before this type of derived metric. For information on derived element effects on derived metrics and how to evaluate them correctly, see *Derived element interactions with derived metrics, page 165*.
- In Report Services documents that use multiple datasets, if you create a derived metric that uses metrics from two or more datasets, the derived metric may display unexpected results for derived elements on Grid/Graphs. To ensure that such derived metrics are evaluated correctly, you must change the evaluation order to evaluate derived elements before this type of derived metric. For information see *Derived element interactions with derived metrics, page 165*.

Copyright © 2024 All Rights Reserved

VIEW FILTERS

Copyright © 2024 All Rights Reserved

A view filter restricts the amount of data displayed on the report, providing you with a different view of the data. This new view can provide a new business perspective for analysis, without having to re-execute the report's SQL against the data warehouse.

For example, you have a report with Region, Category, and Profit displayed, as shown in the report below.

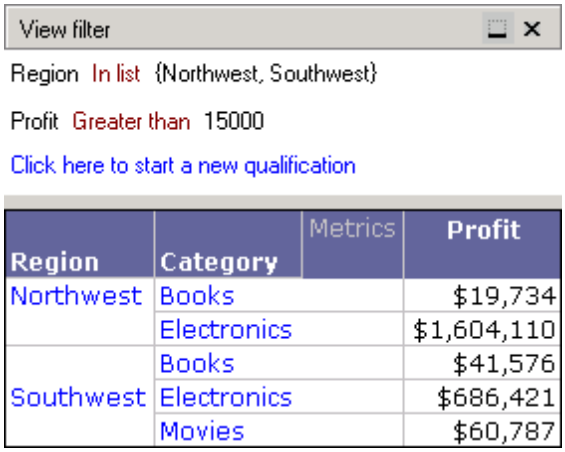
Region	Category	Metrics	Profit
Central	Books		$55,495
	Electronics		$935,123
	Movies		$16,347
	Music		$78,357
Mid-Atlantic	Books		$46,461
	Electronics		$3,933,032
	Movies		$14,351
	Music		$13,737
Northeast	Books		$455,964
	Electronics		$1,500,514
	Movies		$27,367
	Music		$25,895
Northwest	Books		$19,734
	Electronics		$1,604,110
	Movies		$5,639
	Music		$5,508
South	Books		$216,663
	Electronics		$739,064
	Movies		$12,905
	Music		$11,882
Southeast	Books		$25,609
	Electronics		$419,847
	Movies		$37,704
	Music		$6,965
Southwest	Books		$41,576
	Electronics		$686,421
	Movies		$60,787
	Music		$11,294
Web	Books		$29,195
	Electronics		$469,635
	Movies		$42,611
	Music		$7,952

After a view filter is applied, the resulting report below includes the following view filter qualifications:

Copyright © 2024 All Rights Reserved

- Region In list {Northwest, Southwest}: This qualification restricts the report results to display data only for the Northwest and Southwest regions.
- Profit Greater than 15000: This qualification restricts the report results to display data only for product categories in the Northwest or Southwest regions that had greater than $15,000 in profits.

The view filter's definition is displayed above the report, as shown below.



Region	Category	Metrics	Profit
Northwest	Books		$19,734
	Electronics		$1,604,110
Southwest	Books		$41,576
	Electronics		$686,421
	Movies		$60,787

The following table lists scenarios where you can use view filters to best support your business model and enhance the analysis of your reports.

Analysis Capability	Example
Modify the data displayed without re-executing SQL against the data warehouse.	Adding, deleting, or modifying view filters are all executed against a report in memory.
Allow multiple users to create separate views of data on a single report in memory.	Multiple users can define individual view filters to further restrict the data of a report connected to a shared Intelligent Cube.
Filter on attributes included in the report.	With the attribute Year on a report, you can use a view filter to determine which years of data to display on the report.
Perform attribute-to-attribute	With the attributes Customer City and Store City

Copyright © 2024 All Rights Reserved

Analysis Capability	Example
comparisons.	on a report, you can specify that Customer City be the same as the Store City. This can give a view of how a store is performing with local customers.
Filter on metrics included in the report. The output level for the filter can be applied at the report level or the level of the attributes displayed on the report.	With the metric Profit on a report, you can filter on Profit greater than or equal to $1,000,000.
Perform metric-to-metric comparisons.	With Revenue and Operating Cost metrics on a report, you can specify that Revenue be greater than or equal to Operating Cost.
Filter on attributes or metrics that are not displayed on the report.	You can drag-and-drop the Profit metric from the report grid to the Report Objects pane. This removes the Profit metric from the display, but any view filters based on that object are still calculated.

This section discusses the following topics related to view filters:

- *Comparing view filters to report filters and report limits, page 204*
- *Creating a view filter, page 211*
- *Deleting a view filter, page 238*
- *View filter effects on reporting features, page 239*

Comparing view filters to report filters and report limits

Filtering reports restricts the data displayed to highlight the data that is of greatest interest or is most applicable to the area of analysis. In addition to view filters, you can also use report filters and report limits to filter data on

Copyright © 2024 All Rights Reserved

reports. To decide when to use a view filter, report filter, or report limit, you must understand the advantages and implications of each feature. For a detailed comparison between view filters and other standard MicroStrategy features, see the following sections:

- *View filters versus report filters , page 205*
- *View filters versus report limits, page 209*

View filters versus report filters

While they both share some filtering capabilities, view filters and report filters also offer their own unique filtering features that fit different filtering requirements.

The main difference between report filters and view filters is how they are evaluated by the system.

Report filters are a standard MicroStrategy reporting feature that enable you to filter the data retrieved from the data source. Since report filters are evaluated by querying the data source, report filters can perform various types of advanced qualifications, use prompts in qualifications, filter on objects not included in the report, and so on. For more information on report filters in general, refer to the Basic Reporting Help and Advanced Reporting Help.

View filters are an OLAP Services feature that enable you to filter the data available on a report after its data has been retrieved from a data source. View filters are evaluated without having to query the data source. While this enables view filters to be evaluated without the overhead of querying the data source, it also means that view filters only have access to the data available on the report. Due to this limited access to data, view filters cannot perform all of the advanced qualifications possible with report filters.

The table below compares the available features and feature requirements of view filters and report filters:

Copyright © 2024 All Rights Reserved

Features and Feature Requirements	Available in View Filters	Available in Report Filters
Attribute element and attribute form qualifications	Yes	Yes
Simple metric qualifications	Yes	Yes
Relationship qualifications	No	Yes
Joint element list qualifications	No	Yes
Custom expression qualifications	No	Yes
Can define filtering at report run time by including a prompt in a qualification	No	Yes
Can filter on objects not included in the Report Objects pane (See below)	No	Yes
Can filter on objects included in the Report Objects pane, but not included in the report (See below)	Yes	Yes
Evaluated without re-executing SQL and querying the data source	Yes	No
Can be saved as a stand alone object and used in multiple reports	No	Yes
Can quickly switch the level at which the qualification is evaluated from report level to the level of attributes displayed on the report	Yes	No
Can be modified while viewing the report data	Yes	No

Design considerations

The decision to use a view filter or a report filter to answer your business questions relies on two key factors, functionality and system management.

Copyright © 2024 All Rights Reserved

View filters and report filters both provide a rich set of filtering functionalities, which can be used to answer your business questions. However, since report filters are executed against the data warehouse, more advanced filtering is supported. You may need to use a report filter to implement some of your more advanced business questions. For example, you can define a report filter at report run time by including a prompt in the filter definition.

From a system management perspective, report filters and view filters provide two alternatives that affect memory and data warehouse usage.

Report filters help to reduce the memory size of reports by returning less data from the data warehouse. These results can be stored in a cache that decreases the time it takes to access and run the report. The drawback to this approach is that any modifications to the report filter cause the system to access the data source again to create a new report definition, which must be stored in the cache in place of the old definition. The cache still provides quick access to the new report, but this process causes an extra load on the system.

View filters can help reduce memory used by reports, by utilizing Intelligent Cubes. Intelligent Cubes are sets of data that can be shared as a single in-memory copy, among many different reports created by multiple users. Filtering on reports connected to Intelligent Cubes is only achievable with view filters. View filters provide much of the same filtering functionality as report filters, while allowing multiple users to perform analysis on a single Intelligent Cube.

Example: Report filter on an attribute not in the report

For this example, run the **Employee Headcount by Region** report in the MicroStrategy Tutorial project; the report is shown below. This report does not have a report filter.

Copyright © 2024 All Rights Reserved

Region	Metrics	Employee Headcount
Central		4
England		2
France		1
Germany		2
Mid-Atlantic		8
Northeast		6
Northwest		3
South		3
Southeast		4
Southwest		5
Web		1

Add a report filter (Country = U.S.) that qualifies on the attribute Country, which is not part of the report's definition. The result is the report shown below, which contains data for the regions in the U.S. only, as defined in the report filter.

Region	Metrics	Employee Headcount
Central		4
Mid-Atlantic		8
Northeast		6
Northwest		3
South		3
Southeast		4
Southwest		5

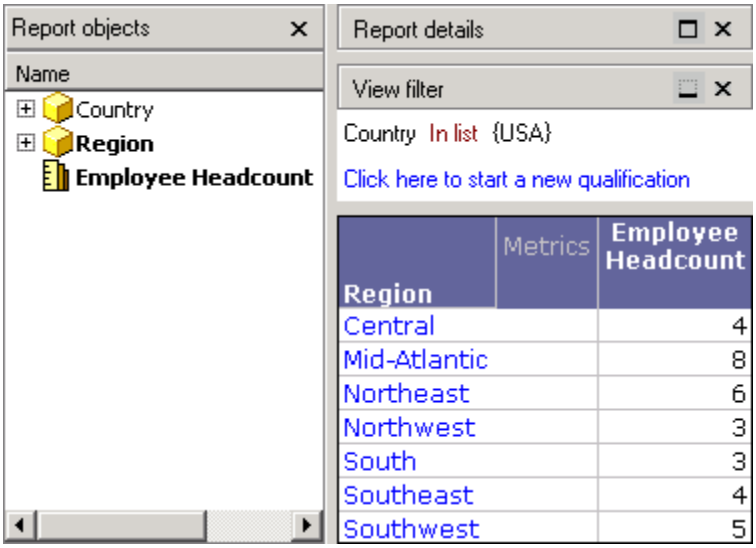
Although the report filter is based on an attribute not in the report itself, the report data is still affected because of the relationships among the objects in the report, which are all part of the same attribute hierarchy.

In contrast, a view filter can be created only on objects that are part of the report's definition. (The objects in a report's definition are displayed in the Report Objects pane.) This is because view filters only have access to data in the Report Objects pane of the report, rather than the entire data warehouse.

To provide the same type of analysis used in the first report that uses a report filter, include Country in the Report Objects pane of the report, but

Copyright © 2024 All Rights Reserved

remove it from the grid. With Country in the Report Objects pane, you can create a view filter to restrict data to Regions in the US, as shown below.



View filters versus report limits

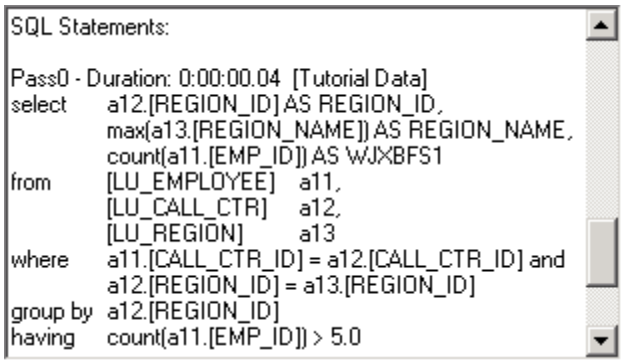
View filters and report limits are similar filtering features, as they can both be used to restrict the final result set by setting criteria on the report metrics.

You can define report limits on the value, rank, or percent of the metric value returned. For example, if you set a report limit of `Employee Headcount greater than (>) 5` in the **Employee Headcount by Region** report in the MicroStrategy Tutorial project, the report returns the following results.

Region	Metrics	Employee Headcount
Mid-Atlantic		8
Northeast		6

An important aspect of report limits is that they are processed by the SQL Engine after metrics are aggregated. In the report SQL, the report limit definition is included in the `Having` clause, instead of the `Where` clause as for the report filter. You can observe this in the report's SQL statement, shown in the image below.

Copyright © 2024 All Rights Reserved



In addition to this functionality, the following features supported for report limits are not supported for view filters:

- Prompts within report limit
- Break By

The table below compares the available features and feature requirements of view filters and report limits:

Features and Feature Requirements	Available in View Filters	Available in Report Filters
Attribute qualifications	Yes	No
Simple metric qualifications	Yes	Yes
Can define filtering at report run time by including a prompt in a qualification	No	Yes
Can define Break By filtering parameters	No	Yes
Evaluated without re-executing SQL and querying the data source	Yes	No
Can quickly switch the level at which the qualification is evaluated from report level to the level of attributes displayed on the report	Yes	No

For general information on report limits, see the Basic Reporting Help.

Copyright © 2024 All Rights Reserved

Design considerations

A view filter is similar to a report limit in that it can also be applied at the report level. However, the report limit and the view filter are not interchangeable. A report limit restricts the size of the report data set that is returned from the data warehouse. In contrast, the view filter is applied to the report dataset without altering its size, allowing you to view a subset of that information. A view filter retrieves information quickly because Intelligence Server dynamically accesses the data already in the report results.

Report designers must consider how to balance the memory usage and the processing power between the data warehouse and Intelligence Server. A report limit is more efficient in terms of report data size because it does not return unnecessary information from the data warehouse. Therefore, the report limit can be used to save space on the Intelligence Server memory. However, if a report limit is too restrictive, you may need to frequently redefine the data definition to yield the information users want to see.

On the other hand, a view filter is more flexible, allowing you to refine your analysis after the report is executed. A view filter gives you more control over the subset of data retrieved from the database you want to see. The view filter may be more useful for analysts because it allows analysts to conduct further investigation and refinement of the report results after the report is executed against the data warehouse.

Creating a view filter

You can create a view filter by defining a condition that is based on any attribute or metric on your report. While attribute and metric qualifications both restrict the data displayed on the report, they are created and defined in slightly different ways.

You can create a view filter in any report. This includes reports connected to an Intelligent Cube, reports returning information directly from a data

Copyright © 2024 All Rights Reserved

warehouse, Freeform SQL reports returning information from an Excel spreadsheet, and so on.

The following sections provide examples of using view filters on reports returning information directly from a data warehouse.

Refer to the following sections to create a view filter that contains attribute and/or metric qualifications:

- *Filtering data based on business attributes, page 213*: Using view filters based on business attributes, you can view a subset of report data that focuses on the business data you are interested in.
- When creating an attribute qualification for a view filter, you can either qualify on a list of attribute elements, or you can qualify on attribute forms:
  - Qualifying on a list of attribute elements is achievable by using the In list or Not in list operators and selecting from a list of attribute elements. For basic steps to filter on a list of attribute elements, see the procedure *To create a view filter with an attribute qualification, page 214*.
  - Qualifying on attribute forms is achievable by using the Where operator and selecting from available attribute forms. For information on qualifying on attribute forms, see *Filtering based on attribute form qualifications, page 216*.
- *Filtering data based on metrics, page 218*: Using view filters based on metrics, you can view a subset of report data that focuses on the data values and ranges you are interested in.
- *Combining view filter qualifications with operators, page 237*: When a view filter has multiple qualifications at the same output level, they are always joined by operators. When qualifications are joined, operators govern the interaction between the different filtering conditions.

Copyright © 2024 All Rights Reserved

Filtering data based on business attributes

Using a view filter based on business attributes, you can view a subset of report data that focuses on the business data you are interested in. Narrowing the focus of a report to the business data that is of interest of you enables another level of report analysis that can highlight business trends and figures.

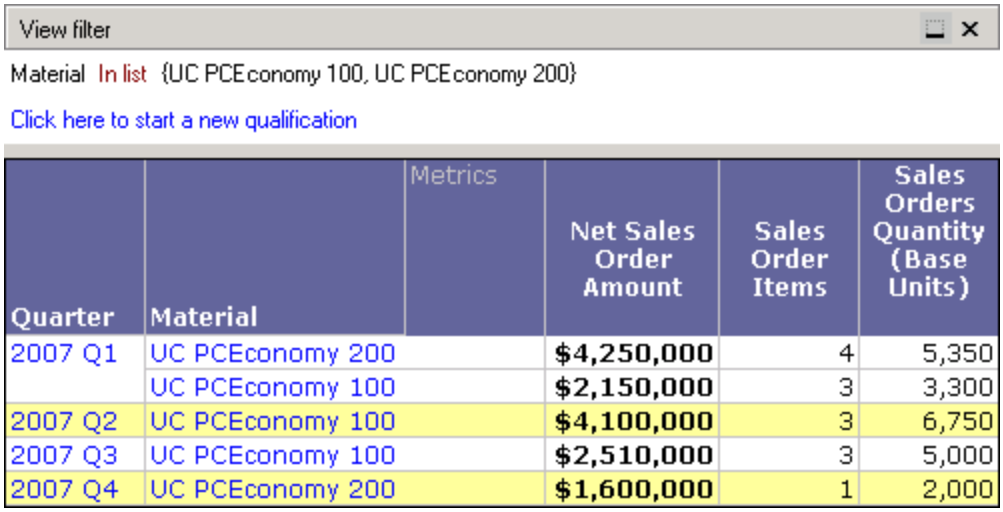
For example, the **TOP 5 Materials by Net Sales Amount and Quarter** report from the Sales and Distribution Analysis Module is used for this example. The year 2007 has been selected to answer the prompt on year. The resulting report is shown below.

Quarter	Material	Metrics	Net Sales Order Amount	Sales Order Items	Sales Orders Quantity (Base Units)
2007 Q1	UC PCEconomy 200		$4,250,000	4	5,350
	UC PCValue 200		$4,150,000	5	3,350
	UC PCEconomy 100		$2,150,000	3	3,300
	UC PCTop 200		$1,870,000	4	900
	UC PCValue 100		$1,725,000	4	1,750
2007 Q2	UC PCEconomy 100		$4,100,000	3	6,750
	UC PCValue 200		$2,740,000	7	2,020
	UC PCValue 100		$2,413,500	6	2,305
	UC PCTop 200		$2,200,000	3	1,000
	UC NBProfessional 100		$1,703,000	4	1,310
2007 Q3	UC PCValue 100		$5,697,500	6	5,575
	UC PCValue 200		$4,165,000	7	3,275
	UC LaserPrinter B/W - E10		$2,743,500	9	9,435
	UC PCEconomy 100		$2,510,000	3	5,000
	UC PCTop 200		$1,570,000	6	735
2007 Q4	UC PCValue 100		$2,175,000	3	2,250
	UC PCTop 200		$1,632,000	5	760
	UC PCEconomy 200		$1,600,000	1	2,000
	UC NBProfessional 100		$1,393,000	3	1,110
	UC PCTop 100		$1,216,000	6	720

The report identifies, for each quarter, the top five materials by net sales order amount.

Copyright © 2024 All Rights Reserved

Once you review the top 5 materials for each quarter in 2007, you decide to focus on the UC PCEconomy 100 and UC PCEconomy 200 by creating a view filter that includes these two materials. The view filter along with the resulting report is shown below.


View filter

Material In list {UC PCEconomy 100, UC PCEconomy 200}

Click here to start a new qualification

Quarter	Material	Metrics	Net Sales Order Amount	Sales Order Items	Sales Orders Quantity (Base Units)
2007 Q1	UC PCEconomy 200		$4,250,000	4	5,350
	UC PCEconomy 100		$2,150,000	3	3,300
2007 Q2	UC PCEconomy 100		$4,100,000	3	6,750
2007 Q3	UC PCEconomy 100		$2,510,000	3	5,000
2007 Q4	UC PCEconomy 200		$1,600,000	1	2,000

The report shown above focuses on the UC PCEconomy 100 and UC PCEconomy 200 materials to show when these two materials had net sales order amounts in the top 5 for a given quarter.

The steps below show you how to create a view filter with an attribute qualification, as well as how to create the example scenario above.

Prerequisites

- You need the Use View Filter Editor (Developer) and/or the Web Use View Filter Editor (Web) privileges. These privileges are part of OLAP Services.

To create a view filter with an attribute qualification

1. Log in to a project in MicroStrategy Developer. For steps to create a view filter in MicroStrategy Web, see the MicroStrategy Web Help.

   For the example scenario, log in to the Sales and Distribution Analysis Module project.

Copyright © 2024 All Rights Reserved

2. Run the report in Grid View.

   For the example scenario, browse to and run the **TOP 5 Materials by Net Sales Amount and Quarter** report. Select the **2007** attribute element to answer the prompt.

3. If the View Filter area is not displayed, from the **View** menu, select **View Filter**.

4. In the **View Filter** area, click **Click here to start a new qualification**.

5. Click **Field**, and then select an attribute.

   For the example scenario, select **Material**.

6. Click **Operator**, and then select one of the following operators to create a filter qualification (for the example scenario, select **In list**):

   - **In list**: Restricts attribute data to the list of attribute elements you select. Click **Value**, and then select the attribute elements to restrict attribute data to.

     For the example scenario, select **UC PCEconomy 100** and **UC PCEconomy 200**.

   - **Not in List**: Restricts attribute data to the list of attribute elements that are not in the list of attribute elements you select. Click **Value**, and select the attribute elements to exclude data for.

   - **Where**: Restricts attribute data based on a filter qualification of an attribute form.

7. Click **OK**.

8. If the Auto-Apply Changes check box is cleared, click **Apply** to apply the view filter to the report.

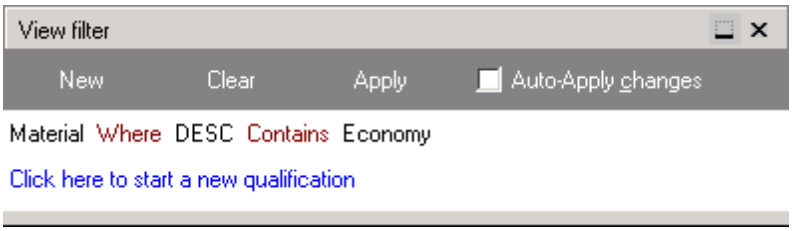
The report is updated and the report data is restricted as defined by the view filter. If you define multiple view filter qualifications at the same output level, you can modify the logical operator used to join the qualifications, as described in *Combining view filter qualifications with operators, page 237*.

Copyright © 2024 All Rights Reserved

For information on the output level of view filter qualifications, see *Evaluating qualifications at the report or grid level, page 232*.

Filtering based on attribute form qualifications

You can qualify on attribute forms using the Where operator. Once you select an attribute form, you can use various logical and mathematical operators to create a view filter qualification based on an attribute form, to restrict data. An attribute form qualification using a Contains operator that restricts data to materials that contain Economy in their DESC form is shown below.



If you use this view filter with the TOP 5 Materials by Net Sales Amount and Quarter report, the resulting report is the same as the example scenario in *Filtering data based on business attributes, page 213*, which uses the attribute element qualification `Material In list {UC PCEconomy 100, UC PCEconomy 200}`.

You can use any of the following operators in attribute form qualifications. These operators are described in detail in *Appendix A, Logical and Mathematical Operators for Filtering* in the Advanced Reporting Help:

Copyright © 2024 All Rights Reserved

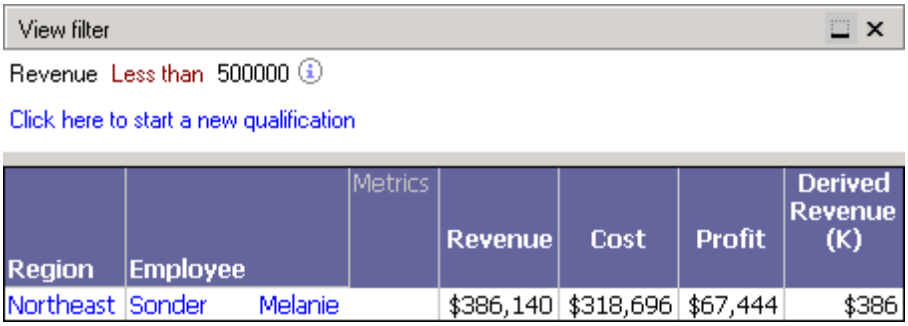
Operator Type	Operator
Comparison Operators Comparison operators compare values. The values can be numbers, text strings, or expressions.	Exactly
	Different from
	Greater than
	Less than
	Greater than or equal to
	Less than or equal to
	Between
	Is Null
	Is Not Null
Pattern Operators Pattern operators allow text strings to be compared. Pattern operators are case-sensitive.	Like
	Not Like
	Contains
	Does not contain
	Begins with
	Does not begin with
	Ends with
	Does not end with

Copyright © 2024 All Rights Reserved

Filtering data based on metrics

Using view filters based on metrics, you can view a subset of report data that focuses on the data values and ranges you are interested in. Narrowing the focus of a report to the data that is of interest of you enables another level of report analysis that can highlight business trends and figures.

For example, review the **16 View Filter - Metric Qualification** report from the Tutorial project, shown below.



Region	Employee	Metrics	Revenue	Cost	Profit	Derived Revenue (K)
Northeast	Sonder Melanie		$386,140	$318,696	$67,444	$386

This report returns revenue, cost, and profit data for employees, while also displaying the region the employee is in. The view filter restricts the report results to only return data for those employees who generated less than $500,000 in revenue. This reduces the large number of employee results to a smaller set of employees that are generating a relatively low amount of revenue.

When creating metric qualifications in a view filter, you can use various logical and mathematical operators. You can use any of the following operators in metric qualifications. These are described in detail in *Appendix A, Logical and Mathematical Operators for Filtering* in the Advanced Reporting Help:

- Exactly
- Different from
- Greater than
- Less than

Copyright © 2024 All Rights Reserved

- Greater than or equal to
- Less than or equal to
- Between
- Not Between
- Is Null
- Is Not Null

Once you select an operator, you can either type in a value or select a metric to return the value to qualify on.

You can also qualify on the rank or percentage of a metric value for a given report. For example, you can restrict the report shown above to display all data for employees in the bottom 20% of revenue. For information on using view filters to restrict report results based on ranks or percentages of metric data, see *Filtering metrics on rank and percentage ranges, page 223*.

The steps below show you how to create a view filter with a metric qualification, as well as how the example scenario above was created.

Prerequisites

- You need the Use View Filter Editor (Developer) and/or the Web Use View Filter Editor (Web) privileges. These privileges are part of OLAP Services.

To create a view filter with a metric qualification

1. Log in to a project in MicroStrategy Developer. For steps to create a view filter in MicroStrategy Web, see the MicroStrategy Web Help.

   For the example scenario, log in to the MicroStrategy Tutorial project.

2. Run the report in Grid View.

   For the example scenario, browse to and run the **16 View Filter - Metric Qualification** report.

Copyright © 2024 All Rights Reserved

3. If the View Filter area is not displayed, from the **View** menu, select **View Filter**.
4. In the **View Filter** area, click **Click here to start a new qualification**.

   You can also create a new view filter qualification on a metric by right-clicking a metric, pointing to **Filter On**, and selecting **Add Condition**.
5. Click **Field**, and then select a metric.

   For the example scenario, select **Revenue**.
6. Click **Operator**, and then select an operator.

   For the example scenario, select **Less than**.
7. Click **Value**, and then select **Type a value**, or select a metric to return the value to qualify on.

   For the example scenario, select **Type a value**, and then type **500000**.
8. If the Auto-Apply Changes check box is cleared, click **Apply** to apply the view filter to the report.

The report is updated. The report data is restricted as defined by the view filter. If you define multiple view filter qualifications at the same output level, you can modify the logical operator used to join the qualifications, as described in *Combining view filter qualifications with operators, page 237*.

Filtering based on metric-to-metric comparisons

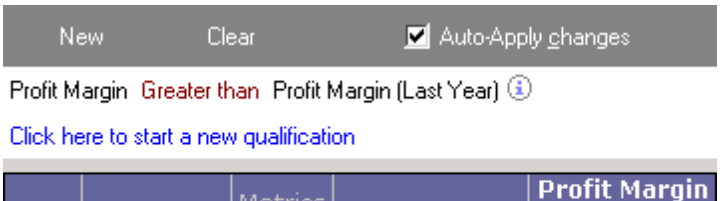
You can compare values of two metrics to filter the results of a report by creating metric-to-metric qualifications. This type of metric qualification can provide analysis such as comparing metric values over time.

For example, you can create a report that restricts the data to profit margins greater than last year's profit margins. The report shown below includes Year, Region, Profit Margin, and Profit Margin (Last Year), and was created in the MicroStrategy Tutorial project.

Copyright © 2024 All Rights Reserved

Year	Region (Metrics)	Profit Margin	Profit Margin (Last Year)
2006	Central	15.81%	15.63%
	Mid-Atlantic	17.29%	17.32%
	Northeast	17.85%	17.40%
	Northwest	17.35%	17.31%
	South	18.07%	17.62%
	Southeast	15.65%	15.44%
	Southwest	15.36%	15.45%
	Web	15.41%	15.08%
2007	Central	15.66%	15.81%
	Mid-Atlantic	17.27%	17.29%
	Northeast	17.69%	17.85%
	Northwest	17.42%	17.35%
	South	16.91%	18.07%
	Southeast	15.68%	15.65%
	Southwest	15.51%	15.36%
	Web	15.40%	15.41%

You can create a view filter to then restrict the data on the report to profit margins greater than last year's profit margins. The metric-to-metric qualification and resulting report are shown below.

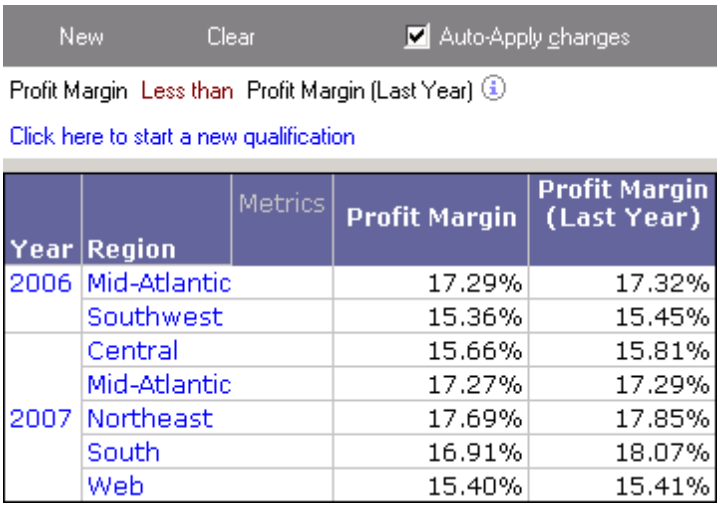


Year	Region (Metrics)	Profit Margin	Profit Margin (Last Year)
2006	Central	15.81%	15.63%
	Northeast	17.85%	17.40%
	Northwest	17.35%	17.31%
	South	18.07%	17.62%
	Southeast	15.65%	15.44%
	Web	15.41%	15.08%
2007	Northwest	17.42%	17.35%
	Southeast	15.68%	15.65%
	Southwest	15.51%	15.36%

You can now review when regions had increases in profit margins. For example, from the report above you can determine that the Northwest and Southeast regions have shown increases in profit margins from 2006 to 2007.

Copyright © 2024 All Rights Reserved

You can take advantage of view filters' ability to update the report results without having to re-execute SQL against the data warehouse to perform further quick analysis. For example, you can switch the operator from Greater Than to Less Than to quickly switch to a view of data for profit margins that are less than the previous year's profit margins. The metric-to-metric qualification in the view filter and resulting report are shown below.



Year	Region	Metrics	Profit Margin	Profit Margin (Last Year)
2006	Mid-Atlantic		17.29%	17.32%
	Southwest		15.36%	15.45%
2007	Central		15.66%	15.81%
	Mid-Atlantic		17.27%	17.29%
	Northeast		17.69%	17.85%
	South		16.91%	18.07%
	Web		15.40%	15.41%

The steps below show you how to create a view filter with a metric-to-metric qualification, as well as how the example scenario above was created.

To create a view filter with a metric-to-metric qualification

1. Log in to a project in MicroStrategy Developer. For steps to create a view filter in MicroStrategy Web, see the MicroStrategy Web Help.

   For the example scenario, log in to the MicroStrategy Tutorial project.

2. Run the report in Grid View.

   For the example scenario, create a report with Year, Region, and Profit Margin on the report. Create a metric that returns a transformation of last year's profit margin values, and name the metric Profit Margin (Last Year). For details to create a transformation metric, see the Advanced Reporting Help.

Copyright © 2024 All Rights Reserved

3. Choose **View** > **View Filter** > **Click here to start a new qualification**.

   You can also create a new view filter qualification on a metric by right-clicking a metric, pointing to **Filter On**, and selecting **Add Condition**.

4. Click **Field**, and then select a metric.

   For the example scenario, select **Profit Margin**.

5. Click **Operator**, and then select an operator.

   For the example scenario, select **Greater than**.

6. Click **Value**, and then select a metric to return the value to qualify on.

   For the example scenario, select **Profit Margin (Last Year)**.

7. Click **Apply** to apply the view filter to the report.

The report is updated. The report data is restricted as defined by the view filter. If you define multiple view filter qualifications at the same output level, you can modify the logical operator used to join the qualifications, as described in *Combining view filter qualifications with operators, page 237*.

Filtering metrics on rank and percentage ranges

In addition to restricting report results based on metric qualifications, you can also restrict report results based on the rank or percentage range of a metric value for a given report. This enables you to display reports with data such as products in the top 40% of profit. You could also create a report to return the top 10 employees in terms of average performance score.

It is important to understand the operators that are available for these types of view filters. When creating view filter qualifications on the rank or percentage ranges of metric values, you can use the following mathematical and logical operators, which are described as they relate to rank and percent metric qualifications in the table below:

Copyright © 2024 All Rights Reserved

In Developer, all of the operators listed below can be used to create rank and percent metric qualifications as part of a view filter. Rank and percent metric qualifications using any of these operators can be viewed in MicroStrategy Web. However, only the Is Highest (referred to as Highest or Highest% in Web) and Is Lowest (referred to as Lowest or Lowest% in Web) can be used to create or modify rank and percent metric qualifications in Web.

Operator	Functionality and considerations
Exactly	Identifies a specific value, as described below: For rank ranges, this restricts a report to data for only one specific rank. For example, a rank range defined as Exactly 1 returns data for only the highest value of a given metric. This scenario produces the same results as using the Is Highest operator described below in this table. For percent ranges, this operator has limited value. This is because it restricts a report to data that is exactly a specific percentage. Percentages are not always integer values and therefore it is difficult to predict what percentage would return meaningful results.
Different from	Identifies values that are other than the specific value indicated: For rank ranges, this restricts a report to data for only ranks that are different from the given value. For example, a rank range defined as different from 1 returns data for all ranks except the highest value of a given metric. For percent ranges, this operator has limited value. This is because it restricts a report to data that is different from a specific percentage. Percentages are not always integer values and therefore it is difficult to predict what percentage would return meaningful results. In most cases, using this operator with percent metric qualifications does not restrict any data.
Between	Identifies values in a range that has both a lower and an upper limit: For rank ranges, you can provide a lower and upper rank limit. For example, you can display only ranks 20 through 40. For percent ranges, you can provide a lower and upper percentage limit. For

Copyright © 2024 All Rights Reserved

Operator	Functionality and considerations
	example, you can display only data with metric values between 20% and 40% of all values.
Not Between	Identifies values in a range that has both a lower and an upper limit. Data is restricted for any values outside of this range: For rank ranges, you can provide a lower and upper rank limit. For example, you can display only ranks not between 20 through 40. This displays ranks 1 through 19 and 41 and down. For percent ranges, you can provide a lower and upper percentage limit. For example, you can display only data with metric values not between 20% and 40% of all values. This displays data with metric values between 1% and 19% and between 41% and 100%.
Is Null	Identifies values that are null. Using the rank or percent metric qualifications is not necessary with this operator. To return all data where metric values are Null, you can simply create a view filter metric qualification on the metric that uses the Is Null function.
Is Not Null	Identifies values that are not null. Using the rank or percent metric qualifications is not necessary with this operator. To return all data where metric values are not null, you can simply create a view filter metric qualification on the metric that uses the Is Not Null function.
Top	Identifies the topmost value range in a given set: For rank ranges, you can provide a topmost rank range. For example, you can display only data within the top 20 rank range. For percentage ranges, you can provide a topmost percentage range. For example, you can display only data within the top 20% range.
Bottom	Identifies the lowest set of values in a given set: For rank ranges, you can provide a lowest rank range. For example, you can display only data within the bottom 20 rank range. For percentage ranges, you can provide a lowest percentage range. For example, you can display only data within the bottom 20% range.
Exclude top	Identifies a value range that is not in the topmost value range in a given set:

Copyright © 2024 All Rights Reserved

Operator	Functionality and considerations
	For rank ranges, you can provide a topmost rank range to exclude from the report results. For example, you can display only data that excludes the top 20 rank range. For percentage ranges, you can provide a topmost percentage range to exclude from the report results. For example, you can display only data that excludes the top 20% range.
Exclude bottom	Identifies a value range that is not in the lowest set of values in a given set: For rank ranges, you can provide a lowest rank range to exclude from the report results. For example, you can display only data that excludes the bottom 20 rank range. For percentage ranges, you can provide a lowest percentage range to exclude from the report results. For example, you can display only data that excludes the bottom 20% range.
Is Highest	Identifies the highest value. This operator should only be used with rank ranges. For rank ranges, this restricts report results to display data for only the highest value of a given metric.
Is Lowest	Identifies the lowest value. This operator should only be used with rank ranges. For rank ranges, this restricts report results to display data for only the lowest value of a given metric.

Creating a view filter on a rank range of metric values

You can create a view filter that restricts report results based on a rank range of metric values for a given report. This can allow you to view analysis such as the bottom 20 products in terms of profit margin. Further analysis can be explored on this rank range of products to determine how to increase profit margins for these products, or decide which products should be discontinued.

This type of report and analysis can be created in the MicroStrategy Tutorial project. You first create a report with Item, Revenue, Profit, and Profit Margin on the report, as shown below.

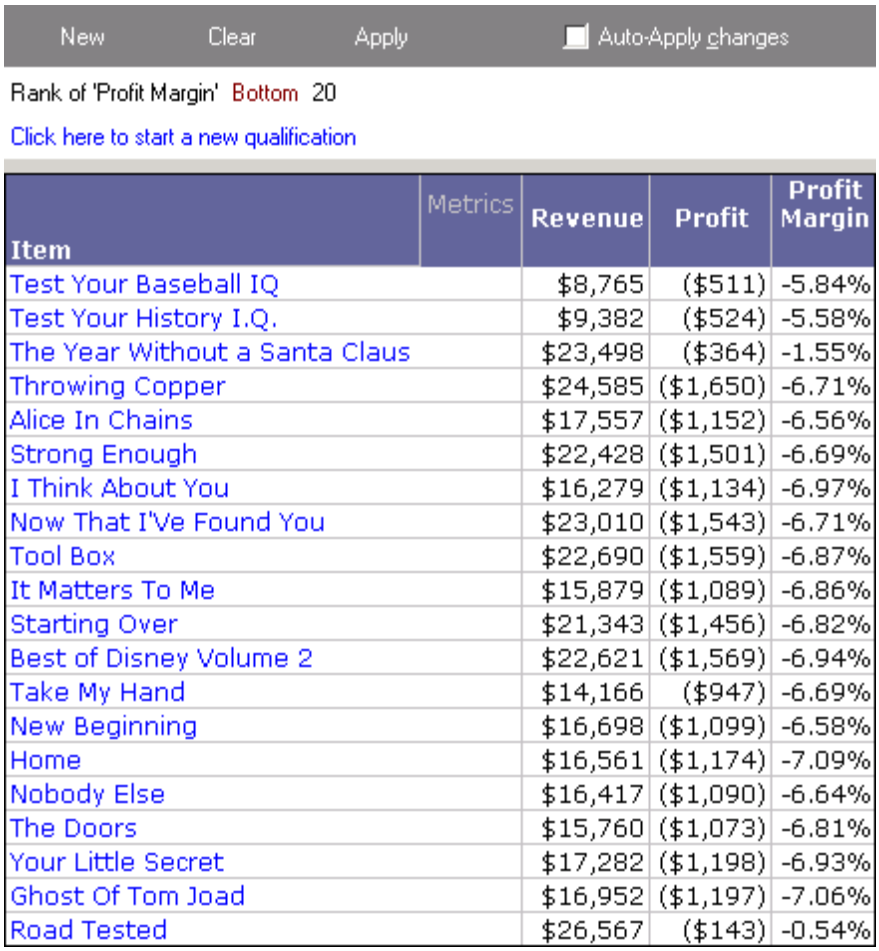
Copyright © 2024 All Rights Reserved

Item	Metrics	Revenue	Profit	Profit Margin
100 Places to Go While Still Young at Heart		$135,930	$35,859	26.38%
Art As Experience		$31,738	$7,312	23.04%
The Painted Word		$17,495	$2,080	11.89%
Hirschfeld on Line		$103,318	$25,087	24.28%
Adirondack Style		$77,428	$19,826	25.61%
Architecture : Form, Space, & Order		$82,099	$20,027	24.39%
50 Favorite Rooms		$51,105	$12,026	23.53%
500 Best Vacation Home Plans		$23,239	$4,771	20.53%
Blue & White Living		$32,014	$7,424	23.19%
Ways of Seeing		$30,961	$7,134	23.04%
Gonzo, the Art		$81,743	$16,697	20.43%
Cabin Fever : Rustic Style Comes Home		$27,230	$5,633	20.68%
American Bungalow Style		$95,617	$23,296	24.36%

Execution complete | Execution Time: 00:00:01 | Rows: 360 | Columns: 3 | Local Template | Standard

Only a subset of the report results are shown above, but notice that data for 360 items have been returned. To narrow the analysis of the report, you create a view filter to restrict the report results to the bottom 20 products in terms of profit margin. The view filter and resulting report are shown below.

Copyright © 2024 All Rights Reserved



Item	Metrics	Revenue	Profit	Profit Margin
Test Your Baseball IQ		$8,765	($511)	-5.84%
Test Your History I.Q.		$9,382	($524)	-5.58%
The Year Without a Santa Claus		$23,498	($364)	-1.55%
Throwing Copper		$24,585	($1,650)	-6.71%
Alice In Chains		$17,557	($1,152)	-6.56%
Strong Enough		$22,428	($1,501)	-6.69%
I Think About You		$16,279	($1,134)	-6.97%
Now That I'Ve Found You		$23,010	($1,543)	-6.71%
Tool Box		$22,690	($1,559)	-6.87%
It Matters To Me		$15,879	($1,089)	-6.86%
Starting Over		$21,343	($1,456)	-6.82%
Best of Disney Volume 2		$22,621	($1,569)	-6.94%
Take My Hand		$14,166	($947)	-6.69%
New Beginning		$16,698	($1,099)	-6.58%
Home		$16,561	($1,174)	-7.09%
Nobody Else		$16,417	($1,090)	-6.64%
The Doors		$15,760	($1,073)	-6.81%
Your Little Secret		$17,282	($1,198)	-6.93%
Ghost Of Tom Joad		$16,952	($1,197)	-7.06%
Road Tested		$26,567	($143)	-0.54%

With this updated report, you can now perform further analysis on each item to determine a strategy to improve your profit margins.

Notice that the view filter above uses the Bottom operator.

The steps below show you how to create a view filter with a rank metric qualification, as well as how the example scenario above was created.

Prerequisites

- You need the Use View Filter Editor (Developer) and/or the Web Use View Filter Editor (Web) privileges. These privileges are part of OLAP Services.

Copyright © 2024 All Rights Reserved

To create a view filter with a rank metric qualification

1. Log in to a project in MicroStrategy Developer. For steps to create a view filter in MicroStrategy Web, see the MicroStrategy Web Help.

   For the example scenario, log in to the MicroStrategy Tutorial project.

2. Run a report in Grid View.

   For the example scenario, create a report with Item, Revenue, Profit, and Profit Margin on the report, as shown in *Creating a view filter on a rank range of metric values, page 226*.

3. If the View Filter area is not displayed, from the **View** menu, select **View Filter**.

4. In the **View Filter** area, click **Click here to start a new qualification**.

5. To create a rank metric qualification, click **Field**, point to **Rank**, and then select a metric.

   For the example scenario, select **Profit Margin**.

6. Click **Operator**, and then select an operator.

   For the example scenario, select **Bottom**.

7. Click **Value**, and then select **Type a value**. Type the value for the rank number you want to restrict data to.

   For the example scenario, type **20**.

8. If the Auto-Apply Changes check box is cleared, click **Apply** to apply the view filter to the report.

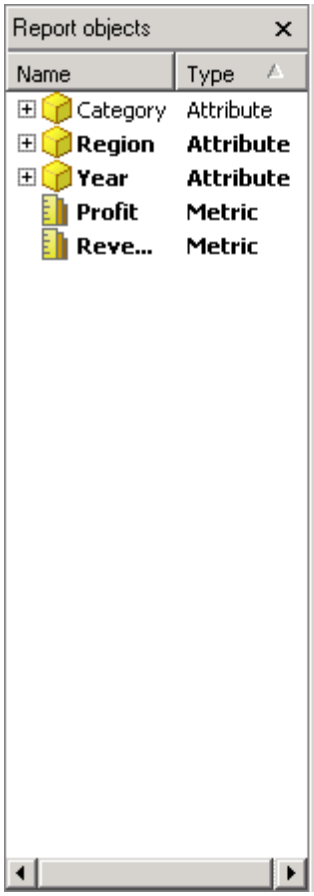
The report is updated. The report data is restricted as defined by the view filter. If you define multiple view filter qualifications at the same output level, you can modify the logical operator used to join the qualifications, as described in *Combining view filter qualifications with operators, page 237*.

Copyright © 2024 All Rights Reserved

Creating a view filter on a percentage range of metric values

You can create view filters that restrict report results based on a percent range of metric values for a given report. This can allow you to view analysis such as employees between 30% and 60% of tenure length with the company. You can then explore performance metrics within this percentage range of employees and determine where promotions and other actions are necessary.

For example, you create a report with Year, Region, Category, Revenue, and Profit on the report, with Category not displayed on the report grid, as shown below.



Year	Region	Metrics	Revenue	Profit
2005	Central		$1,823,715	$285,087
	Mid-Atlantic		$6,131,826	$1,062,064
	Northeast		$2,958,077	$514,633
	Northwest		$2,573,339	$445,363
	South		$2,011,972	$354,550
	Southeast		$836,399	$129,154
	Southwest		$1,418,189	$219,109
	Web		$720,560	$108,652
2006	Central		$2,182,632	$345,159
	Mid-Atlantic		$7,662,786	$1,324,696
	Northeast		$3,668,765	$655,013
	Northwest		$3,189,483	$553,323
	South		$1,901,549	$343,679
	Southeast		$1,036,970	$162,298
	Southwest		$1,625,727	$249,691
	Web		$1,133,901	$174,770
2007	Central		$2,906,587	$455,074
	Mid-Atlantic		$9,384,211	$1,620,821
	Northeast		$4,748,571	$840,094
	Northwest		$3,653,039	$636,303
	South		$1,669,019	$282,283
	Southeast		$1,266,947	$198,673
	Southwest		$2,135,390	$331,277
	Web		$1,726,816	$265,971

You decide to analyze this report to show only the data that is within the top 10% of profit. The view filter and resulting report are shown below.

Copyright © 2024 All Rights Reserved

Year	Region	Metrics	Revenue	Profit
2006	Mid-Atlantic		$7,662,786	$1,324,696
2007	Mid-Atlantic		$9,384,211	$1,620,821

With this analysis, you can now perform further analysis to determine why profit was at its highest during these years and within these regions.

Notice that the view filter above uses the Top operator.

The steps below show you how to create a view filter with a percent metric qualification, as well as how the example scenario above was created.

To create a view filter with a percent metric qualification

1. Log in to a project in MicroStrategy Developer. For steps to create a view filter in MicroStrategy Web, see the *MicroStrategy Web Help*.

   For the example scenario, log in to the MicroStrategy Tutorial project.

2. Run the report in Grid View.

   For the example scenario, create a report with Year, Region, Category, Revenue, and Profit on the report with Category not displayed on the report grid.

3. Choose **View** > **View Filter** > **Click here to start a new qualification**.

4. To create a percent metric qualification, click **Field**, point to **Percent**, and then select a metric.

   For the example scenario, select **Profit**.

5. Click **Operator**, and then select an operator.

   For the example scenario, select **Top**.

Copyright © 2024 All Rights Reserved

6. Click **Value**, and then select **Type a value**.Type the value for the rank number you want to restrict data to.

   For the example scenario, type **20**.

   Valid numbers for percent metric qualifications are 1 to 100.

7. If the Auto-Apply Changes check box is cleared, click **Apply** to apply the view filter to the report.

The report is updated. The report data is restricted as defined by the view filter. Further analysis on this report is performed to demonstrate how you can change the level of evaluation for view filter metric qualifications.

If you define multiple view filter qualifications at the same output level, you can modify the logical operator used to join the qualifications, as described in *Combining view filter qualifications with operators, page 237*.

Evaluating qualifications at the report or grid level

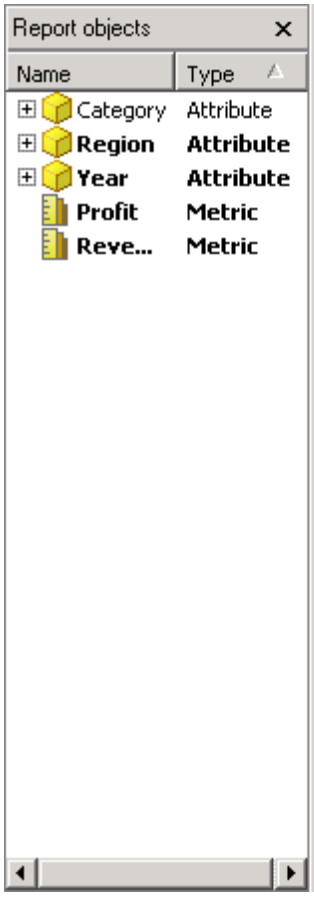
When a metric qualification in a view filter is evaluated by the system, the evaluation can be performed for all data that is returned for the report, or only the view of data that is currently available on the report grid. These two options can produce different report results when using the OLAP Services feature called dynamic aggregation.

Dynamic aggregation enables you to remove attributes from the report grid, but keep them as part of the report definition. The action of moving attributes on or off of the report grid aggregates the metric values at the new level of the report. For information on dynamic aggregation, see *Chapter 9, Dynamic Aggregation*.

By default, metric qualifications in a view filter are evaluated at the level of data that is available on the report grid. This means that any attributes that are included in the Report Objects pane but not on the report grid are not used to determine the level of the metric qualification.

Copyright © 2024 All Rights Reserved

For example, you create a report with Year, Region, Category, Revenue, and Profit on the report, with Category not displayed on the report grid, as shown below.



Year	Region	Metrics	Revenue	Profit
2005	Central		$1,823,715	$285,087
	Mid-Atlantic		$6,131,826	$1,062,064
	Northeast		$2,958,077	$514,633
	Northwest		$2,573,339	$445,363
	South		$2,011,972	$354,550
	Southeast		$836,399	$129,154
	Southwest		$1,418,189	$219,109
	Web		$720,560	$108,652
2006	Central		$2,182,632	$345,159
	Mid-Atlantic		$7,662,786	$1,324,696
	Northeast		$3,668,765	$655,013
	Northwest		$3,189,483	$553,323
	South		$1,901,549	$343,679
	Southeast		$1,036,970	$162,298
	Southwest		$1,625,727	$249,691
	Web		$1,133,901	$174,770
2007	Central		$2,906,587	$455,074
	Mid-Atlantic		$9,384,211	$1,620,821
	Northeast		$4,748,571	$840,094
	Northwest		$3,653,039	$636,303
	South		$1,669,019	$282,283
	Southeast		$1,266,947	$198,673
	Southwest		$2,135,390	$331,277
	Web		$1,726,816	$265,971

This report can be created in the MicroStrategy Tutorial project.

You can use dynamic aggregation to drag and drop the Category attribute from the report grid to the Report Objects pane. This allows Category to affect the report level without being displayed on the grid.

You decide to analyze this report to show only the data that is within the top 10% of profit. The view filter and resulting report are shown below.

Copyright © 2024 All Rights Reserved

Year	Region	Metrics	Revenue	Profit
2006	Mid-Atlantic		$7,662,786	$1,324,696
2007	Mid-Atlantic		$9,384,211	$1,620,821

Notice that only two rows of data are returned. The metric qualification has been evaluated at the level of the report grid, which is Year and Region. This gives you a view of data within the top 10% of profit for the data displayed on the report grid.

However, this report also includes the Category attribute in the Report Objects pane. Since this attribute is available on the report, you can also view data within the top 10% of profit at the Category, Region, and Year level. Evaluating the metric qualification at this level returns the report results shown below.

Year	Region	Metrics	Revenue	Profit
2005	Mid-Atlantic		$5,971,153	$1,042,487
	Northwest		$2,502,737	$436,648
2006	Mid-Atlantic		$7,460,047	$1,299,891
	Northeast		$2,755,465	$487,923
	Northwest		$3,106,552	$543,087
2007	Central		$2,247,755	$393,976
	Mid-Atlantic		$9,134,630	$1,590,653
	Northeast		$3,576,010	$627,891
	Northwest		$3,556,006	$624,374

Notice that there are many more rows of data that are within the top 10% of profit. This is because Category is now included in the calculation of the metric qualification. While this evaluation option for metric qualifications returns a different type of analysis, the same analysis can be achieved by simply adding all attributes from the Report Objects pane onto the report grid, so that all attributes are then present on the report grid.

Copyright © 2024 All Rights Reserved

The following information should be taken into consideration when choosing an evaluation level for a metric qualification. This information assumes you are familiar with report levels as explained in the Advanced Reporting Help:

- **Evaluation at the report grid level**: Evaluating metric qualifications at the level present on the report grid allows the view filter to dynamically display analysis that reflects the data available on the report grid. If all attributes are on the report grid, then this level is used to calculate the metric qualification. Additionally, anytime an attribute is moved between the Report Objects pane and the report grid, the view filter dynamically recalculates the metric qualification to reflect the new level of data on the report grid.

  You can join metric qualifications evaluated at the grid level to any other metric qualifications evaluated at the grid with logical operators, as described in *Combining view filter qualifications with operators, page 237*.

  If derived metrics are also on this report, evaluating metric qualifications at the grid level also causes the metric qualifications to be evaluated after derived metrics by default. This means that these qualifications filter the results of any derived metric calculations. For more information, see *View filter effects on derived metrics, page 239*.

- **Evaluation at the report level**: Evaluating metric qualifications at the report level, regardless of what attributes are on the report grid or the Report Objects pane, provides a consistent level of analysis during dynamic aggregation.

  You can also join metric qualifications evaluated at the report level to attribute qualifications or other metric qualifications evaluated at the report level with logical operators, as described in *Combining view filter qualifications with operators, page 237*.

  If derived metrics are also on this report, evaluating metric qualifications at the report level also causes the metric qualifications to be evaluated before derived metrics by default. This means that these qualifications

Copyright © 2024 All Rights Reserved

filter data before any derived metric calculations are applied. For more information, see *View filter effects on derived metrics, page 239*.

- **Metric-to-metric qualifications**: The evaluation level of metric-to-metric qualifications cannot be modified. All metric-to-metric qualifications are evaluated at the report level.

The steps below show you how to modify the evaluation of metric qualifications in a view filter.

Prerequisites

- A report with a metric qualification in the view filter that is not a metric-to-metric qualification.
- To observe how this modification can affect report results, the report should also have some attributes in the Report Objects pane, but not on the report grid.

You cannot modify the evaluation level of metric-to-metric qualifications. All metric-to-metric qualifications are evaluated at the report level.

To modify the evaluation level of metric qualifications in a view filter

1. Log in to a project in MicroStrategy Developer. For steps to modify the evaluation level of metric qualifications in a view filter in MicroStrategy Web, see the MicroStrategy Web Help.
2. Run the report in Grid View.
3. Choose **View** > **View Filter**.
4. In the **View Filter** area, right-click a metric qualification and select one of the following options, which you can switch between:
   - **Apply Condition at the Grid Level** (default): Evaluates the metric qualification only for the attributes included on the report grid.

Copyright © 2024 All Rights Reserved

Attributes in the Report Objects pane but not on the report grid are not included in the metric qualification evaluation.

- **Apply Condition at the {*attributes in Report Objects*} Level**: Evaluates the metric qualification for all attributes included in the Report Objects pane, regardless of whether they are displayed on the report grid.

You can choose different evaluation options for separate metric qualifications in the same view filter.

5. Select **Apply**.

The report is updated. The report data is restricted as defined by the view filter.

Combining view filter qualifications with operators

When a view filter has multiple qualifications at the same output level, they are always combined by operators. When qualifications are combined, operators govern the interaction between different filtering conditions. Whenever you have more than one qualification in a view filter at the same output level, you can define the operator as any of the following:

- AND
- OR
- OR NOT
- AND NOT

For information and examples on how each of these operators govern interaction between filter qualifications, see the Basic Reporting Help.

You cannot change the logical operator between two metric qualifications if all of the following are true:

Copyright © 2024 All Rights Reserved

- The metric qualifications use two different metrics (for example, Revenue in Qualification 1 and Profit in Qualification 2).
- The metric qualifications are not metric-to-metric qualifications, but instead compare the metrics to numeric values.
- The metric qualifications are evaluated at the grid level.

Because the output level of view filter qualifications determines which qualifications can be joined with logical operators, by default all attribute qualifications are at the same output level as other attribute qualifications, and all metric qualifications are at the same output level as other metric qualifications. This mean that an attribute qualification can be joined with other attribute qualifications, and metric qualifications can be joined with other metric qualifications, but attribute and metric qualifications cannot be joined.

However, you can modify the output level of metric qualifications. If you modify a metric qualification to be evaluated at the report level, the metric qualification can be joined with attribute qualifications and with any other metric qualifications that have been defined to be evaluated at the report level. Metric qualifications evaluated at the report level cannot be joined with metric qualifications evaluated at the grid level. For information on the benefits of evaluating metric qualifications at the report level versus the grid level, see *Evaluating qualifications at the report or grid level, page 232*.

Deleting a view filter

When deleting a view filter, you have the option of deleting all of the qualifications of a view filter at once, or deleting the qualifications individually. The basic steps to perform both options are given below.

Copyright © 2024 All Rights Reserved

To delete all view filter qualifications

1. In an opened report with a view filter, choose **View Filter** > **Clear**.
2. If you do not have Auto-Apply changes selected, click **Apply**. If you only perform Clear without Apply, the view filter appears in the report the next time you open it.

To delete a view filter qualification individually

1. In an opened report with a view filter, right-click the qualification and select **Remove qualification**.
2. If you do not have Auto-Apply changes selected, click **Apply**. If you only perform Clear without Apply, the view filter appears in the report the next time you open it.

View filter effects on reporting features

Applying a view filter to a report restricts the data that is displayed on the report grid. However, unlike report filters, data restricted from display by a view filter is still available for a report. This difference has an effect on how view filters interact with various reporting features. This section discusses:

- *View filter effects on derived metrics, page 239*
- *View filter effects on metrics with relative functions, page 246*
- *View filter effects on dynamic aggregation, page 248*
- *View filter effects on derived elements, page 252*

View filter effects on derived metrics

For most derived metrics, view filters have the same effects as they do on standard metrics created with the Metric Editor. When data is restricted from view, the derived metric values along with any related data is hidden from

Copyright © 2024 All Rights Reserved

the view of the report. For example, if a view filter restricts data to the year 2007, then any derived metrics and standard metrics included on the report only display data for the year 2007.

However, derived metrics created with relative functions (OLAP functions and Rank and Ntile functions) are affected by the evaluation of view filters in a unique way.

A derived metric uses the data available on a report to calculate its values. Since view filter qualifications restrict the data displayed on the report, this can also restrict the data available for derived metrics, depending on the evaluation order of the two objects. This restriction of data, combined with derived metrics with relative functions that depend on other values on the report, can effect the display of report results.

The view filter effects on derived metrics with relative functions depends on how view filters are used in such reports, as described in the sections listed below:

- *Using view filters that affect derived metrics with relative functions, page 241*: Derived metrics with relative functions such as RunningSum or Rank return values that are relative to other values on a report. When data is restricted by view filter qualifications on other objects of a report, you can allow the values of derived metrics with relative functions to calculate their relative values based on the new view of report data.
- *Using derived metrics with relative functions in view filters, page 243*: Derived metrics with relative functions such as RunningSum or Rank return values that are relative to the data on a report. When derived metrics with relative functions are used to define view filters, you can allow the relative values to reflect the new view of data or retain the relative values for all report data.

Copyright © 2024 All Rights Reserved

Using view filters that affect derived metrics with relative functions

Derived metrics with relative functions such as RunningSum or Rank return values that are relative to other values on a report. When data is restricted by view filter qualifications on other objects of a report, you can allow the values of derived metrics with relative functions to calculate their relative values based on the new view of report data.

For example, consider a report created in the MicroStrategy Tutorial project with Year, Category, Profit, Profit Margin, and a rank shortcut derived metric named Rank (Profit Margin) as shown below.

Year	Category	Metrics	Profit	Profit Margin	Rank (Profit Margin)
2005	Books		$251,345	22.04%	10
	Electronics		$2,768,984	17.43%	7
	Movies		$55,041	6.69%	1
	Music		$43,243	6.90%	5
2006	Books		$294,789	22.32%	12
	Electronics		$3,389,107	17.56%	9
	Movies		$71,068	6.88%	4
	Music		$53,666	7.17%	6
2007	Books		$344,562	22.04%	11
	Electronics		$4,129,652	17.46%	8
	Movies		$91,602	6.87%	2
	Music		$64,682	6.88%	3

Notice that the 12 rows are ranked in ascending order from 1 to 12 by their profit margin values. You then create a view filter qualification to display data only when profit is less than $1,000,000. The view filter and resulting report are shown below.

Copyright © 2024 All Rights Reserved

New Clear Auto-Apply changes

Profit Less than 1000000

Click here to start a new qualification

Year	Category	Metrics	Profit	Profit Margin	Rank (Profit Margin)
2005	Books		$251,345	22.04%	10
	Movies		$55,041	6.69%	1
	Music		$43,243	6.90%	5
2006	Books		$294,789	22.32%	12
	Movies		$71,068	6.88%	4
	Music		$53,666	7.17%	6
2007	Books		$344,562	22.04%	11
	Movies		$91,602	6.87%	2
	Music		$64,682	6.88%	3

Notice that while some data no longer appears on the report, the values of the Rank (Profit Margin) derived metric remain the same. This allows you to view the rank of profit margins as applied over all the data for the report, including the data that has been filtered from view.

Since the Rank (Profit Margin) metric is a derived metric, you can modify the report so that the metric's values are relative to the new view of data supplied by the view filter, as shown in the report below.

New Clear Auto-Apply changes

Profit Less than 1000000

Click here to start a new qualification

Year	Category	Metrics	Profit	Profit Margin	Rank (Profit Margin)
2005	Books		$251,345	22.04%	3
	Movies		$55,041	6.69%	9
	Music		$43,243	6.90%	5
2006	Books		$294,789	22.32%	1
	Movies		$71,068	6.88%	6
	Music		$53,666	7.17%	4
2007	Books		$344,562	22.04%	2
	Movies		$91,602	6.87%	8
	Music		$64,682	6.88%	7

Copyright © 2024 All Rights Reserved

You can apply this type of analysis in one of two ways:

- Evaluate the view filter qualification at the report level. This causes the view filter to be evaluated before any derived metrics. The derived metrics then calculate their data using the view of data created by the view filter. For information on evaluating view filter qualifications at the report level, see *Evaluating qualifications at the report or grid level, page 232*.
- Evaluate the view filter qualification at the grid level and change the evaluation order of derived metrics to be calculated after the view filter. For information on changing the evaluation order of individual report objects such as view filter qualifications and derived metrics, see the Advanced Reporting Help.

Using derived metrics with relative functions in view filters

Derived metrics with relative functions such as RunningSum or Rank return values that are relative to the data on a report. When data is restricted by view filter qualifications on derived metrics with relative functions, you can allow the relative values to reflect the new view of data, or retain the relative values for all report data.

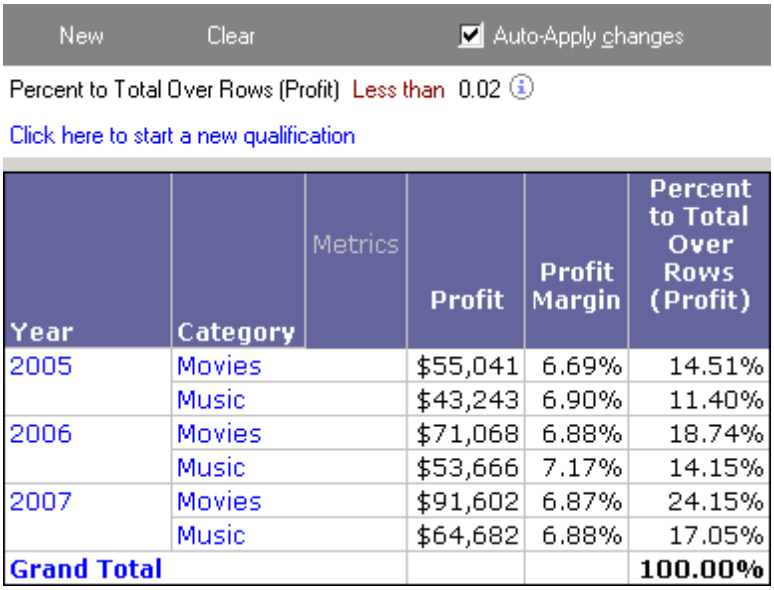
When derived metrics with relative functions are used to define view filters, you can allow the relative values to reflect the new view of data or retain the relative values for all report data.

For example, consider a report created in the MicroStrategy Tutorial project with Year, Category, Profit, Profit Margin, and a percent-to-total shortcut derived metric named Percent to Total Over Rows (Profit Margin), as shown below.

Copyright © 2024 All Rights Reserved

Year	Category	Metrics	Profit	Profit Margin	Percent to Total Over Rows (Profit)
2005	Books		$251,345	22.04%	2.17%
	Electronics		$2,768,984	17.43%	23.96%
	Movies		$55,041	6.69%	0.48%
	Music		$43,243	6.90%	0.37%
2006	Books		$294,789	22.32%	2.55%
	Electronics		$3,389,107	17.56%	29.32%
	Movies		$71,068	6.88%	0.61%
	Music		$53,666	7.17%	0.46%
2007	Books		$344,562	22.04%	2.98%
	Electronics		$4,129,652	17.46%	35.73%
	Movies		$91,602	6.87%	0.79%
	Music		$64,682	6.88%	0.56%

Notice that the percent-to-total values display the distribution of profits over all 12 rows of data. You then create a view filter qualification based on the percent-to-total derived metric to display data only when the percent-to-total profit is less than two percent (.02). The view filter and resulting report are shown below.



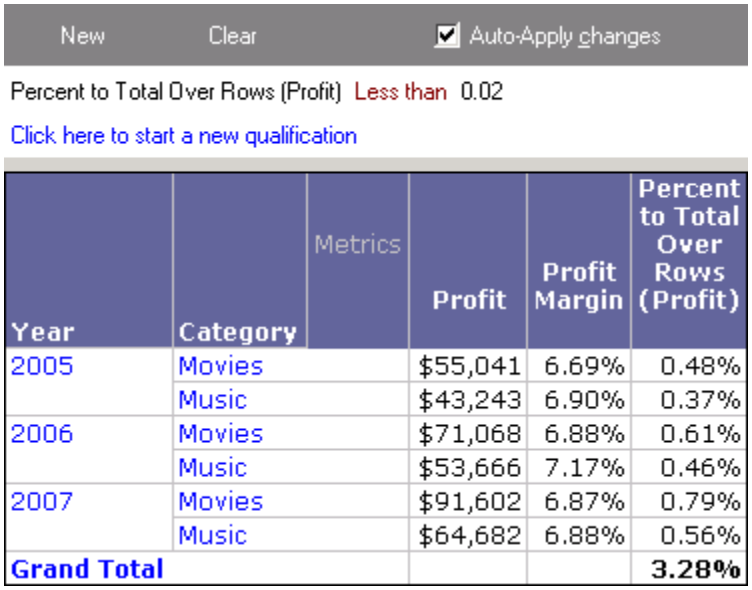
Year	Category	Metrics	Profit	Profit Margin	Percent to Total Over Rows (Profit)
2005	Movies		$55,041	6.69%	14.51%
	Music		$43,243	6.90%	11.40%
2006	Movies		$71,068	6.88%	18.74%
	Music		$53,666	7.17%	14.15%
2007	Movies		$91,602	6.87%	24.15%
	Music		$64,682	6.88%	17.05%
Grand Total					**100.00%**

While it appears that data is being shown for percent-to-totals greater than two percent, this is because the Percent to Total Over Rows (Profit) derived metric values have dynamically changed to reflect the new view of data

Copyright © 2024 All Rights Reserved

displayed on the report. A grand total is displayed to show that the derived metric values add up to 100%, even though these rows of data combined account for less than four percent of the profits of the original report. This analysis is applied because the view filter qualification is evaluated at the report level by default.

The report above provides a view of percent-to-total profit data as displayed on the report. However, view filter qualifications on derived metrics can also be evaluated at the grid level so that the derived metrics retain their values that reflect all data available for the report, as shown in the report below.



Year	Category	Metrics	Profit	Profit Margin	Percent to Total Over Rows (Profit)
2005	Movies		$55,041	6.69%	0.48%
	Music		$43,243	6.90%	0.37%
2006	Movies		$71,068	6.88%	0.61%
	Music		$53,666	7.17%	0.46%
2007	Movies		$91,602	6.87%	0.79%
	Music		$64,682	6.88%	0.56%
Grand Total					**3.28%**

Notice that the percent-to-total profit values now appear to be less than two percent and reflect the values of the original report that included all report data. This is also reflected in the grand total of 3.28%.

As these scenarios illustrate, you have two options to evaluate view filter qualifications based on derived metrics with relative functions, summarized below:

- Evaluate the view filter qualification at the report level (default). This causes the derived metric values to dynamically reflect the new view of data on the report after the view filter qualification is applied.

Copyright © 2024 All Rights Reserved

- Evaluate the view filter qualification at the grid level. This causes the derived metric to retain its values that reflect all data available for the report.

For information on evaluating view filter qualifications at the report level versus the grid level, see *Evaluating qualifications at the report or grid level, page 232*.

View filter effects on metrics with relative functions

Metrics with relative functions (OLAP functions and Rank and Ntile functions) such as RunningSum or Rank return values that are relative to other values on a report. When data is restricted by view filter qualifications, some of the report data is hidden from view while the data available is not changed. This can cause the values returned by metrics with relative functions to appear to have incorrect values for the data displayed on the report.

For example, consider a report with Customer Group, Year, Average Net Sales Order Amount per Customer, and RunningAvg (Average Net Sales Order Amount per Customer), as shown below.

Customer Group	Year	Metrics	Average Net Sales Order Amount per Customer	RunningAvg (Avg Net Sales Order Amount per Customer)
Retailers	2006		$6,019,000	$6,019,000
	2007		$5,024,400	$5,521,700
	2008		$460,000	$3,834,467
Wholesalers	2006		$575,000	$575,000
	2007		$1,695,100	$1,135,050
	2008		$750,000	$1,006,700
Large Company	2006		$225,750	$225,750
	2007		$3,324,333	$1,775,042
	2008		$3,934,375	$2,494,819
Medium Company	2006		$116,500	$116,500
	2007		$409,913	$263,206
	2008		$114,000	$213,471

Copyright © 2024 All Rights Reserved

You then create a view filter to restrict data to only years 2007 and 2008. The view filter qualification and resulting report are shown below.

Customer Group	Year	Metrics	Average Net Sales Order Amount per Customer	RunningAvg (Avg Net Sales Order Amount per Customer)
Retailers	2007		$5,024,400	$5,521,700
	2008		$460,000	$3,834,467
Wholesalers	2007		$1,695,100	$1,135,050
	2008		$750,000	$1,006,700
Large Company	2007		$3,324,333	$1,775,042
	2008		$3,934,375	$2,494,819
Medium Company	2007		$409,913	$263,206
	2008		$114,000	$213,471

Notice that the values for the RunningAvg (Average Net Sales Order Amount per Customer) metric do not change, only the 2006 values are hidden. The values displayed do not accurately reflect the view of data and instead reflect the data available for the entire report, including the 2006 data hidden from view.

This is because the view filter is calculated without re-executing SQL against the data warehouse, and is evaluated after calculating the metrics on the report. Therefore, the metric is not recalculated to reflect the view of data shown on the report as restricted by the view filter.

If you plan to use metrics with relative functions and require them to reflect the data displayed on the report, you can use one of the options summarized below:

- Use report filters rather than view filters. A report filter causes a report to re-execute its SQL against the data warehouse, which can cause more processing time than a view filter. However, this allows a metric with a

Copyright © 2024 All Rights Reserved

relative function to recalculate its values based on the filtering criteria.

- Use derived metrics with relative functions rather than standard metrics. Derived metrics can be evaluated after view filters and thus reflect the view of data on a report without having to re-execute SQL. However, derived metrics cannot be saved for use in multiple reports and can require modification to work as intended with view filters. For information on the interaction between view filters and derived metrics, see *View filter effects on derived metrics, page 239*.

View filter effects on dynamic aggregation

Dynamic aggregation occurs when an attribute is moved between the report layout and the Report Objects pane. This type of aggregation dynamically affects the data returned for the report. For more information on dynamic aggregation, refer to *Chapter 9, Dynamic Aggregation*.

A view filter also dynamically affects the data returned for a report, by restricting the report data using various types of qualifications. You can use both view filters and dynamic aggregation to return different analyses of report data, as described in the sections listed below:

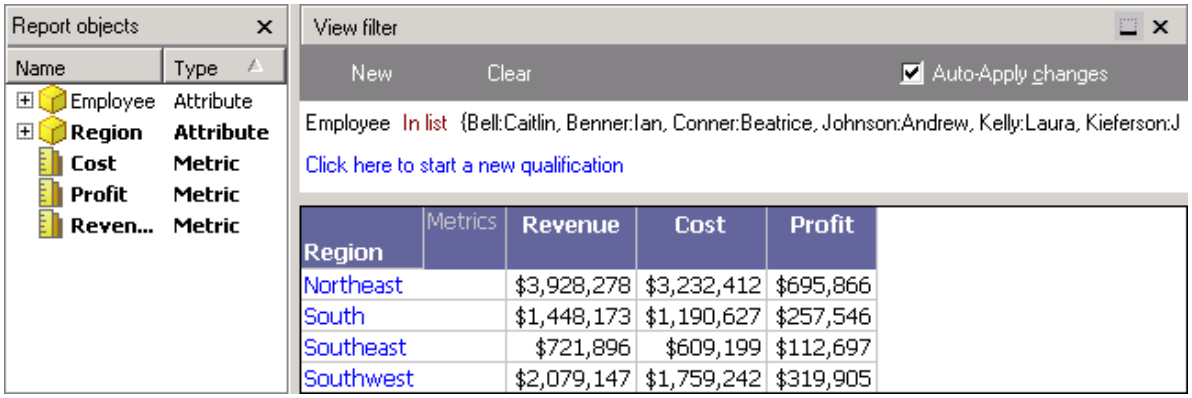
- *Using attribute qualifications on hidden attributes, page 248*: When report data is restricted using a view filter attribute qualification, this can affect the data displayed on the report even if the attribute it qualifies on is not displayed on the report grid.
- *Using metric qualifications at the report or grid level, page 250*: When report data is restricted using a view filter metric qualification, this can affect the data displayed on the report in various ways depending on when the view filter metric qualification is evaluated.

Using attribute qualifications on hidden attributes

When report data is restricted using a view filter attribute qualification, this can affect the data displayed on the report even if the attribute it qualifies on is not displayed on the report grid.

Copyright © 2024 All Rights Reserved

For example, consider the View Filter - Dynamic Aggregation report in the MicroStrategy Tutorial project, as shown below.



Notice that dynamic aggregation has been used to remove the Employee attribute from the report grid. However, there is also a view filter attribute qualification on the Employee attribute. Even though Employee is not shown on the report, the data displayed is restricted by the view filter to only display data for the employees Caitlin Bell, Beatrice Conner, Andrew Johnson, Laura Kelly, and Jack Kieferson. This can be verified by using dynamic aggregation to drag and drop Employee on the report grid, as shown below.

Region	Employee		Metrics	Revenue	Cost	Profit
	Kelly	Laura		$3,135,209	$2,579,374	$555,835
Northeast	Kieferson	Jack		$793,070	$653,038	$140,031
	Total			**$3,928,278**	**$3,232,412**	**$695,866**
South	Conner	Beatrice		$1,448,173	$1,190,627	$257,546
	Total			**$1,448,173**	**$1,190,627**	**$257,546**
Southeast	Benner	Ian		$721,896	$609,199	$112,697
	Total			**$721,896**	**$609,199**	**$112,697**
	Bell	Caitlin		$1,448,086	$1,223,908	$224,178
Southwest	Johnson	Andrew		$631,060	$535,333	$95,727
	Total			**$2,079,147**	**$1,759,242**	**$319,905**

Data is dynamically aggregated to be displayed at the region and employee level. A subtotal has been added to the report above to show that the original report was only displaying data for the employees listed in the view filter attribute qualification. For example, notice that the revenue, cost, and

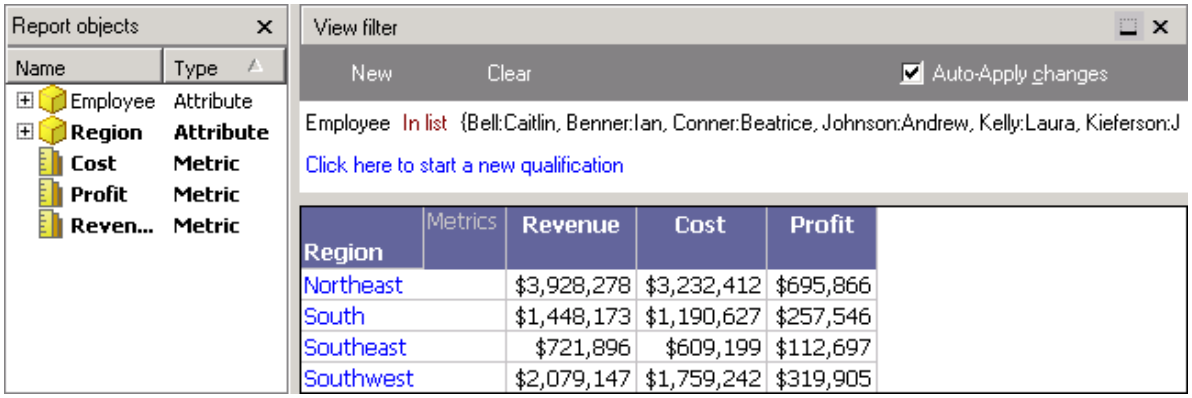
Copyright © 2024 All Rights Reserved

profit data for the Northeast region of the original report matches the values for the Northeast subtotal values of the report above.

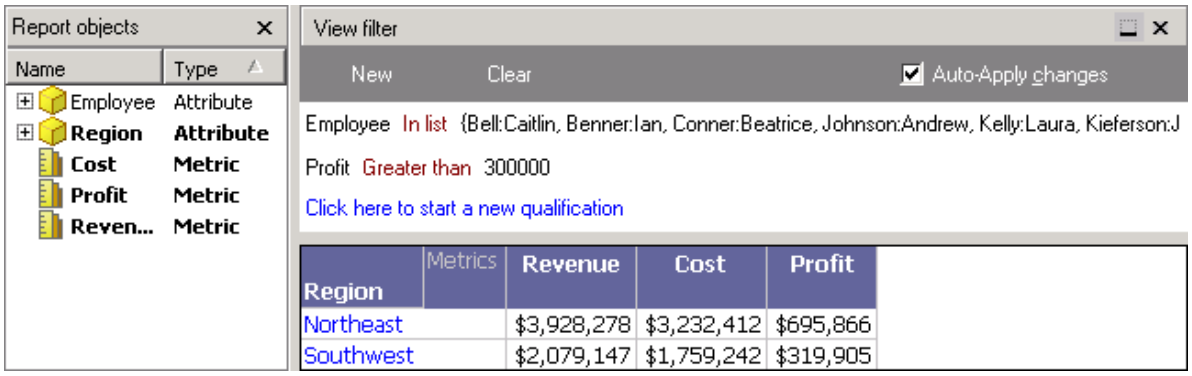
Using metric qualifications at the report or grid level

When report data is restricted using a view filter metric qualification, this can affect the data displayed on the report in various ways depending on when the view filter metric qualification is evaluated.

For example, consider the View Filter - Dynamic Aggregation report in the MicroStrategy Tutorial project, as shown below.



Notice that dynamic aggregation has been used to remove the Employee attribute from display on the report grid. You then create a view filter metric qualification to restrict the report data for profits greater than $300,000. The view filter qualification and resulting report are shown below.

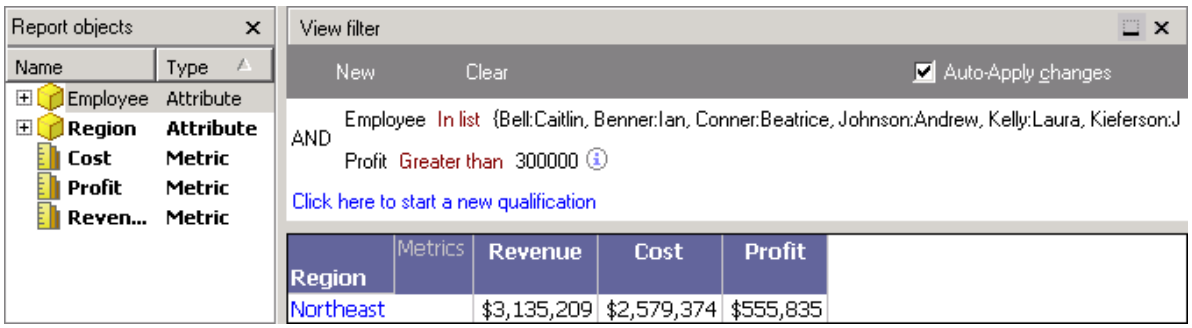


Copyright © 2024 All Rights Reserved

The view filter metric qualification is evaluated at the grid level by default. This means that the dynamic aggregation of removing Employee from the report layout is evaluated first, and then the view filter metric qualification restricts data based on the remaining data displayed on the report.

With this evaluation order, the view filter metric qualification returns regions with profits greater than $300,000.

However, you can also evaluate view filter metric qualifications at the report level. You can right-click the Profit Greater than $300,000 qualification, and select **Apply Condition at the {Employee, Region} level**. Evaluating the metric qualification at the report level returns the report results shown below.



Notice that only data for the Northeast region is returned, and all metric values are lower. This is because a view filter metric qualification evaluated at the report level includes all attributes, in the calculation to restrict data from the report.

Data is first restricted to employees with profits greater than $300,000, and then this data is aggregated and displayed at the region level. In the first report the Southwest region included two employees who combined to have more than $300,000 in profit, but neither employee had more than $300,000 alone. These employees are restricted by the view filter and are not included when aggregating the data at the region level.

These two options provide two different types of analysis on report data, summarized below:

Copyright © 2024 All Rights Reserved

- **Evaluating view filter metric qualifications at the grid level**: When view filter metric qualifications are evaluated at the grid level and dynamic aggregation is used, only the attributes displayed on the report grid are used to determine the data restricted from the report.

  In the example above, any regions with profits greater than $300,000 for the included employees are displayed on the report.

- **Evaluating view filter metric qualifications at the report level**: When view filter metric qualifications are evaluated at the report level and dynamic aggregation is used, all attributes in the Report Objects pane are used to determine the data restricted from the report. This includes attributes that are not on the report grid.

  In the example above, any employees with profits greater than $300,000 for the included employees are displayed on the report. The data for any remaining employees is then aggregated and displayed at the region level.

For information on evaluating view filter qualifications at the report level versus the grid level, see *Evaluating qualifications at the report or grid level, page 232*.

View filter effects on derived elements

View filters restrict data on reports by enabling users to create qualifications on attributes and metrics. These qualifications then determine what data is shown on the report, as well as what data is available to analyze and format with derived elements.

Derived elements cannot be selected in a view filter, and therefore cannot be a part of a view filter qualification that restricts the report results. View filters can only include qualifications on attributes and metrics.

However, derived elements are associated with attribute elements. Due to this association, any view filter qualifications that restrict the data for a derived element's associated attribute elements also restricts data returned for the derived element.

Copyright © 2024 All Rights Reserved

For example, a report includes a Group derived element defined to combine profit values for the Music and Movies elements of the Category attribute. This report is shown below.

Category	Metrics	Profit
Music and Movies		$379,300
Books		$890,696
Electronics		$10,287,744

The Music and Movies derived element combines the profit values for the Music attribute element and Movies attribute element. You then create a view filter qualification that restricts the report data to the individual Movies attribute element and Electronics attribute element. The report results are shown below.

View filter

New Auto-Apply changes

Category In list {Electronics, Movies}

Click here to start a new qualification

Category	Metrics	Profit
Music and Movies		$217,710
Electronics		$10,287,744

Notice that the Music and Movies derived element is still displayed, but the profit value has decreased. This is because the view filter has restricted the data to only Movies and Electronics. The Music and Movies derived element can only return profit values for the available Movies attribute element data. When using view filters and derived elements on the same report, be aware that any view filter qualifications restrict the data available to analyze and format with derived elements.

Copyright © 2024 All Rights Reserved

DYNAMIC SOURCING

Copyright © 2024 All Rights Reserved

The general goal of reporting on data in Intelligent Cubes is to create reports that are linked to a published Intelligent Cube. Dynamic sourcing makes Intelligent Cubes more accessible by allowing regular reports to automatically access published Intelligent Cubes that can satisfy the data requirements of the report.

This can improve the performance of reports, which get executed against an Intelligent Cube, without having to manually find and select the Intelligent Cube that fits the analysis of the report. If the data requirements for the report change, or cannot be satisfied by an available Intelligent Cube, the reports automatically access the data warehouse to return the required information.

To support dynamic sourcing, you will need to configure your projects, Intelligent Cubes, reports, attributes, metrics, and any aggregate tables you use. This appendix describes the benefits of dynamic sourcing, the necessary steps to support dynamic sourcing, and how to track how successful dynamic sourcing is for your project. This section assumes you are familiar with the information in *Chapter 2, Sharing Sets of Data Among Reports: Intelligent Cubes*.

Scenarios that benefit from dynamic sourcing

Dynamic sourcing can greatly improve the performance of your projects, because it allows reports to automatically detect an Intelligent Cube that can satisfy its data requirements, and execute against the Intelligent Cube rather than the data warehouse.

This section also discusses how you can tune your project to better support dynamic sourcing for these features, as described in *Tuning your project for dynamic sourcing, page 256*.

Dynamic sourcing complements the ability to create reports that are connected to a specific Intelligent Cube. This feature gives you the following performance benefits:

Copyright © 2024 All Rights Reserved

- Report designers do not need to know whether an Intelligent Cube includes the information they need, or which Intelligent Cube they should use. With dynamic sourcing, an Intelligent Cube that can satisfy the report's data requirements is detected automatically, without the report designer having to consider which Intelligent Cubes are available.
- The performance of pre-existing reports can be improved without having to modify the report to access a specific Intelligent Cube. Dynamic sourcing can allow these reports to automatically detect an Intelligent Cube that satisfies the report's data requirements.
- The performance of prompted reports can be greatly improved. Prompted reports can cause performance issues, because it is difficult to use report caches with them. When different prompt answers are chosen, a report cache cannot return information for the report and the report request must be submitted through the data warehouse again.
- With dynamic sourcing, Intelligent Cubes can provide a set of data that can satisfy the data requirements of reports executed with different prompt answers.
- Reports can drill from one Intelligent Cube to another Intelligent Cube.

Tuning your project for dynamic sourcing

You can tune your project to provide a reporting environment that is more likely to be able to take advantage of dynamic sourcing, as described in the sections listed below.

For information on the various features that are supported for dynamic sourcing, see *Best practices for supporting dynamic sourcing, page 257*.

Tuning Intelligent Cubes for dynamic sourcing

One of the most important aspects in providing dynamic sourcing for as many reports as possible in your project is the creation and availability of Intelligent Cubes. Once Intelligence Server determines which Intelligent

Copyright © 2024 All Rights Reserved

Cubes can satisfy a report's needs, it automatically selects the Intelligent Cube that offers the best performance for the report.

You can improve the chances that a report will use dynamic sourcing by making more Intelligent Cubes available to it. When creating additional Intelligent Cubes, however, you must consider that Intelligent Cubes can consume a significant amount of Intelligence Server's memory. If you can determine the objects and type of analysis most commonly used in reports, you can use this information to tailor the Intelligent Cubes to these requirements.

Be aware that enabling dynamic sourcing requires some system overhead to determine whether a report can access an Intelligent Cube rather than the data warehouse. This overhead increases as the number of Intelligent Cubes enabled for dynamic sourcing increases. However, the overhead only affects Intelligence Server's performance if there are over a thousand Intelligent Cubes, which is an unlikely business scenario.

Tracking the success of dynamic sourcing

If you enable dynamic sourcing for a project, you can track various information related to dynamic sourcing that can help determine why dynamic sourcing succeeded or failed for reports. By determining why dynamic sourcing failed for a given report, you can modify your reports, Intelligent Cubes, and other objects to allow reports in your project to use dynamic sourcing. Tracking dynamic sourcing information is described in *Tracking the use of dynamic sourcing, page 300*.

Best practices for supporting dynamic sourcing

For a report to successfully retrieve its information from an Intelligent Cube, various conditions must be met. The considerations and steps required to fully support dynamic sourcing are listed below:

Copyright © 2024 All Rights Reserved

- While many MicroStrategy features can be used with dynamic sourcing, some features may prevent the use of dynamic sourcing between reports and Intelligent Cubes. Refer to the following sections for a list of features that can and cannot be used with dynamic sourcing:
  - *Features that can be used with dynamic sourcing, page 260*
  - *Features that prevent the use of dynamic sourcing, page 261*
- You can use Cube Advisor to create Intelligent Cubes that will allow as many reports as possible to use dynamic sourcing. For steps to use Cube Advisor to support dynamic sourcing, see *Using Cube Advisor to support dynamic sourcing, page 287*.
- For an Intelligent Cube to satisfy the data requirements of a report, the Intelligent Cube and report must have matching settings for VLDB properties that can affect the data that is returned.
- If you use dynamic sourcing, you must keep your Intelligent Cubes up to date with any changes to the data in your data warehouse, otherwise reports can return outdated data. You can use schedules and subscriptions to update the data of Intelligent Cubes. For information on these techniques, see *Updating Intelligent Cubes using schedules, page 35* and *Publishing Intelligent Cubes using a schedule, page 45*.
- In general, you should not create restrictive filter qualifications on Intelligent Cubes. The more restrictive the filter, the less likely it is that the Intelligent Cube can satisfy the data requirements of reports. Restrictive filters can be applied to the reports that access Intelligent Cubes.
- Reports cannot combine the data from multiple Intelligent Cubes to use dynamic sourcing. All the data requirements of a report must be satisfied by a single Intelligent Cube.
- Dynamic sourcing must be enabled for projects, reports, and Intelligent Cubes; this is described in the sections listed below:

Copyright © 2024 All Rights Reserved

- *Enabling or disabling dynamic sourcing for projects, page 266*
- *Enabling or disabling dynamic sourcing for reports, page 273*
- *Enabling or disabling dynamic sourcing for Intelligent Cubes, page 274*

- To ensure that correct data is available in an Intelligent Cube for a report, you must verify that the following objects can support dynamic sourcing:
  - Attributes are available for dynamic sourcing by default. You should disable dynamic sourcing for attributes if:
    - Attribute data in fact and lookup tables contains NULL values.
    - The attribute elements in fact and lookup tables are not identical.

    For steps to disable dynamic sourcing for attributes, see *Disabling dynamic sourcing for attributes, page 277*.
  - Metrics are available for dynamic sourcing by default. You should disable dynamic sourcing for metrics if metric data in fact tables contains NULL values. For steps to disable dynamic sourcing for metrics, see *Disabling dynamic sourcing for metrics, page 282*.
  - Aggregate tables are available for dynamic sourcing by default. You should disable dynamic sourcing for aggregate tables if:
    - Aggregation functions other than Sum are used.
    - The aggregate table includes different data than is available in lookup and fact tables. For example, an aggregate table with years 2006, 2007, and 2008 should not be used for dynamic sourcing if your lookup and fact tables only include the years 2007 and 2008.

    For steps to disable dynamic sourcing for aggregate tables, see *Disabling dynamic sourcing for aggregate tables, page 284*.
- If the report is based on an MDX data source, such as an SAP BW Cube, it can also use dynamic sourcing, and retrieve data from an Intelligent Cube that is also based on an MDX data source.

Copyright © 2024 All Rights Reserved

In such a scenario, the following additional conditions apply:

- The report and Intelligent Cube must be based on the same source MDX cube.
- Filters on the report must meet the following criteria:
  - Attributes used in the filter should also be on the Intelligent Cube's definition.
  - The report filter must include at least the same restrictions as the Intelligent Cube's filter, if present. For example, if the Intelligent Cube restricts data to only the year 2010, the report must include the same restriction.

For additional information on creating reports that access MDX sources, refer to the MDX Cube Reporting Help.

Features that can be used with dynamic sourcing

Features that can be supported by dynamic sourcing include, but are not limited to, the list below:

- Security filters
- Compound attributes and joint element lists
- Conditional metrics
- Prompts in a report
- Transformation metrics, under the following conditions:
  - A metric with the same formula and transformation exists in the Intelligent Cube.
  - For the hierarchy used for the transformation, the report must be calculated at the same level as the Intelligent Cube. For example, if the transformation is along the Time hierarchy and the Intelligent Cube is calculated at the Month level, the report must also be calculated at the Month level.

Copyright © 2024 All Rights Reserved

- Metrics that include Count, Sum, Min, or Max in their formulas
- Metrics that include Count Distinct in their formula for reports and Intelligent Cubes that share the same level of attribute data
- Nested metrics for which the inner metric and all attributes used for aggregation are available in the Intelligent Cube
- Non-aggregatable metrics for which the non-aggregation attribute is available in the Intelligent Cube
- Filter qualifications including the functions Exactly, Different from, In list, Not in list, Greater than, Less than, Greater than or equal to, Less than or equal to, Between, Not Between, Is Null, and Is Not Null for data types Numeric, BigDecimal, DateTime, and String
- OLAP Services features in the report, including:
  - View filters
  - Derived metrics
  - Dynamic aggregation
- OLAP features used in a report, such as:
  - Pivoting
  - Banding
  - Thresholds
  - Page-by
  - Outline mode
  - Sorting
  - Subtotals

Features that prevent the use of dynamic sourcing

Some features prevent the use of dynamic sourcing between Intelligent Cubes and report, as described in the sections listed below:

Copyright © 2024 All Rights Reserved

- *Features that prevent reports from using dynamic sourcing, page 262*
- *Features that prevent Intelligent Cubes from being available for dynamic sourcing, page 264*

In addition to these features, if you use filter qualifications on attribute forms in reports and Intelligent Cubes, a mismatch in the attribute forms used in the qualifications can prevent dynamic sourcing between the report and Intelligent Cube. For example, a report includes a filter qualification on the ID form of the Customer attribute. An Intelligent Cube includes a filter qualification on the Description form of the Customer attribute. Since different attribute forms are used to restrict the data, a direct relationship between the data cannot be verified and dynamic sourcing cannot be used to connect the report to the Intelligent Cube.

You can track various information related to dynamic sourcing that can help determine why dynamic sourcing succeeded or failed for reports, as described in *Tracking the use of dynamic sourcing, page 300*.

Features that prevent reports from using dynamic sourcing

If any of the features listed below are present in reports, the reports may not be able to use dynamic sourcing:

- Consolidations
- Custom groups
- Query Builder or Freeform SQL reports cannot use dynamic sourcing.
- Set qualifications in the report filter or metric conditions:
  - Metric qualifications
  - Relationship qualifications that do not include a filter in their definition, or that use a Relate By option other than Use System Default.
- OR logical operator used in the report filter to combine qualifications which include different attributes or attribute forms.

Copyright © 2024 All Rights Reserved

- AND NOT or OR NOT logical operators
- Metrics that use passthrough functions such as ApplySimple. Alternatively, facts can be defined using passthrough functions such as ApplySimple. These facts can then be used to create metrics, which can be used on reports that can support dynamic sourcing.
- Freeform MDX metrics. On reports based on MDX data sources, these are analogous to metrics with passthrough functions.
- Metrics that use facts with extensions or degradations
- Data marts
- Report as filter used in the report filter
- Using any of the options listed below for the following VLDB properties:

VLDB property	Options that prevent report from using dynamic sourcing
Downward outer join	Preserve all rows for metrics higher than template level without report filter
	Do not do downward outer join for database that support full outer join
OLAP function support	Preserve backward compatibility with 8.1.x and earlier This setting prevents the report from using dynamic sourcing only under the following conditions: The report has metrics that use Rank or NTile functions The report has metrics that use functions with a SortBy parameter For information on functions and function parameters, refer to the Functions Reference.

Copyright © 2024 All Rights Reserved

Features that prevent Intelligent Cubes from being available for dynamic sourcing

If the features listed below are present in an Intelligent Cube, the Intelligent Cube may not be available for dynamic sourcing:

- Set qualifications in the Intelligent Cube filter:
  - Metric qualifications
  - Relationship qualifications
- OR logical operator used in the Intelligent Cube filter to combine qualifications which include different attributes or attribute forms
- AND NOT or OR NOT logical operators used in the Intelligent Cube filter
- Passthrough functions such as ApplySimple
- Report as filter used in the Intelligent Cube filter
- Report limits
- For Intelligent Cubes based on MDX data sources, dynamic sourcing is not supported if the Intelligent Cube uses incremental refresh settings. For information on incremental refresh, see *Updating Intelligent Cubes without re-processing: Incremental Refresh, page 48*.

If the features listed below are present in an Intelligent Cube the features may not be available for dynamic sourcing, but the rest of the Intelligent Cube may be able to support dynamic sourcing:

- Metrics that use facts with fact extensions or degradations are not available for dynamic sourcing.
- Conditional metrics are not available for dynamic sourcing. Conditional metrics are described in the Advanced Reporting Help.
- Compound metrics that are not defined as smart metrics in an Intelligent Cube are not available for dynamic sourcing.

Copyright © 2024 All Rights Reserved

Dynamic sourcing for incrementally refreshed Intelligent Cubes

Apart from the best practices listed above, additional considerations apply when you are using Intelligent Cubes that are incrementally refreshed. For information on incrementally refreshing Intelligent Cubes, see *Updating Intelligent Cubes without re-processing: Incremental Refresh, page 48*.

Intelligent Cube refresh options supported for dynamic sourcing

The following Intelligent Cube refresh options are supported for dynamic sourcing:

- Full refresh
- Dynamic refresh
- Insert and Update, only if data is being updated for one attribute form

For information on Intelligent Cube refresh settings, see *Defining Intelligent Cube refresh settings, page 50*.

Incremental refresh types supported for dynamic sourcing

You can create an incremental refresh filter or report to refresh the data in an Intelligent Cube. If the Intelligent Cube is refreshed using a filter, the following increment types are supported:

- Update Only
- Insert, Update and Delete, only if the data is being updated for one attribute form

If the Intelligent Cube is refreshed using an incremental refresh report, it cannot be used for dynamic sourcing. For information on incremental refresh filters and reports, see *Defining an incremental refresh filter or report, page 52*.

Copyright © 2024 All Rights Reserved

Configuring dynamic sourcing

To enable reports to dynamically connect to an Intelligent Cube for their data requirements, you must enable various aspects of the MicroStrategy system for dynamic sourcing.

In general, you must enable projects, reports, and Intelligent Cubes for dynamic sourcing. Depending on the objects available in or accessed by the reports and Intelligent Cubes, you may also need to define other objects as available for dynamic sourcing. The sections listed below describe the requirements and considerations for configuring dynamic sourcing for various aspects of the MicroStrategy system:

Enabling or disabling dynamic sourcing for projects

By default, dynamic sourcing is enabled for projects. However, you can disable it for a project to prevent reports from connecting to Intelligent Cubes through the use of dynamic sourcing. This allows project administrators to have project-wide control over the use of dynamic sourcing.

The steps below show you how to enable or disable dynamic sourcing for projects as well as define default dynamic sourcing behavior for Intelligent Cubes.

For information on defining default dynamic sourcing behavior for reports, Intelligent Cubes, attributes, metrics, and aggregate tables, see *Accessing the dynamic sourcing VLDB properties for a project, page 268*.

Enabling or disabling dynamic sourcing for a project

1. In MicroStrategy Developer, log in to a project source with a user account that has administrative privileges.
2. Right-click a project and select **Project Configuration**. The Project Configuration Editor opens.

Copyright © 2024 All Rights Reserved

3. In the **Categories** list, expand **Intelligent Cubes,** and then select **General**.
4. Select or clear the **Enable dynamic sourcing** check box to either enable or disable dynamic sourcing.
5. You can define a default dynamic sourcing behavior for all Intelligent Cubes in a project using the check boxes described below:

   - **Make Intelligent Cubes available for dynamic sourcing by default**: Select this check box to enable dynamic sourcing for all Intelligent Cubes in a project. You can clear this check box to disable dynamic sourcing as the default behavior for all Intelligent Cubes in a project.

     You can enable and disable dynamic sourcing for individual Intelligent Cubes, as described in *Enabling or disabling dynamic sourcing for Intelligent Cubes, page 274*.

   - **Allow dynamic sourcing even if outer join properties are not set**: Select this check box to allow reports to connect to Intelligent Cubes using dynamic sourcing even when some outer join properties are not defined. However, this can cause reports to return incorrect data in scenarios when outer joins would be helpful.

     For example, a report includes the attribute Day and the metric Revenue. It connects to an Intelligent Cube that includes the attribute Day, and the metrics Revenue and Cost. For some days there is data for Revenue, but no data for Cost. If the Intelligent Cube does not support any outer joins, then the data for Revenue for which there is no data for Cost is not included in the final result. In this scenario, the report cannot return complete information from the Intelligent Cube without outer joins.

     It is recommended that you define your Intelligent Cube to support and use outer joins when necessary, as described in *Enabling or*

Copyright © 2024 All Rights Reserved

disabling dynamic sourcing for Intelligent Cubes, page 274. This ensures that all data is returned. Be aware that outer joins can cause additional load on your database and may require larger Intelligent Cubes.

6. Click **OK** to save your changes.

You can track various information related to dynamic sourcing that can help determine why dynamic sourcing succeeded or failed for reports, as described in *Tracking the use of dynamic sourcing, page 300*.

Accessing the dynamic sourcing VLDB properties for a project

A number of VLDB properties control whether dynamic sourcing is enabled for reports, attributes, metrics, and aggregate tables. Defining these VLDB properties at the project level allows you to define a default dynamic sourcing behavior for these objects. While this defines the default behavior, any modifications to the VLDB properties for a specific report, attribute, or metric take precedence over the project-wide default.

The steps below show you how to access the dynamic sourcing VLDB properties for a project to define project-wide defaults, and includes links to information on how to set VLDB properties for each object type.

To define dynamic sourcing VLDB properties for a project

1. In MicroStrategy Developer, log in to a project source with a user account with administrative privileges.
2. Right-click a project and select **Project Configuration**. The Project Configuration Editor opens.
3. In the **Categories** list, expand **Project definition**, and then select **Advanced**.
4. In the **Analytical Engine VLDB Properties** area, click **Configure**. The VLDB Properties Editor opens.

Copyright © 2024 All Rights Reserved

5. Choose **Tools** > **Show Advanced Settings** option, if it is not already selected.
6. In the **VLDB Settings** list, expand **Dynamic Sourcing**. The dynamic sourcing VLDB properties are displayed. For information on how each VLDB property can be used to enable or disable dynamic sourcing, refer to the sections listed below:
   - **Enable Dynamic Sourcing for Report**: *Enabling or disabling dynamic sourcing for reports, page 273*
   - **Attribute Validation**: *Disabling dynamic sourcing for attributes, page 277*
   - **Metric Validation**: *Disabling dynamic sourcing for metrics, page 282*
   - **Aggregate Table Validation**: *Disabling dynamic sourcing for aggregate tables, page 284*
   - **String Comparison Behavior**: *Supporting filtering on attributes for dynamic sourcing, page 280*
7. Click **Save and Close** and then **OK**.

Improving report and cube performance against the data warehouse by using an attribute form caching cube

The data fetch time is often too high for reports and cubes that are executed against the data warehouse as most of the data transfer time (for example, network bandwidth) is spent in fetching text columns that are non-ID attribute forms.

To reduce data transfer time used for fetching non-ID attribute forms, you can use an attribute form caching Intelligent cube. Cache the attribute forms using this cube. When there is requirement to fetch non-ID attribute forms, the attribute form caching cube is used instead of database.

To learn more about attribute form caching cubes, see KB484994.

Copyright © 2024 All Rights Reserved

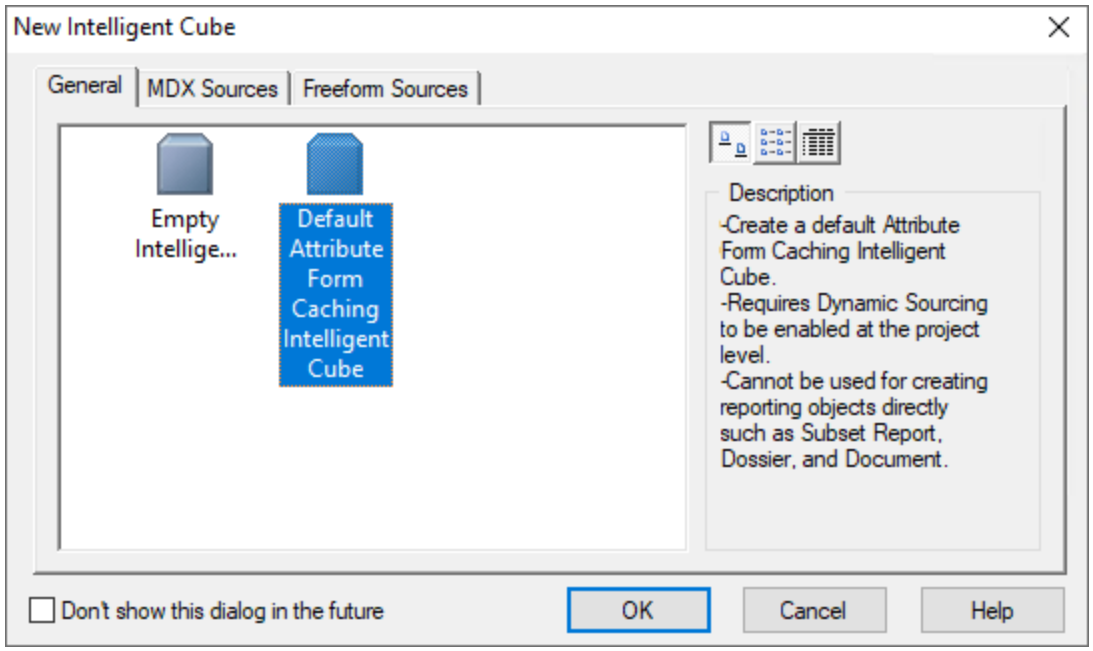
To use this functionality in Workstation, see Improve Report and Cube Performance Against the Data Warehouse by Using an Attribute Form Caching Cube.

Prerequisites

- Your metadata must be upgraded to MicroStrategy 2021 or above to populate the new cube template.
- You must enable dynamic sourcing at the project level to access the attribute form caching cube.

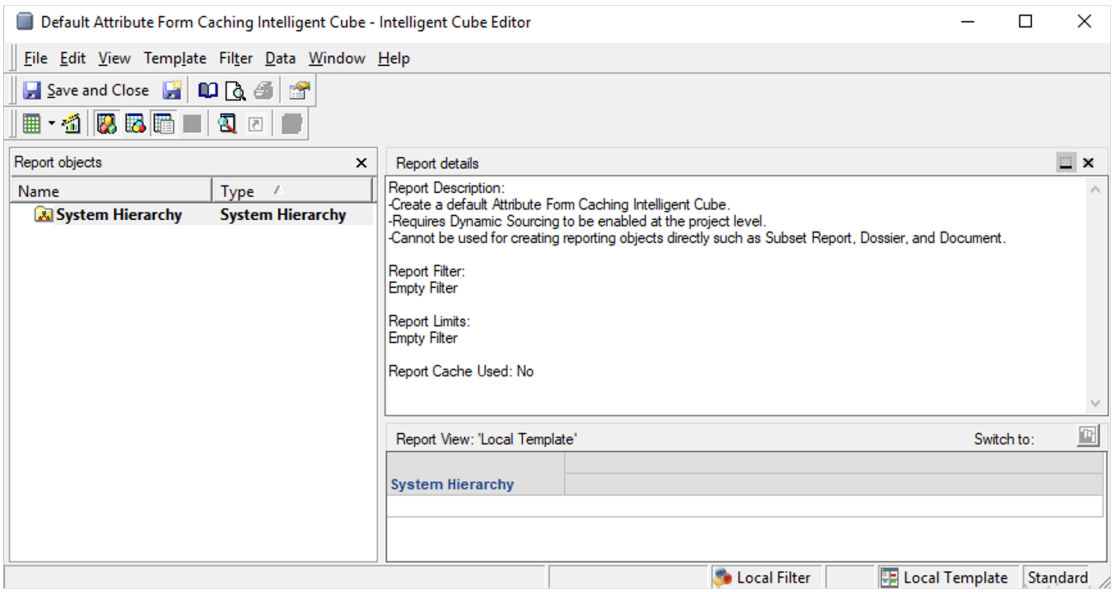
To create an attribute form caching cube

1. In MicroStrategy Developer, select the **Empty Attribute Form Caching Intelligent Cube** template when creating a new Intelligent cube.

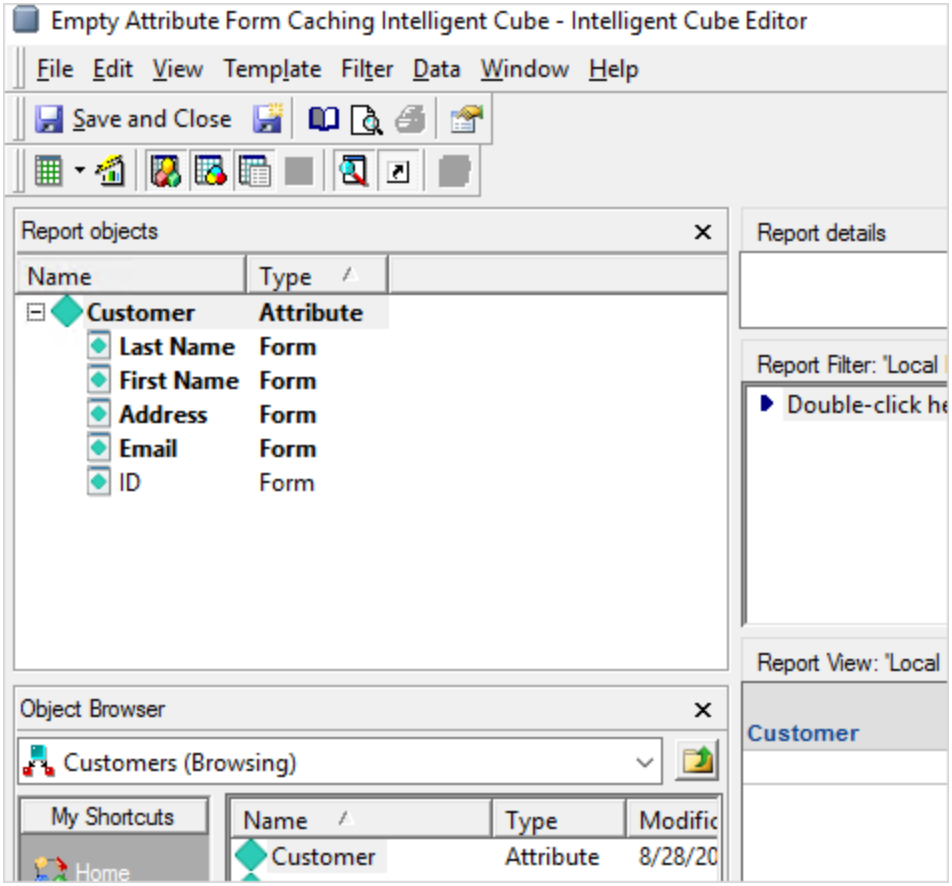


2. By default, there is a system hierarchy on the cube.

Copyright © 2024 All Rights Reserved



3. You can remove the system hierarchy and add your own attributes into the template and include specific attribute forms for caching.



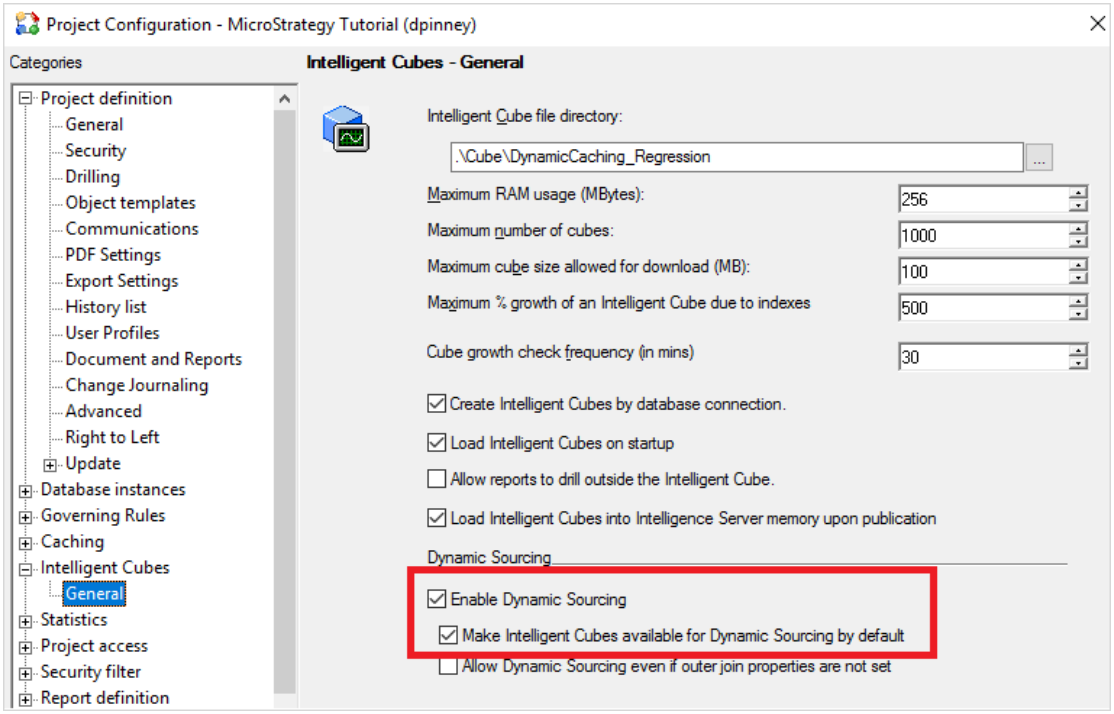
Copyright © 2024 All Rights Reserved

To use an attribute form caching cube

To use an attribute form caching cube, you must enable dynamic sourcing at the project level.

If multiple attribute form caching cubes are available, only the latest published cube is used.

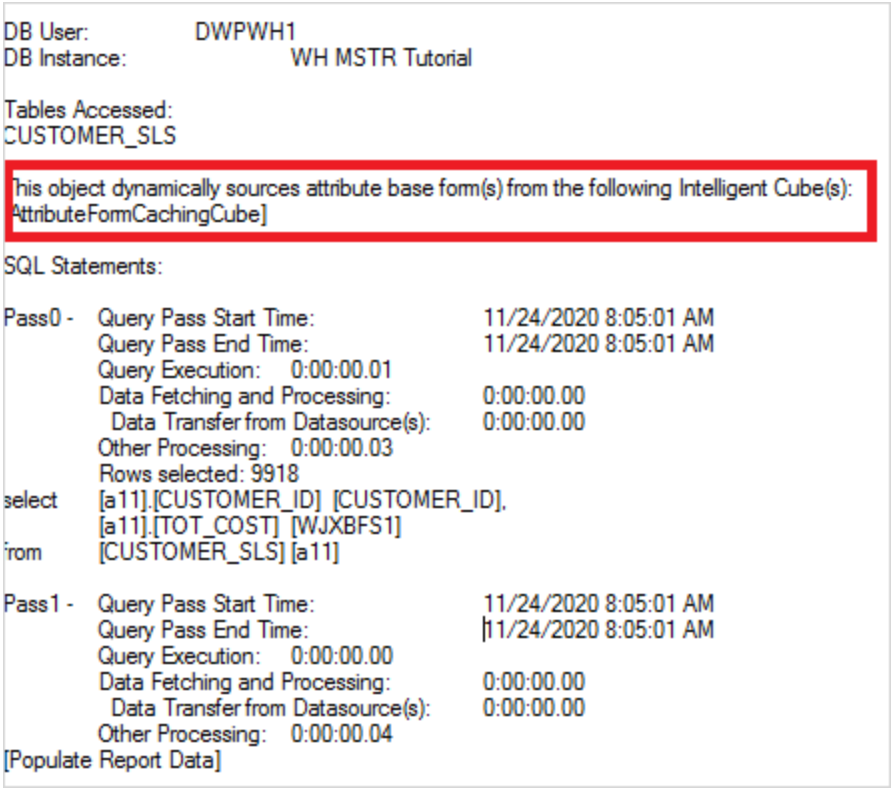
1. In MicroStrategy Developer, go to **Project Configuration** > **Intelligent Cubes** > **Enable Dynamic Sourcing**.
2. Select **Enable Dynamic Sourcing** and **Make Intelligent Cubes available for dynamic sourcing by default**.



3. When executing a normal report or publishing an Intelligent cube that includes the cached attribute forms, the message, "This object dynamically sources attribute base form(s) from the following Intelligent Cube(s):xxx" appears in the SQL view. Only the ID form of the cached

Copyright © 2024 All Rights Reserved

attributes are selected in the SQL statement.



Enabling or disabling dynamic sourcing for reports

By default, dynamic sourcing is enabled for reports. You can enable dynamic sourcing for a report so that active Intelligent Cubes (that are also enabled for dynamic sourcing) are checked to see if the report can retrieve its data from an Intelligent Cube. If an Intelligent Cube fits the data requirements of a report, the report can be run without executing against the data warehouse.

You can enable or disable dynamic sourcing for reports by modifying the **Enable Dynamic Sourcing for Report** VLDB property.

You can enable dynamic sourcing for reports individually or you can enable dynamic sourcing for all reports within a project. While the definition of the VLDB property at the project level defines a default for all reports in the project, any modifications at the report level take precedence over the project level definition. For information on accessing the VLDB Properties

Copyright © 2024 All Rights Reserved

Editor for a project to define a default dynamic sourcing option for all reports, see *Accessing the dynamic sourcing VLDB properties for a project, page 268*.

You can track various information related to dynamic sourcing that can help determine why dynamic sourcing succeeded or failed for reports, as described in *Tracking the use of dynamic sourcing, page 300*.

The procedure below describes how to enable or disable dynamic sourcing for an individual report.

To enable or disable dynamic sourcing for a report

1. In MicroStrategy Developer, browse to a report, and then right-click the report and select **Edit**. The report opens in the Report Editor.
2. Choose **Data** > **VLDB Properties**. The VLDB Properties Editor opens.
3. Choose **Tools** > **Show Advanced Settings** option if it is not already selected.
4. In the **VLDB Settings** list, expand **Dynamic Sourcing**, and then select **Enable Dynamic Sourcing for Report**.
5. Clear the **Use default inherited value** check box.
6. Select either Disable or Enable depending on whether you want to disable or enable dynamic sourcing for the report.
7. Click **Save and Close** twice.

Enabling or disabling dynamic sourcing for Intelligent Cubes

You can enable dynamic sourcing for all Intelligent Cubes in a project, as described in *Enabling or disabling dynamic sourcing for projects, page 266*, or you can enable dynamic sourcing for each Intelligent Cube individually.

By default, dynamic sourcing is disabled for Intelligent Cubes, and they are therefore unavailable for reports checking whether Intelligent Cubes can satisfy their data requirements.

Copyright © 2024 All Rights Reserved

While enabling dynamic sourcing for Intelligent Cubes at the project level defines a default for all Intelligent Cubes in the project, any modifications for individual Intelligent Cubes take precedence over the project level definition.

The steps below show you how to enable or disable dynamic sourcing for an individual Intelligent Cube.

To enable or disable dynamic sourcing for an Intelligent Cube

1. In MicroStrategy Developer, browse to an Intelligent Cube, and then right-click the Intelligent Cube and select **Edit**. The Intelligent Cube opens in the Report Editor.
2. From the **Data** menu, select **Configure Intelligent Cube**. The Intelligent Cube Options dialog box opens.
3. In the **Dynamic Sourcing** area, use the **Make Intelligent Cubes available for dynamic sourcing by default** options to enable or disable dynamic sourcing for the Intelligent Cube:
   - To enable dynamic sourcing for the Intelligent Cube, select **Enabled**.
   - To disable dynamic sourcing for the Intelligent Cube, select **Disabled**.
   - To specify that the Intelligent Cube inherits its dynamic sourcing behavior from the project settings, select **Use Default Project-Level Behavior**.
4. In the **Dynamic Sourcing** area, use the **Allow dynamic sourcing even if outer join properties are not set** options to determine if reports can connect to Intelligent Cubes using dynamic sourcing even when some outer join properties are not defined:

Copyright © 2024 All Rights Reserved

- **Enabled**: Select this option to allow reports to connect to Intelligent Cubes using dynamic sourcing even when some outer join properties are not defined. However, this can cause incorrect data to be returned in scenarios when outer joins would be helpful.

  For example, a report includes the attribute Day and the metric Revenue. It connects to an Intelligent Cube that includes the attribute Day and the metric Revenue, and it also includes the metric Cost. For some days there is data for Revenue, but there is no data for Cost. If the Intelligent Cube does not support any outer joins, then the values for Revenue which do not have corresponding values for Cost aren't returned. In this scenario, the report cannot return complete information from the Intelligent Cube without outer joins.

- **Disabled**: Select this option to prohibit reports from connecting to Intelligent Cubes using dynamic sourcing when some outer join properties are not defined. Selecting this option avoids the possibility of displaying incorrect data in reports.

  You can define your Intelligent Cube to support and use outer joins when necessary. This ensures all data is returned. However, outer joins can cause additional load on your database and require larger Intelligent Cubes. You can enable this support by defining the Metric Join Type, described below.

  - **Metric Join Type**: Any metrics included in the Intelligent Cube that are to be available for dynamic sourcing must be defined to use outer joins in the Intelligent Cube. With the Intelligent Cube open, from the **Data** menu, select **Report Data Options**. In the Report Data Options dialog box, expand **Calculations**, and select **Metric Join Type**. For each metric to make available for dynamic sourcing, change the **Join Type** to **Outer**. Click **OK**.

  For details and examples of VLDB properties, see the System Administration Guide.

Copyright © 2024 All Rights Reserved

- **Use Default Project-Level Behavior**: Select this option to define the Intelligent Cube to inherit its dynamic sourcing behavior from the project settings discussed in *Enabling or disabling dynamic sourcing for projects, page 266*.

5. Click **OK**. The Intelligent Cube Options dialog box closes and you are returned to the Intelligent Cube.
6. Click **Save and Close**.

You can track various information related to dynamic sourcing that can help determine why dynamic sourcing succeeded or failed for reports, as described in *Tracking the use of dynamic sourcing, page 300*.

Disabling dynamic sourcing for attributes

Attributes are available for dynamic sourcing by default, but there are some data modeling conventions that should be considered when using dynamic sourcing.

In general, if attributes use outer joins, accurate data can be returned to reports from Intelligent Cubes through dynamic sourcing. However, if attributes use inner joins, which is more common, the resulting data set may be incomplete. In such cases, you should verify that the attribute data can be correctly represented through dynamic sourcing.

Two scenarios can cause attributes that use inner joins to return incorrect or incomplete data when dynamic sourcing is used:

- Attribute information in lookup and fact tables includes NULL values.
- All attribute elements in fact tables are not also present in lookup tables.

These scenarios are uncommon.

If some attributes fit these scenarios, you can disable dynamic sourcing when these attributes are used in reports and Intelligent Cubes. You can enable and disable dynamic sourcing for attributes by modifying the

Copyright © 2024 All Rights Reserved

VLDB property. This VLDB property has the following options:

- **Attribute columns in fact tables and lookup tables do not contain NULLs and all attribute elements in fact tables are present in lookup tables**: This is the default option for attributes, which enables attributes for dynamic sourcing.
- **Attribute columns in fact tables and lookup tables may contain NULLs and/or some attribute elements in fact tables are not present in lookup tables**: This option disables dynamic sourcing for attributes. This setting should be used if your attribute data is not modeled to support dynamic sourcing. The inclusion of NULLs in your attribute data, or a mismatch between available attribute data in your fact and lookup tables, can cause incorrect data to be returned to reports from Intelligent Cubes through dynamic sourcing.

You can disable dynamic sourcing for attributes individually or you can disable dynamic sourcing for all attributes within a project. While the definition of the VLDB property at the project level defines a default for all attributes in the project, any modifications at the attribute level take precedence over the project level definition. For information on accessing the VLDB Properties Editor for a project to define a default dynamic sourcing option for all attributes, see *Accessing the dynamic sourcing VLDB properties for a project, page 268*.

The steps below show you how to disable or enable dynamic sourcing for an individual attribute. If your database is case-sensitive, you should also review *Supporting filtering on attributes for dynamic sourcing, page 280* to ensure that dynamic sourcing can correctly return data for your attributes.

To enable or disable dynamic sourcing for an attribute

1. In MicroStrategy Developer, browse to an attribute, right-click on it, and select **Edit**. The attribute opens in the Attribute Editor.

Copyright © 2024 All Rights Reserved

2. From the **Tools** menu, select **VLDB Properties**. The VLDB Properties Editor opens.
3. From the **Tools** menu, select the **Show Advanced Settings** option if it is not already selected.
4. In the **VLDB Settings** list, expand **Dynamic Sourcing**, and then select **Attribute Validation**.
5. Clear the **Use default inherited value** check box.
6. Select one of the options depending on whether you want to disable or enable dynamic sourcing for an attribute:
   - To enable attributes to use dynamic sourcing (the default option), select **Attribute columns in fact tables and lookup tables do not contain NULLs and all attribute elements in fact tables are present in lookup tables**.
   - To disable dynamic sourcing for attributes unless outer joins are used for the attribute, select **Attribute columns in fact tables and lookup tables may contain NULLs and/or some attribute elements in fact tables are not present in lookup tables**. This setting should be used if your attribute data is not modeled to support dynamic sourcing.
7. Click **Save and Close** to save your changes to VLDB properties and close the VLDB Properties Editor.
8. Click **Save and Close** to save the attribute and close the Attribute Editor.

You can track various information related to dynamic sourcing that can help determine why dynamic sourcing succeeded or failed for reports, as described in *Tracking the use of dynamic sourcing, page 300*.

Copyright © 2024 All Rights Reserved

Supporting filtering on attributes for dynamic sourcing

To ensure that dynamic sourcing can return the correct results for attributes, you must also verify that filtering on attributes achieves the same results when executed against your database, versus an Intelligent Cube.

A filter on attributes can potentially return different results when executing against the database, compared to using dynamic sourcing to execute against an Intelligent Cube. This can occur if your database is case-sensitive and you create filter qualifications that qualify on the text data of attribute forms.

If your database is case-sensitive, this is enforced for the filter qualification. However, filtering for an Intelligent Cube is handled by the Analytical Engine which does not enforce case sensitivity.

Consider a filter qualification that filters on customers that have a last name beginning with the letter h. If your database is case-sensitive and uses uppercase letters for the first letter in a name, a filter qualification using a lowercase h is likely to return no data. However, this same filter qualification on the same data stored in an Intelligent Cube returns all customers that have a last name beginning with the letter h, uppercase or lowercase.

You can define attributes to either allow filter qualifications to be completed without enforcing case sensitivity, or to disable dynamic sourcing if these types of filters are used on attributes. You can configure this dynamic sourcing behavior for attributes by modifying the String Comparison Behavior VLDB property. This VLDB property has the following options:

- **Use case insensitive string comparison with dynamic sourcing**: This is the default option for attributes. When attempting to use dynamic sourcing, filter qualifications can qualify on the text data of attribute forms without enforcing case sensitivity.

  This is a good option if your database does not enforce case sensitivity. In this scenario, dynamic sourcing returns the same results that would be

Copyright © 2024 All Rights Reserved

returned by the filter qualification if the report was executed against the database.

- **Do not allow any string comparison with dynamic sourcing**: This option disables dynamic sourcing for attributes when a filter qualification is used to qualify on the text data of attribute forms.

  This is a good option if your database is case sensitive. In this scenario, dynamic sourcing could return different results than what would be returned by the filter qualification if the report was executed against the database.

You can modify this VLDB property for attributes individually or you can modify it for all attributes within a project. While the definition of the VLDB property at the project level defines a default for all attributes in the project, any modifications at the attribute level take precedence over the project level definition. For information on accessing the VLDB Properties Editor for a project to define a default dynamic sourcing option for all attributes, see *Accessing the dynamic sourcing VLDB properties for a project, page 268*.

The procedure below describes how to modify the **String Comparison Behavior** VLDB property for an individual attribute.

To modify the String Comparison Behavior VLDB property for an attribute

1. In MicroStrategy Developer, browse to an attribute, then right-click the attribute and select **Edit**. The attribute opens in the Attribute Editor.
2. Choose **Tools** > **VLDB Properties**. The VLDB Properties Editor opens.
3. Choose **Tools** > **Show Advanced Settings** option if it is not already selected.
4. In the **VLDB Settings** list, expand **Dynamic Sourcing**, and then select **String Comparison Behavior**.
5. Clear the **Use default inherited value** check box.

Copyright © 2024 All Rights Reserved

6. Select one of the options depending on whether you want to disable or enable dynamic sourcing for an attribute:
   - **Use case insensitive string comparison with dynamic sourcing**: This is the default option for attributes. When dynamic sourcing is used, it allows filter qualifications to qualify on the text data of attribute forms without enforcing case sensitivity.
   - **Do not allow any string comparison with dynamic sourcing**: This option disables dynamic sourcing for attributes when a filter qualification is used to qualify on the text data of attribute forms.
7. Click **Save and Close** twice

You can track various information related to dynamic sourcing that can help determine why dynamic sourcing succeeded or failed for reports, as described in *Tracking the use of dynamic sourcing, page 300*.

Disabling dynamic sourcing for metrics

Metrics are available for dynamic sourcing by default, but some data modeling conventions should be considered when using dynamic sourcing.

In general, if metrics use outer joins, accurate data can be returned to reports from Intelligent Cubes through dynamic sourcing. However, if metrics use inner joins, which is more common, you should verify that the metric data can be correctly represented through dynamic sourcing.

If the fact table that stores data for metrics includes NULL values for metric data, this can cause metrics that use inner joins to return incomplete data when dynamic sourcing is used. This scenario is uncommon.

If some metrics do fit this scenario, you can disable dynamic sourcing when these metrics are used in reports and Intelligent Cubes. You can enable and disable dynamic sourcing for metrics by modifying the Metric Validation VLDB property. This VLDB property has the following options:

Copyright © 2024 All Rights Reserved

- **Enable dynamic sourcing for metric**: This is the default option for metrics, which enables metrics for dynamic sourcing.
- **Disable dynamic sourcing for metric**: This option disables dynamic sourcing for metrics. This setting should be used if your metric data is not modeled to support dynamic sourcing. For example, the inclusion of NULLs in fact tables that contain your metric data can cause incorrect data to be returned to reports from Intelligent Cubes through dynamic sourcing.

You can disable dynamic sourcing for metrics individually or you can disable dynamic sourcing for all metrics within a project. While the definition of the VLDB property at the project level defines a default for all metrics in the project, any modifications at the metric level take precedence over the project level definition. For information on accessing the VLDB Properties Editor for a project to define a default dynamic sourcing option for all metrics, see *Accessing the dynamic sourcing VLDB properties for a project, page 268*.

The procedure below describes how to disable or enable dynamic sourcing for an individual metric.

To enable or disable dynamic sourcing for a metric

1. In MicroStrategy Developer, browse to a metric, then right-click the metric and select **Edit**. The metric opens in the Metric Editor.
2. From the **Tools** menu, point to **Advanced Settings**, and then select **VLDB Properties**. The VLDB Properties Editor opens.
3. From the **Tools** menu, select the **Show Advanced Settings** option if it is not already selected.
4. In the **VLDB Settings** list, expand **Dynamic Sourcing**, and then select **Metric Validation**.
5. Clear the **Use default inherited value** check box.

Copyright © 2024 All Rights Reserved

6. Select one of the options depending on whether you want to disable or enable dynamic sourcing for a metric:
   - **Enable dynamic sourcing for metric**: This is the default option, which enables dynamic sourcing for the metric.
   - **Disable dynamic sourcing for metric**: This option disables dynamic sourcing for the metric. This setting should be used if your metric data is not modeled to support dynamic sourcing.
7. Click **Save and Close** to save your changes to VLDB properties and close the VLDB Properties Editor.
8. Click **Save and Close** to save the attribute and close the Attribute Editor.

You can track various information related to dynamic sourcing that can help determine why dynamic sourcing succeeded or failed for reports, as described in *Tracking the use of dynamic sourcing, page 300*.

Disabling dynamic sourcing for aggregate tables

Reports that use aggregate tables are available for dynamic sourcing by default, but there are some data modeling conventions that should be considered when using dynamic sourcing.

In general, aggregate tables allow accurate data to be returned to reports from Intelligent Cubes through dynamic sourcing. However, if the aggregate tables use an aggregation other than Sum, or there is different data between aggregate tables and other tables in the data warehouse, this can cause aggregate tables to return incorrect data when dynamic sourcing is used. An example of an aggregate table not containing the same data is if an aggregate table includes data for years 2006, 2007, and 2008 but the lookup table for Year only includes data for 2007 and 2008.

These scenarios are uncommon.

If some aggregate tables do fit these scenarios, you can disable dynamic sourcing when these aggregate tables are used in reports and Intelligent

Copyright © 2024 All Rights Reserved

Cubes. You can enable and disable dynamic sourcing for aggregate tables by modifying the Aggregate Table Validation VLDB property. This VLDB property has the following options:

- **Aggregate tables contain the same data as corresponding detail tables and the aggregation function is SUM**: This is the default option for aggregate tables, which enables aggregate tables for dynamic sourcing.
- **Aggregate tables contain either less data or more data than their corresponding detail tables and/or the aggregation function is not SUM**: This option disables dynamic sourcing for aggregate tables. This setting should be used if your aggregate tables are not modeled to support dynamic sourcing. The use of an aggregation function other than Sum or the mismatch of data in your aggregate tables with the rest of your data warehouse can cause incorrect data to be returned to reports from Intelligent Cubes through dynamic sourcing.

You can disable dynamic sourcing individually for reports that use aggregate tables or you can disable dynamic sourcing for all reports that use aggregate tables within a project. While the definition of the VLDB property at the project level defines a default for all reports in the project, any modifications at the report level take precedence over the project level definition. For information on accessing the VLDB Properties Editor for a project to define a default dynamic sourcing option for all metrics, see *Accessing the dynamic sourcing VLDB properties for a project, page 268*.

The procedure below describes how to disable or enable dynamic sourcing for an individual report that uses an aggregate table.

Prerequisite

- A report has been created in a project.

Copyright © 2024 All Rights Reserved

To enable or disable dynamic sourcing for a report that uses an aggregate table

1. In MicroStrategy Developer, browse to a report, then right-click the report and select **Edit**. The report opens in the Report Editor.
2. From the **Data** menu, select **VLDB Properties**. The VLDB Properties Editor opens.
3. From the **Tools** menu, select the **Show Advanced Settings** option if it is not already selected.
4. In the **VLDB Settings** list, expand **Dynamic Sourcing**, and then select **Aggregate Table Validation**.
5. Clear the **Use default inherited value** check box.
6. Select one of the options depending on whether you want to disable or enable dynamic sourcing for a report that uses aggregate tables:
   - **Aggregate tables contain the same data as corresponding detail tables and the aggregation function is SUM**: This is the default option for aggregate tables, which enables aggregate tables for dynamic sourcing.
   - **Aggregate tables contain either less data or more data than their corresponding detail tables and/or the aggregation function is not SUM**: This option disables dynamic sourcing for aggregate tables. This setting should be used if your aggregate tables are not modeled to support dynamic sourcing.
7. Click **Save and Close** to save your changes to VLDB properties and close the VLDB Properties Editor.
8. Click **Save and Close** to save the report and close the Report Editor.

Copyright © 2024 All Rights Reserved

Using Cube Advisor to support dynamic sourcing

MicroStrategy Cube Advisor allows you to create and support a dynamic sourcing strategy that can best support the reports in your projects. Cube Advisor provides the following features for creating and supporting a dynamic sourcing strategy:

- Analyzes reports to determine if their report definitions can support the use of dynamic sourcing.
- Recommends and creates Intelligent Cubes that could provide data to reports using dynamic sourcing. This includes the following:
  - Recommends Intelligent Cube definitions for Intelligent Cubes that could provide data to reports via dynamic sourcing. This includes providing the number of reports that could connect to the Intelligent Cube using dynamic sourcing, as well as the names and locations of these reports.

    You can also supply a MicroStrategy Enterprise Manager report to include more information on the performance benefits of each Intelligent Cube as part of your dynamic sourcing strategy. For information on locating and using this Enterprise Manager report, see *Enterprise Manager report for Cube Advisor, page 290*.
- Creates the Intelligent Cubes you select to create to support your dynamic sourcing strategy. These Intelligent Cubes are created in a folder along with shortcuts to all the reports that can connect to the Intelligent Cube using dynamic sourcing. For information on creating Intelligent Cubes using Cube Advisor, see *Reviewing and creating recommended Intelligent Cubes, page 292*.

Prerequisites for using Cube Advisor

You must complete a few prerequisites before using Cube Advisor to recommend and create Intelligent Cubes to support a dynamic sourcing strategy. These prerequisites include:

Copyright © 2024 All Rights Reserved

- Enable dynamic sourcing for the project and other aspects of the MicroStrategy system.
- Create the metric levels log file used to track the use of dynamic sourcing.
- Locate and execute the MicroStrategy Enterprise Manager report to provide additional performance benefit information on each Intelligent Cube recommended by Cube Advisor. This is an optional prerequisite.

Creating the metric levels log

The metric levels log file is one of the files that can be used to track the use of dynamic sourcing (see *Tracking the use of dynamic sourcing, page 300*).

Cube Advisor uses this log file to help recommend and create Intelligent Cubes to support dynamic sourcing. The steps below show you how to create this log file using the MicroStrategy Diagnostics and Performance Logging tool.

If you install and use Cube Advisor on the same machine that hosts Intelligence Server, the metric levels log is created automatically. In this scenario, you do not have to perform the steps provided below.

To create the metric levels log file

1. In Windows, choose **Start** > **Programs** > **MicroStrategy Tools** > **Diagnostics Configuration**. The MicroStrategy Diagnostics and Performance Logging tool opens.
2. In the **Select Configuration** drop-down list, select from one of two configurations to see the current default setup, as follows:

- **Machine Default**: The components and counters that are displayed reflect the client machine.
- **CastorServer instance**: The components and counters that are displayed reflect server-specific features.

Copyright © 2024 All Rights Reserved

When you select the CastorServer instance, you can select whether to use the default configuration.

- On the **Diagnostics Configuration** tab, this check box is named **Use Default Diagnostics Configuration**
- On the **Performance Configuration** tab, this check box is called **Use Default Performance Configuration**.

This check box refers to the Machine Default settings. No matter what you have changed and saved on either tab when CastorServer instance is selected, if you check the **Use Default Configuration** box, the system logs whatever information is configured for Machine Default at run time.

1. On the **Diagnostics** tab, locate the **Dynamic Sourcing** component.
2. For the **Metric Levels Log** dispatcher, select the check box in the **Console Log** column.
3. For the **Metric Levels Log** dispatcher, in the **File Log** column, click the drop-down list and select **<New>**. The Log Destination Editor opens.
4. In the **File name** field, type a name for the metric levels log. In later examples, the log file is named `MetricLevelsLog`.
5. You can leave the default settings for the other options, and then click **Save**. Click **Close**.
6. For the **Metric Levels Log** dispatcher, in the **File Log** column, click the drop-down list and select the metrics level log file that you created in the previous steps.
7. Select **Save**. The metric levels log is created in the Log directory within the MicroStrategy common files. The default directory is `C:\Program Files\Common Files\MicroStrategy\ Log`.

Copyright © 2024 All Rights Reserved

Enterprise Manager report for Cube Advisor

You can also use a MicroStrategy Enterprise Manager report to include more information on the performance benefits of each Intelligent Cube as part of your dynamic sourcing strategy. When using this Enterprise Manager report, the information that can be returned for each potential Intelligent Cube includes:

- **Jobs**: The number of jobs that would not have to be executed against the data warehouse if the Intelligent Cube is created to support your dynamic sourcing strategy.
- **Result Rows**: The number of rows of data for all reports that could be returned from the Intelligent Cube rather than the data warehouse if the Intelligent Cube is created to support your dynamic sourcing strategy.
- **Users**: The number of users that execute the reports that could connect to the Intelligent Cube using dynamic sourcing if the Intelligent Cube is created to support your dynamic sourcing strategy.
- **Database Time**: The amount of time to execute SQL statements and retrieve the results from the data warehouse that is saved if the Intelligent Cube is created to support your dynamic sourcing strategy.

This report is available with MicroStrategy Enterprise Manager. Search the Enterprise Manager project for the report named **Report Request Statistics per Project**.

To use this report with Cube Advisor, you must execute the report and export it as a Microsoft Excel file or a text file.

The steps below show you how to execute and export this report in MicroStrategy Developer.

Copyright © 2024 All Rights Reserved

To execute and export the Enterprise Manager report in Developer

1. In Developer, log in to the Enterprise Manager project.
2. Locate the Enterprise Manager report for Cube Advisor, by searching the Enterprise Manager project for the report named **Report Request Statistics per Project**.
3. Right-click the **Report Request Statistics per Project** report and select **Run**. The Report Editor opens and the report is executed.
4. Choose **Data** > **Export Options**. The Excel Options dialog box opens.
5. In the **Application** drop-down list, select **MS Excel Worksheet**.
6. On the **Appearance** tab, select the following export options:
   - In the **Exported fields** area, select the **Report Data** check box. All other check boxes must be cleared.
   - In the **Formatting** area, select the **Apply formatting** and **Use 6.x layout** check boxes.
7. Click **OK**.
8. Choose **Data** > **Export To** > **MS Excel**. Once the export is complete, the exported report is displayed in Microsoft Excel.
9. Save the Excel file as a tab delimited text file (`.txt`).
10. If you use an internationalized MicroStrategy environment that uses various character sets for your data, you must open the tab delimited text file in a text editor and save the file with a **UTF-8** encoding.

To execute and export the Enterprise Manager report in MicroStrategy Web

The steps below show you how to execute and export this report in MicroStrategy Web.

Copyright © 2024 All Rights Reserved

1. In MicroStrategy Web, log in to the Enterprise Manager project.
2. Locate the Enterprise Manager report for Cube Advisor. Search the Enterprise Manager project for the report named **Report Request Statistics per Project**.
3. Click the **Report Request Statistics per Project** report to execute it.
4. Choose **Home** > **Export** > **Plain text**. The Export Options page opens.
5. In the **Delimiter** drop-down list, select **Tab**.
6. Clear all available check boxes.
7. In the **Remove extra column** drop-down list, select **Yes**.
8. Click **Export**. The File Download dialog box opens.
9. Click **Save**.
10. Type a name and select a location for the text file, and click **Save**.
11. If you use an internationalized MicroStrategy environment that uses various character sets for your data, you must open the tab delimited text file in a text editor and save the file with a **UTF-8** encoding.

Reviewing and creating recommended Intelligent Cubes

After you have completed the prerequisites for using Cube Advisor, you can begin using Cube Advisor to create Intelligent Cubes.

The steps below show you how to use Cube Advisor to create Intelligent Cubes to support dynamic sourcing for the reports in your project.

Prerequisites

- You must complete the prerequisites described in *Prerequisites for using Cube Advisor, page 287* before using Cube Advisor to recommend and create Intelligent Cubes to support a dynamic sourcing strategy.
- In order to use Cube Advisor, you must log in to a project using a MicroStrategy user account with the following privileges:

Copyright © 2024 All Rights Reserved

- Privileges required to execute reports and documents.
- Privileges required to create folders in a MicroStrategy project.
- Privileges required to create Intelligent Cubes.

You can use Integrity Manager to perform a before and after test to identify which reports actually connect to a suggested Intelligent Cube and also to test data integrity.

To review and create recommended Intelligent Cubes using Cube Advisor

1. In Windows choose **Start** > **Programs** > **MicroStrategy Tools** > **Cube Advisor**. Cube Advisor opens.
2. You can analyze a new set of reports using Cube Advisor or use the results of a previous Cube Advisor analysis:
   - To use Cube Advisor to analyze a new set of reports, choose **Tools** > **Options**. The Options dialog box opens. Continue to step 3 in this procedure and complete all of the steps provided.
   - If you have previously used Cube Advisor to create Intelligent Cubes to support dynamic sourcing for the reports in your project, you can use the results file of this analysis. Choose **File** > **Open**. In the Open dialog box, select the Cube Advisor results file to use and click **Open**.

     The results of a Cube Advisor analysis are stored in the MicroStrategy common files folder (the default is `C:\Program Files\Common Files\MicroStrategy`). The name of the file is in the format *`ProjectName`*`.details.txt`. For example, analyzing the MicroStrategy Tutorial project creates a Cube Advisor results file named `MicroStrategy Tutorial.details.txt`.

Copyright © 2024 All Rights Reserved

3. Click **...** (browse) to supply the following files to support Cube Advisor and define the following options:

   - **Metric Level File**: The metric levels log file is one of the files that can be used to track the use of dynamic sourcing. Cube Advisor uses this log file to help recommend and create Intelligent Cubes to support dynamic sourcing.

     If you install and use Cube Advisor on the same machine that hosts Intelligence Server, the metric levels log file is automatically created by Cube Advisor. In this scenario you do not need to create the metric levels log file or provide the location for the file. Otherwise, you must create this file manually and provide the directory it is stored in.

     When manually creating the metric levels log file, the file is created in the `Log` directory within the MicroStrategy common files. The default directory is `C:\Program Files\Common Files\MicroStrategy\Log`. For steps on how to create this file, see *Creating the metric levels log, page 288*.

   - **Enterprise Manager File**: You can also supply a MicroStrategy Enterprise Manager report to include more information on the performance benefits of each Intelligent Cube as part of your dynamic sourcing strategy. For information on locating, executing, and exporting this report as well as the information provided by supplying the report, see *Enterprise Manager report for Cube Advisor, page 290*.

   - **Ignore Reports That Are Covered By Existing Cubes**: Select this option to exclude reports that already connect to an Intelligent Cube from the Cube Advisor analysis. This prevents Cube Advisor from analyzing and recommending Intelligent Cubes for reports that already connect to Intelligent Cubes.

Copyright © 2024 All Rights Reserved

- **Ignore Reports Not In Enterprise Manager File**: Select this option to exclude reports that are not included in the Enterprise Manager report from Cube Advisor analysis. This allows Cube Advisor to focus only on the reports that were included in the Enterprise Manager report for further performance analysis.

4. Click **OK** to close the Options dialog box and return to Cube Advisor.
5. Provide the following connection information:
   - **Computer Name**: Type the name of the machine on which Intelligence Server is hosted.
   - **Port**: Type the port number used for Intelligence Server. The default is 34952.
   - **User**: Type the MicroStrategy user's user name to connect to a MicroStrategy project. The MicroStrategy user must have the privileges listed in the prerequisites for these steps.
   - **Password**: Type the password for the MicroStrategy user.
6. Click **Connect**. MicroStrategy projects are displayed.
7. Select a project, and then click **Connect To Project**. The Report Selection page opens.

   ⚠ If you supplied the Enterprise Manager report, you must select the project that was analyzed using the Enterprise Manager report.

8. Expand the folders of the project to locate the reports to analyze with Cube Advisor.

   Selecting a check box for a report includes the report in the analysis. Selecting a check box for a folder includes all reports in that folder and all reports in folders within the folder in the analysis.

Copyright © 2024 All Rights Reserved

9. Once you have selected all the reports to analyze with Cube Advisor, click **Get Cube Recommendations**. The Analyzing Reports page opens and report analysis begins.

   To analyze reports, only the report SQL is analyzed, which allows a large number of reports to be analyzed without having to execute the report SQL against the database.

   If the analysis completes with a message that no reports can use dynamic sourcing, this can be caused by various scenarios:

   - The features used in the reports prevent the reports from being able to use dynamic sourcing. To review a list of features that prevent the use of dynamic sourcing, see *Features that prevent the use of dynamic sourcing, page 261*.

   - The metric levels log file was not created properly. Review the steps to create a metrics level log file (see *Creating the metric levels log, page 288*) and attempt the Cube Advisor analysis again.
   - Dynamic sourcing is not enabled for the project, reports, or other objects and features in the project. For information on configuring dynamic sourcing, see *Configuring dynamic sourcing, page 266*.

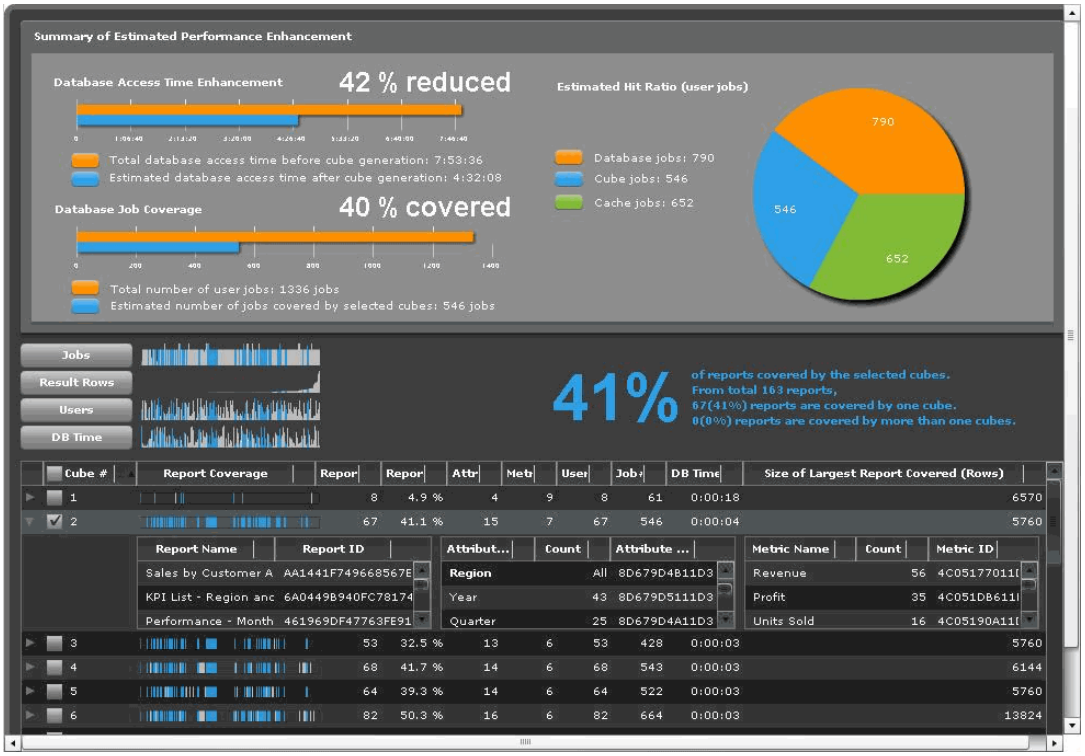
10. Once analysis is complete, the Cube Recommendations page opens. This page allows you to review the recommended Intelligent Cubes, and select which Intelligent Cubes to create. Depending on whether you supplied an Enterprise Manager report, you can review Intelligent Cubes as described below:

    - If you did not supply an Enterprise Manager report, the information on the recommended Intelligent Cubes that is displayed includes the number of reports that could connect to the Intelligent Cube using dynamic sourcing.

Copyright © 2024 All Rights Reserved

You can select a recommended Intelligent Cube to display the attributes and metrics that would be included in the Intelligent Cube, as well as a list of reports that could connect to the Intelligent Cube using dynamic sourcing.

- If you supplied an Enterprise Manager report, a Flash visualization provides detailed information on the recommended Intelligent Cubes, an example of which is shown below:



Some of the benefits of the Flash visualization include:

- You can click **Jobs**, **Result Rows**, **Users**, and **DB Time** to sort the Intelligent Cubes by these various usage and performance statistics. These usage and performance statistics are displayed as microcharts, which are aligned with the report coverage microcharts provided for each Intelligent Cube. For information on these usage and performance statistics, see *Enterprise Manager report for Cube*

Copyright © 2024 All Rights Reserved

Advisor, page 290.

- You can select an Intelligent Cube to see how creating it affects various performance statistics.
- The area at the top of the Flash visualization provides a summary of the estimated performance improvements that would come from building the selected set of Intelligent Cubes.
- You can expand an Intelligent Cube to review more information on the attributes and metrics that would be included in the Intelligent Cube, as well as a list of reports that could connect to the Intelligent Cube using dynamic sourcing.

  This also includes information on how many reports that could connect to the Intelligent Cube require certain attributes and metrics. This analysis allows you to determine the benefit of including each attribute and metric in an Intelligent Cube.
- Each Intelligent Cube includes a microchart, which represents the subset of reports that could connect to the Intelligent Cube using dynamic sourcing. A bar in the microchart represents that the report is covered by the Intelligent Cube and no bar means that report is not covered by the Intelligent Cube. By moving the cursor over the microchart, a tooltip is displayed that lists the various reports that could connect to the Intelligent Cube using dynamic sourcing.
- Once you select a specific Intelligent Cube, the bars for all reports covered by that Intelligent Cube become blue. This color coding also helps to visually analyze the overlap of reports across Intelligent Cubes. If you see two Intelligent Cubes have a very similar report coverage distribution, then only one of the two Intelligent Cubes should be created.

Copyright © 2024 All Rights Reserved

11. Once you have reviewed the information, select the check boxes for the Intelligent Cubes to create using Cube Advisor, and then click **Create Cube Design**. The Browse for Folder dialog box opens.
12. Select a folder to create the Intelligent Cubes in. Each Intelligent Cube is created in a separate folder that is created within the folder you select. Click **OK**. The Cube Design Options dialog box opens.
13. Select one of the following options:
    - **Design Intelligent Cubes with outer join properties**: Select this option to enable outer joins on all Intelligent Cubes to be created. This ensures that all warehouse data is captured; however, this can potentially increase the size of the Intelligent Cubes.
    - **Allow Dynamic Sourcing without enabling outer join properties**: Select this option to allow reports to connect to Intelligent Cubes using dynamic sourcing even when some outer join properties are not defined. However, this can cause incorrect data to be returned in certain scenarios.

      For additional information on enabling dynamic sourcing for Intelligent Cubes, see *Enabling or disabling dynamic sourcing for Intelligent Cubes, page 274*.

      Click **OK** to begin creating the Intelligent Cubes. The Creating Cubes dialog box opens.
14. Once the Intelligent Cubes are created, click **OK**.

Each Intelligent Cube is created in a separate folder. Each folder also contains shortcuts to all the reports that can connect to the Intelligent Cube using dynamic sourcing. This provides easy recognition of the reports that can connect to each Intelligent Cube using dynamic sourcing. You must publish the Intelligent Cubes to allow the reports to connect to the new Intelligent Cubes using dynamic sourcing. For information on how to publish Intelligent Cubes, see *Publishing Intelligent Cubes, page 43*.

Copyright © 2024 All Rights Reserved

The results of the Cube Advisor analysis are stored in the MicroStrategy common files folder (the default is `C:\Program Files\Common Files\MicroStrategy`). The name of the file is in the format *`ProjectName`*`.details.txt`. For example, analyzing the MicroStrategy Tutorial project creates a Cube Advisor results file named `MicroStrategy Tutorial.details.txt`. You can use this file with Cube Advisor to review and create Intelligent Cubes at another time.

Tracking the use of dynamic sourcing

If you enable dynamic sourcing for a project, you can track various information related to dynamic sourcing that can determine why dynamic sourcing succeeds or fails for reports. By determining why dynamic sourcing fails for a given report, you can modify your reports, Intelligent Cubes, and other objects to allow reports to utilize dynamic sourcing.

You can track the use of dynamic sourcing with the MicroStrategy Diagnostics and Performance Logging tool. For information on how to use the Diagnostics and Performance Logging tool, see the System Administration Help.

This tool allows you to enable or disable various logs related to dynamic sourcing, which record various information about dynamic sourcing, including why dynamic sourcing cannot be used in different scenarios. The sections listed below describe the dynamic sourcing log files and error codes that can be tracked using the Diagnostics and Performance Logging tool:

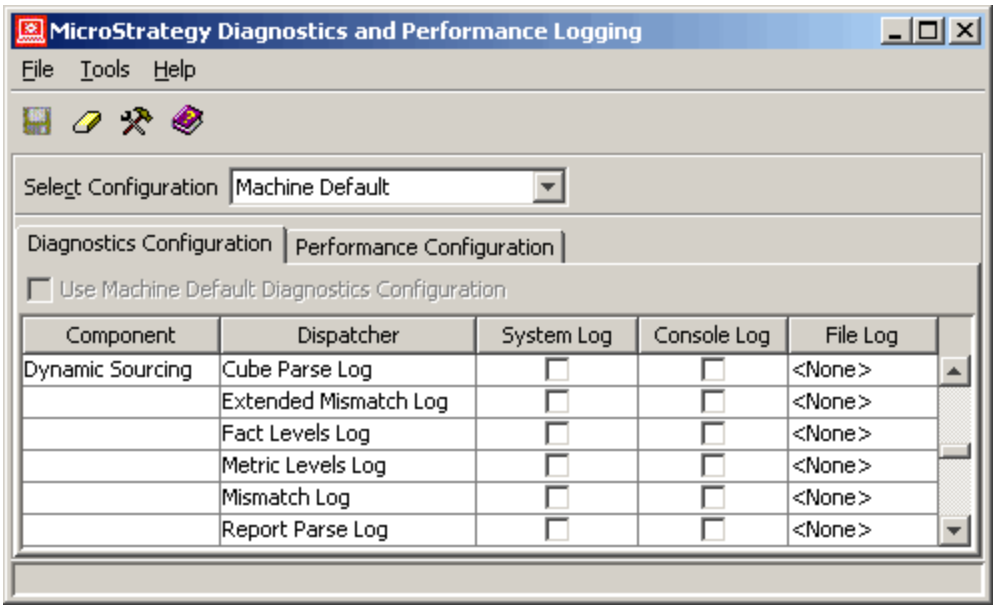
- *Dynamic sourcing log files, page 300*
- *Dynamic sourcing error codes and explanations, page 308*

Dynamic sourcing log files

In the Diagnostics and Performance Logging tool, all of the dynamic sourcing log files are available on the Diagnostics Configuration tab as part

Copyright © 2024 All Rights Reserved

of the Dynamic Sourcing component, as shown in the image below. For information on accessing the Diagnostics and Performance Logging tool, see the System Administration Help.



With the Diagnostics and Performance Logging tool, you can enable the following dynamic sourcing log files to track information about the use of dynamic sourcing.

Intelligent Cube parse log

Information about an Intelligent Cube is included in the Intelligent Cube parse log when the Intelligent Cube is published and made available by Intelligence Server.

This log displays whether an Intelligent Cube has been defined to be available for dynamic sourcing. It also lists any metrics included in the Intelligent Cube that are disabled for dynamic sourcing, and other issues that may prevent the Intelligent Cube from being available for dynamic sourcing. For explanations of various error codes that can be included in this log, see *Dynamic sourcing error codes and explanations, page 308*.

If the entries in this log for an Intelligent Cube end with `CMI_NO_ERROR`, this means the Intelligent Cube is available to reports for dynamic sourcing. Be

Copyright © 2024 All Rights Reserved

aware that some of the metrics in an Intelligent Cube still may not be available for dynamic sourcing. You can review this log to determine whether any metrics for the Intelligent Cube are not available for dynamic sourcing.

You can also display this log in the SQL View of Intelligent Cubes. This can help determine which reports use dynamic sourcing to connect to the Intelligent Cube, as well as why some reports cannot use dynamic sourcing to connect to the Intelligent Cube. The steps below show you how to include this log for Intelligent Cubes in a project.

To display the Intelligent Cube parse log in Intelligent Cubes

1. In Developer, log in to a project.
2. Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand the **Project definition** category, and select the **Advanced** category.
4. Within the **Analytical Engine VLDB properties** area, click **Configure**. The VLDB Properties Editor opens.

   The VLDB property defined using the steps below can also be defined for individual Intelligent Cubes. To access the VLDB Properties Editor for an Intelligent Cube, with the Intelligent Cube open, from the **Data** menu, select **VLDB Properties**.

5. From the **Tools** menu, ensure that the **Show Advanced Settings** option is selected.
6. In the **VLDB Settings** list, expand **Dynamic Sourcing**, and then select **Enable Cube Parse Log in SQL View**.
7. Clear the **Use default inherited value** check box.
8. Select **Enable Cube Parse Log in SQL View**.

Copyright © 2024 All Rights Reserved

9. Click **Save and Close** to save your changes and close the VLDB Properties Editor.
10. Click **OK**.

Report parse log

When a report is executed, information about the report is included in the report parse log.

Reports connected to a specific Intelligent Cube do not send information to the report parse log.

This log lists any reasons why a report cannot use dynamic sourcing. For explanations of various error codes that can be included in this log, see *Dynamic sourcing error codes and explanations, page 308*.

If the entries in this log for a report end with `CMI_NO_ERROR`, this means the report can use dynamic sourcing. However, this does not necessarily mean that dynamic sourcing is used; an Intelligent Cube that meets the data requirements for the report must be available.

The report parse log can increase in size quickly and thus is best suited for troubleshooting purposes.

You can also display this log in the SQL View of reports. This log helps determine whether a report can use dynamic sourcing to connect to an Intelligent Cube. The steps below show you how to include this log for reports in a project.

To display the report parse log in reports

1. In Developer, log in to a project.
2. Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.

Copyright © 2024 All Rights Reserved

3. Expand the **Project definition** category, and select the **Advanced** category.
4. Within the **Analytical Engine VLDB properties** area, click **Configure**. The VLDB Properties Editor opens.

   The VLDB property defined using the steps below can also be defined for individual reports. To access the VLDB Properties Editor for a report, with the report open, from the **Data** menu, select **VLDB Properties**.

5. From the **Tools** menu, ensure that the **Show Advanced Settings** option is selected.
6. In the **VLDB Settings** list, expand **Dynamic Sourcing**, and then select **Enable Report Parse Log in SQL View**.
7. Clear the **Use default inherited value** check box.
8. Select **Enable Report Parse Log in SQL View**.
9. Click **Save and Close** to save your changes and close the VLDB Properties Editor.
10. Click **OK**.

Mismatch log

When a report that can use dynamic sourcing is executed, which can be verified with the report parse log, and Intelligent Cubes are available for dynamic sourcing, information about whether a matching Intelligent Cube can be found for the report is included in the mismatch log.

The mismatch log can increase in size quickly and thus is best suited for troubleshooting purposes.

This log lists any reasons why a report cannot use a specific Intelligent Cube to satisfy its data requirements. For explanations of various error codes that

Copyright © 2024 All Rights Reserved

can be included in this log, see *Dynamic sourcing error codes and explanations, page 308*.

If the entries in this log for a report and Intelligent Cube combination end with `CMI_NO_ERROR`, this means the report can use dynamic sourcing to access the Intelligent Cube.

If dynamic sourcing cannot be used because of a metric (`CMI_NO_GOOD_METRIC_FOUND`), information on why the metric prevents the use of dynamic sourcing is provided in the extended mismatch log.

You can also display this log in the SQL View of reports. This log helps determine why a report that can use dynamic sourcing cannot connect to a specific Intelligent Cube. The steps below show you how to include this log for reports in a project.

To display the mismatch log in reports

1. In Developer, log in to a project.
2. Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand the **Project definition** category, and select the **Advanced** category.
4. Within the **Analytical Engine VLDB properties** area, click **Configure**. The VLDB Properties Editor opens.

   The VLDB property defined using the steps below can also be defined for individual reports. To access the VLDB Properties Editor for a report, with the report open, from the **Data** menu, select **VLDB Properties**.

5. From the **Tools** menu, ensure that the **Show Advanced Settings** option is selected.

Copyright © 2024 All Rights Reserved

6. In the **VLDB Settings** list, expand **Dynamic Sourcing**, and then select **Enable Mismatch Log in SQL View**.
7. Clear the **Use default inherited value** check box.
8. Select **Enable Mismatch Log in SQL View**.
9. Click **Save and Close** to save your changes and close the VLDB Properties Editor.
10. Click **OK**.

Extended mismatch log

When a report that can use dynamic sourcing is executed, which can be verified with the report parse log, and Intelligent Cubes are available for dynamic sourcing, information about whether a matching Intelligent Cube can be found for the report is included in the mismatch log.

However, if dynamic sourcing cannot be used because of a metric (`CMI_NO_GOOD_METRIC_FOUND`), information on why the metric prevents the use of dynamic sourcing is provided in the extended mismatch log. This information is listed for every metric that prevents the use of dynamic sourcing.

The extended mismatch log can increase in size quickly and thus is best suited for troubleshooting purposes.

You can also display this log in the SQL View of reports. This log helps determine why a metric prevents the use of dynamic sourcing is provided in the extended mismatch log. The steps below show you how to include this log for reports in a project.

Copyright © 2024 All Rights Reserved

To display the extended mismatch log in reports

1. In Developer, log in to a project.
2. Right-click the project and select **Project Configuration**. The Project Configuration Editor opens.
3. Expand the **Project definition** category, and select the **Advanced** category.
4. Within the **Analytical Engine VLDB properties** area, click **Configure**. The VLDB Properties Editor opens.

   The VLDB property defined using the steps below can also be defined for individual reports. To access the VLDB Properties Editor for a report, with the report open, from the **Data** menu, select **VLDB Properties**.

5. From the **Tools** menu, ensure that the **Show Advanced Settings** option is selected.
6. In the **VLDB Settings** list, expand **Dynamic Sourcing**, and then select **Enable Extended Mismatch Log in SQL View**.
7. Clear the **Use default inherited value** check box.
8. Select **Enable Extended Mismatch Log in SQL View**.
9. Click **Save and Close** to save your changes and close the VLDB Properties Editor.
10. Click **OK**.

Metric levels log

When a report that can use dynamic sourcing is executed, which can be verified with the report parse log, information about the report is included in this log, including:

Copyright © 2024 All Rights Reserved

- The lowest level attributes on a report, which defines the report level
- The formulas of metrics on the report

This information can help determine what attributes and metrics need to be included in an Intelligent Cube for the report to be able to connect to it through dynamic sourcing.

This log file can also be used with MicroStrategy Cube Advisor to recommend and create Intelligent Cubes that could be connected to by reports using dynamic sourcing. For information on how Cube Advisor allows you to create and support a dynamic sourcing strategy that can best support the reports in your projects, see *Using Cube Advisor to support dynamic sourcing, page 287*.

Fact levels log

When a report that can use dynamic sourcing is executed, which can be verified with the report parse log included in this log. This log tracks the same information as the metrics levels log. The only difference is that the fact levels log lists the facts used for metrics rather than the metric formulas.

Dynamic sourcing error codes and explanations

There are various reasons for why a report cannot use dynamic sourcing. To help determine why reports in your projects are not able to use dynamic sourcing, the table below provides explanations of all of the error codes that can be triggered for dynamic sourcing scenarios. These error codes are included in the dynamic sourcing log files, which are described in *Dynamic sourcing log files, page 300*.

Error Code	Explanation
`CMI_AGG_METRIC_BRANCH_WITH_ZERO_CHECKING`	Dynamic sourcing cannot be used because a metric included in the Intelligent Cube uses zero checking, which is enabled with the Zero Check VLDB property.

Copyright © 2024 All Rights Reserved

Error Code	Explanation
	Using zero checking on metrics in Intelligent Cubes causes the Intelligent Cubes to be unavailable for dynamic sourcing.
CMI_AGGREGATION_ON_ATTRIBUTES_RELATED_TO_TRANSFORMATION_ATTRIBUTES	Dynamic sourcing cannot be used because a transformation metric is included in the Intelligent Cube and it cannot be aggregated to the level required on the report.
CMI_ALL_METRICS_REMOVED_FROM_CUBE	The Intelligent Cube included metrics, but the metrics were later removed or no longer enable dynamic sourcing. This Intelligent Cube cannot be used for dynamic sourcing.
CMI_ANDNOT_OR_ORNOT_IN_FILTER	Dynamic sourcing cannot be used because filter qualifications are combined with AND NOT or OR NOT logical operators.
CMI_ATTRIBUTE_UNAVAILABLE	Dynamic sourcing cannot be used because an attribute form present in the report is not available in the Intelligent Cube.
CMI_BAD_OR_IN_FILTER	Dynamic sourcing cannot be used because two filter qualifications on different attributes or attribute elements are combined with the logical operator OR.
CMI_CG_OR_CONSOLIDATION	A custom group or consolidation is used on the report. Dynamic sourcing cannot be used for reports that include custom groups or consolidations.
CMI_COMPLEX_METRIC_EXPRESSION_ON_CUBE	Dynamic sourcing cannot be used for the Intelligent Cube because a metric's formula is too complex. Metrics that are too complex include compound metrics that are not defined as smart metrics, and simple metrics that use a formula other than a single aggregation function of a single fact. You can remove the metric or modify the metric's formula so that dynamic sourcing can be supported for the Intelligent Cube. For information on simple and compound metrics, see the Advanced

Copyright © 2024 All Rights Reserved

Error Code	Explanation
	Reporting Help.
CMI_COUNT_DISTINCT_FOUND	Dynamic sourcing cannot be used because the Count Distinct function is used.
CMI_CSI_OR_ELEMENT_BROWSING	The report cannot use dynamic sourcing because it is connected to a specific Intelligent Cube, or the user is browsing the elements of an attribute from within the report.
CMI_CUBE_TEMPLATE_MISSING_FILTER_ATTRIBUTE_FORM	Dynamic sourcing cannot be used because an attribute form used in the report's filter qualifications is not available in the Intelligent Cube.
CMI_CUBES_FOR_AD_HOC_DISABLED_FOR_REPORT	Dynamic sourcing is disabled for the report. To enable dynamic sourcing for a report, see *Enabling or disabling dynamic sourcing for reports, page 273*.
CMI_CUBES_FOR_AD_HOC_DISABLED_MASTERSWITCH	Dynamic sourcing is disabled for the project. To enable dynamic sourcing for a project, see *Enabling or disabling dynamic sourcing for projects, page 266*.
CMI_DB_ONLY_FUNCTION_IN_FILTER	Dynamic sourcing cannot be used because a passthrough or database function such as ApplySimple is used in the filter.
CMI_DB_ONLY_FUNCTION_IN_METRIC_FORMULA	Dynamic sourcing cannot be used because a passthrough or database function such as ApplySimple is used in a metric's formula.
CMI_DIFFERENT_AGG_TABLE_HIT	Dynamic sourcing cannot be used because of one of the following scenarios: A metric retrieves its data from different aggregate tables for the report and Intelligent Cube. The metric uses an aggregation function other than Sum for the aggregate table. The metric uses the Sum aggregation function, but the aggregate table is disabled for dynamic sourcing.

Copyright © 2024 All Rights Reserved

Error Code	Explanation
	For information on disabling dynamic sourcing for aggregate tables, see *Disabling dynamic sourcing for aggregate tables, page 284*.
CMI_DIFFERENT_DB_ROLES	Dynamic sourcing cannot be used because the report and Intelligent Cube use different database instances.
CMI_DIFFERENT_METRIC_FORMULA	This error signifies that, due to differing metric formulas, a metric in an Intelligent Cube is not a match for a metric on a report. All metrics on an Intelligent Cube are checked to determine if they are a match for a metric on a report. This process is done until a match is found, or no match is found and dynamic sourcing is not used (finding no matches for a metric returns the error CMI_NO_GOOD_CANDIDATE_METRICS_FOUND).
CMI_DIFFERENT_TRANSFORMATIONS_USED	Dynamic sourcing cannot be used because a metric on the report and a metric on the Intelligent Cube use different transformations.
CMI_DIMMETRIC_BRANCH_IN_CUBE	This error commonly occurs because a compound metric is included in the Intelligent Cube, which is preventing the Intelligent Cube from being available for dynamic sourcing. Only compound metrics that are defined as smart metrics can be included in Intelligent Cubes.
CMI_DISTINCT_FOUND_ON_AGGREGATION_FUNCTION	Dynamic sourcing cannot be used because the Distinct parameter is used in the aggregation of a metric and the level of data does not match between the report and the Intelligent Cube. To support dynamic sourcing when a Distinct parameter is used in an aggregation, the Intelligent Cube and report must contain the exact same level of data.
CMI_DISTINGUISH_DUPLICATED_ROWS_MISMATCH_FOR_CUBE_AND_REPORT	The report cannot connect to the Intelligent Cube using dynamic sourcing because the Distinguish Duplicated Rows VLDB property is not defined the same in the report and the Intelligent Cube. For information on this

Copyright © 2024 All Rights Reserved

Error Code	Explanation
	and other VLDB properties, see the System Administration Guide.
CMI_FACT_ENTRY_LVL_ERROR	Dynamic sourcing cannot be used because a metric uses a fact with a fact extension or degradation. For information on fact level extensions, see the *Facts* section in the Project Design Help.
CMI_FACT_ENTRY_LVL_GREATER_THAN_A_FILTER_LEVEL	Dynamic sourcing cannot be used because the fact entry level of a metric on the report and the filtering criteria of the report cannot be satisfied by the Intelligent Cube. For example, a report includes a metric that has a fact entry level of Year, and the filter includes a qualification on the Month attribute, which is at a lower logical level. If the lower level attribute used in the filter (for this example, Month) is not included in the Intelligent Cube, dynamic sourcing cannot be used. To support dynamic sourcing, the lower level attribute must be included in the Intelligent Cube.
CMI_FACT_NOT_SUPPORTED_AT_REQUIRED_CUBE_LEVEL	The report cannot connect to the Intelligent Cube using dynamic sourcing because a metric on the report is at a higher level than what is available on the Intelligent Cube. For example, a report requires metric data to be displayed at a yearly level, but the Intelligent Cube can only display metric data at the quarterly level. Dynamic sourcing cannot be used to connect the report to this Intelligent Cube.
CMI_FILTER_FACT_NODE_NOT_SUPPORTED	Dynamic sourcing cannot be used because a fact is included in a filter. This can only occur if you use one of the SQL optimization options available with the SQL Global Optimization VLDB property.
CMI_FILTER_KEY_ATTRIBUTE_UNAVAILABLE	Dynamic sourcing cannot be used because an attribute form is used in a conditional metric included on the report or in the report filter, and the attribute form is not included in the Intelligent Cube.

Copyright © 2024 All Rights Reserved

Error Code	Explanation
CMI_FILTER_METRIC_EXPR_NODE_NOT_SUPPORTED	Dynamic sourcing cannot be used because a metric qualification is used in the report or Intelligent Cube.
CMI_FILTER_METRIC_NODE_NOT_SUPPORTED	Dynamic sourcing cannot be used because a metric qualification is used in the report or Intelligent Cube.
CMI_FILTER_NOT_RESTRICTIVE_ENOUGH	The Intelligent Cube cannot be used for dynamic sourcing for the report because it cannot be verified that the Intelligent Cube filter on a particular metric is less restrictive than the report filter.
CMI_FILTER_PRED_NODE_NOT_SUPPORTED	Dynamic sourcing cannot be used because a metric qualification is used in the report or Intelligent Cube.
CMI_FILTER_RELATIONSHIP_NODE_NOT_SUPPORTED	Dynamic sourcing cannot be used because a relationship qualification that cannot be supported is used in the report or Intelligent Cube.
CMI_FILTER_TABLE_NODE_NOT_SUPPORTED	Dynamic sourcing cannot be used because a relationship qualification that cannot be supported is used in the report or Intelligent Cube.
CMI_FILTER_UNIT_ROOT_NODE_NOT_SUPPORTED	Dynamic sourcing cannot be used because a report is used as a filter of the report or Intelligent Cube.
CMI_FILTERING_KEY_ATTRIBUTE_UNAVAILABLE	Dynamic sourcing cannot be used because an attribute form is used in a conditional metric included on the report or in the report filter, and the attribute form is not included in the Intelligent Cube.
CMI_FILTERS_CANNOT_ENSURE_ALL_DATA_NEEDED_IS_IN_CUBE	The Intelligent Cube cannot be used for dynamic sourcing for the report because it cannot be verified that the Intelligent Cube filter is less restrictive than the report filter.
CMI_FULL_OUTER_JOIN_NOT_SUPPORTED	Dynamic sourcing cannot be used because full outer joins are not supported or enabled in the project or by the database.
CMI_GROUP_BY_ATTRIBUTE_	Dynamic sourcing cannot be used because an attribute

Copyright © 2024 All Rights Reserved

Error Code	Explanation
UNAVAILABLE	form used to define the level for a level metric is not included in the Intelligent Cube.
CMI_IF_NO_METRIC_REPORT_AND_CUBE_LOWEST_LVL_MUST_BE_THE_SAME	Dynamic sourcing cannot be used for the report because the report contains no metrics, and the Intelligent Cube does not use the same level as the report. For reports with no metrics, Intelligent Cubes must have the same level as these reports to use dynamic sourcing.
CMI_INCOMPATIBLE_DATA_TYPES_IN_FILTER	Dynamic sourcing cannot be used because a filter qualification includes two incompatible data types. One scenario in which this can occur is when identity forms are used to filter data, and smaller identity numbers use an integer type, while larger identity numbers use big decimal.
CMI_INCOMPLETE_HIERARCHY_DANGEROUS_ATTRIBUTE	The report or Intelligent Cube includes an attribute that is defined as possibly containing null values, which disables dynamic sourcing for the report or Intelligent Cube. For information on how to disable or enable dynamic sourcing for an attribute, see *Disabling dynamic sourcing for attributes, page 277*.
CMI_INVALID_BETWEEN_IN_FILTER	Dynamic sourcing cannot be used because an invalid filter qualification that uses the Between operator is used. The first value must be less than or equal to the second value.
CMI_JOINT_PARENT_OR_JOINT_CHILD_OR_MANY_TO_MANY_NOT_SUPPORTED	Joint child relationships and many-to-many relationships are not supported for dynamic sourcing.
CMI_METRIC_AND_TEMPLATE_LVL_NOT_MATCHING_IN_CUBE	The metric in an Intelligent Cube cannot be used for dynamic sourcing because it is a level metric. For information on level metrics, see the Advanced Reporting Help.
CMI_METRIC_BREAK_BY_NOT_ON_	Dynamic sourcing cannot be used because the break by

Copyright © 2024 All Rights Reserved

Error Code	Explanation
CUBE_TEMPLATE	parameter for a metric on the Intelligent Cube is not included in the Intelligent Cube. The break by parameter determines when calculations such as running summations or moving averages restart their calculations. For example, a running sum of revenue uses a break by parameter of the Year attribute. Without the Year attribute on the Intelligent Cube, the metric cannot be calculated correctly.
CMI_METRIC_CROSS_JOINED_NO_GROUP_BY	Dynamic sourcing cannot be used because a cross join is used for a metric.
CMI_METRIC_FITLER_KEY_NOT_IN_TEMPLATE	Dynamic sourcing cannot be used because an attribute form is used in a conditional metric included on the report or the report filter, and the attribute form is not included in the Intelligent Cube.
CMI_METRIC_GROUP_BY_KEY_NOT_IN_TEMPLATE	Dynamic sourcing cannot be used because an attribute form used to define the level for a level metric is not included in the Intelligent Cube. This can also be caused by defining the filtering of a target for a level metric to be absolute or ignore.
CMI_METRIC_NOT_OJ	The Intelligent Cube cannot be used for dynamic sourcing because it includes a metric that does not use outer joins. Intelligent Cubes of this type can be made available for dynamic sourcing by using outer joins for all metrics in the Intelligent Cube, or selecting the **Allow dynamic sourcing even if outer join properties are not set** option for the Intelligent Cube (see *Enabling or disabling dynamic sourcing for Intelligent Cubes, page 274*).
CMI_METRIC_NOT_SAFE	The metric has been disabled for dynamic sourcing, as described in *Disabling dynamic sourcing for metrics, page 282*.

Copyright © 2024 All Rights Reserved

Error Code	Explanation
CMI_METRIC_WITH_CONDITIONALITY_IN_CUBE	The Intelligent Cube cannot be used for dynamic sourcing because it includes a conditional metric. For information on conditional metrics, see the Advanced Reporting Help.
CMI_METRICS_WITH_NESTED_AGGREGATION_NOT_SUPPORTED	Dynamic sourcing cannot be used because nested aggregation is used for a metric. Nested aggregation is not supported for dynamic sourcing.
CMI_NO_ERROR	No error occurred, the report can use dynamic sourcing if an Intelligent Cube with the correct data requirements can be found.
CMI_NO_GOOD_CANDIDATE_METRICS_FOUND	Dynamic sourcing cannot be used because no suitable metric can be found in the Intelligent Cube for a metric included in the report.
CMI_NON_AGG_FUNCTION_FOUND	Dynamic sourcing cannot be used because a non-aggregation function such as First is used, and the metric level in the report is different than the metric level in the Intelligent Cube.
CMI_NON_AGG_NOT_REPORT_LEVEL_METRIC_FOUND	Dynamic sourcing cannot be used because a metric uses a function that cannot use dynamic aggregation by default. This prevents the metric from being able to display data at the level defined for the Intelligent Cube. For information on functions that are not dynamically aggregated by default, see *Metrics that are not dynamically aggregated by default, page 330*.
CMI_NON_HDA_USER_CANNOT_USE_HDA_CUBE	Dynamic sourcing cannot be used because the Intelligent Cube uses MultiSource Option to connect to multiple data sources, but the user running the report dose not have the privileges to use the MultiSource Option. For information on MultiSource Option, see the Project Design Help.
CMI_NONAGGREGATABLE_METRIC	Dynamic sourcing cannot be used because a metric that cannot be aggregated is used, and the metric level in

Copyright © 2024 All Rights Reserved

Error Code	Explanation
	the report is different than the metric level in the Intelligent Cube.
CMI_NONAGGREGATION_IN_METRIC	The Intelligent Cube cannot be used for dynamic sourcing because it includes a metric that cannot be aggregated, and the metric level in the report is different than the metric level in the Intelligent Cube.
CMI_NULL_CHECKING_FOR_ANAL_ENGINE_MISMATCH	Dynamic sourcing cannot be used because the option selected for the Null checking for Analytical Engine VLDB Property is different for the report and Intelligent Cube. The same option must be used to match a report and Intelligent Cube for dynamic sourcing.
CMI_OLD_OLAP_FUNCTION_IN_METRIC	Dynamic sourcing cannot be used because old OLAP Services behavior is enabled. To use the new OLAP Services behavior, select the **Use OLAP's new behavior** option for the **OLAP New Behavior** VLDB Property.
CMI_PERFORM_OR_FAILED	Dynamic sourcing cannot be used because filtering qualifications could not be combined correctly with the logical operator OR. To support dynamic sourcing you can remove or modify filter qualifications. For information on how filter qualifications can be used with dynamic sourcing, see *Best practices for supporting dynamic sourcing, page 257*.
CMI_REPORT_IS_XDA	The report is either a: OLAP cube report connected to SAP BI, Microsoft Analysis Services, or Hyperion Essbase Query Builder report Freeform SQL report These types of reports cannot use dynamic sourcing.
CMI_REPORT_LIMIT_ON_CUBE_FOUND	A report limit is present on the Intelligent Cube.

Copyright © 2024 All Rights Reserved

Error Code	Explanation
	Intelligent Cubes with report limits cannot be used for dynamic sourcing.
CMI_REPORT_OLAP_METRIC_BREAK_BY_MISSING_FROM_CUBE_TEMPLATE	Dynamic sourcing cannot be used because the break by parameter for a metric on a report is not included in the Intelligent Cube. The break by parameter determines when calculations such as running summations or moving averages restart their calculations. For example, a running sum of revenue uses a break by parameter of the Year attribute. Without the Year attribute on the Intelligent Cube, the metric cannot be calculated correctly.
CMI_REPORT_OLAP_METRIC_SORT_BY_MISSING_FROM_CUBE_TEMPLATE	Dynamic sourcing cannot be used because the sort by parameter for a metric is not included in the Intelligent Cube. The sort by parameter determines the order in which calculations such as running summations or moving averages perform their calculations. For example, a running sum of revenue uses a sort by parameter of the Quarter attribute. Without the Quarter attribute on the Intelligent Cube, the metric cannot be calculated correctly.
CMI_REPORT_TEMPLATE_ATTRIBUTE_MISSING_FROM_CUBE_TEMPLATE	Dynamic sourcing cannot be used because an attribute included in the report is not included in the Intelligent Cube.
CMI_REPORTS_WITH_ZERO_RL_METRICS_CAN_ONLY_HIT_CUBE_WITH_FOJ_SUPPORT	Dynamic sourcing cannot be used because the report uses metrics that are not at the report level. To support dynamic sourcing for these types of reports, Intelligent Cubes must support outer joins. For information on outer join support with Intelligent Cubes, see *Enabling or disabling dynamic sourcing for Intelligent Cubes, page 274*.
CMI_SECURITY_FILTER_KEY_ATTRIBUTE_UNAVAILABLE	Dynamic sourcing cannot be used because an attribute used to define the security filter of the user executing the report is not included in the Intelligent Cube.

Copyright © 2024 All Rights Reserved

Error Code	Explanation
CMI_SECURITY_FILTER_LEVEL_ATTRIBUTE_UNAVAILABLE	Dynamic sourcing cannot be used because an attribute used to define the security filter of the user executing the report is not included in the Intelligent Cube.
CMI_STRING_COMPARISON_NOT_ALLOWED	Dynamic sourcing cannot be used because the report uses a filter qualification that qualifies on the text data of an attribute form, and these types of qualifications have been disabled for dynamic sourcing. For information on the **String Comparison Behavior** VLDB property which defines this functionality, see *Supporting filtering on attributes for dynamic sourcing, page 280*.
CMI_STRING_COMPARISON_NOT_ALLOWED_IN_SORT_BY	Dynamic sourcing cannot be used because the report uses a sort on the text data of an attribute form, and these types of sorts have been disabled for dynamic sourcing. For information on the **String Comparison Behavior** VLDB property which defines this functionality, see *Supporting filtering on attributes for dynamic sourcing, page 280*.
CMI_UNEXPECTED_REPORT_TYPE	The report is of a type that cannot use dynamic sourcing, which includes: MDX cube report Query Builder report Freeform SQL report Data marts
CMI_UNSATISFIABLE_CONDITION_IN_FILTER	Dynamic sourcing cannot be used because an unsupported condition is included in the filter. For a list of features that are not supported for dynamic sourcing, see *Features that prevent the use of dynamic sourcing, page 261*.
CMI_UNSUPPORTED_ELEMENT_IN_FILTER	Dynamic sourcing cannot be used because an element that is not supported for dynamic sourcing is included in the filter. For a list of features that are not supported for

Copyright © 2024 All Rights Reserved

Error Code	Explanation
	dynamic sourcing, see *Features that prevent the use of dynamic sourcing, page 261*.
CMI_UNSUPPORTED_EXPR_TYPE_IN_FILTER	Dynamic sourcing cannot be used because an unsupported expression is included in a filter. For a list of features that are not supported for dynamic sourcing, see *Features that prevent the use of dynamic sourcing, page 261*.
CMI_UNSUPPORTED_FILTER_NODE	Dynamic sourcing cannot be used because an unsupported filter type is used. For a list of features that are not supported for dynamic sourcing, see *Features that prevent the use of dynamic sourcing, page 261*.
CMI_UNSUPPORTED_FUNCTION_IN_FILTER	Dynamic sourcing cannot be used because an unsupported function is included in a filter. Supported functions in filters are listed in *Features that can be used with dynamic sourcing, page 260*.
CMI_USER_LOCALE_NOT_SUPPORTED_IN_CUBE	The report cannot connect to the Intelligent Cube using dynamic sourcing because the Intelligent Cube is not available for the locale of the user viewing the report. For MicroStrategy projects that support multiple languages and character sets, users can view reports that display data for their locale. To use dynamic sourcing, an Intelligent Cube must be available for the user's locale.
CMI_USER_DOESNT_HAVE_DYNAMIC_SOURCING_PRIVILEGE	The report cannot connect to the Intelligent Cube using dynamic sourcing because the user account being used does not have the Use Dynamic Sourcing privilege. This privilege is required to use dynamic sourcing.
CMI_VLDB_SETTING_MISMATCH_FOR_CUBE_AND_REPORT	The report cannot connect to the Intelligent Cube using dynamic sourcing because the options used for various VLDB properties do not match between these two objects. For a list of VLDB properties that prevent reports from using dynamic sourcing, see *Features that*

Copyright © 2024 All Rights Reserved

Error Code	Explanation
	prevent reports from using dynamic sourcing, page 262. For information on VLDB properties, see the System Administration Guide.
CMI_ZERO_METRICS_NEEDED	The report cannot use dynamic sourcing because it does not contain any metrics, and no matching Intelligent Cube with no metrics can be found.
CMI_UNSUPPORTED_INCREMENTAL_REFRESH	The Intelligent Cube cannot be used for dynamic sourcing due to an unsupported incremental refresh that modifies the data in the Intelligent Cube.
CMI_RAGGED_HIERARCHY_NO_METRICS_NOT_SUPPORTED	The report cannot be used for dynamic sourcing because it has no metrics, and attributes from ragged hierarchies. For more information, see the MDX Cube Reporting Help.
CMI_NONAGG_BY_LKP_ON_INCR_REFR_CUBE_NOT_SUPPORTED	The Intelligent Cube cannot be used for dynamic sourcing because it uses non-aggregation by lookup and is incrementally refreshed
CMI_DIFFERENT_MDX_SOURCE_TABLE_USED	The MDX-based Intelligent Cube cannot be used for dynamic sourcing because the report and Intelligent Cube point to different MDX data sources.

Copyright © 2024 All Rights Reserved

Dynamic Aggregation

Copyright © 2024 All Rights Reserved

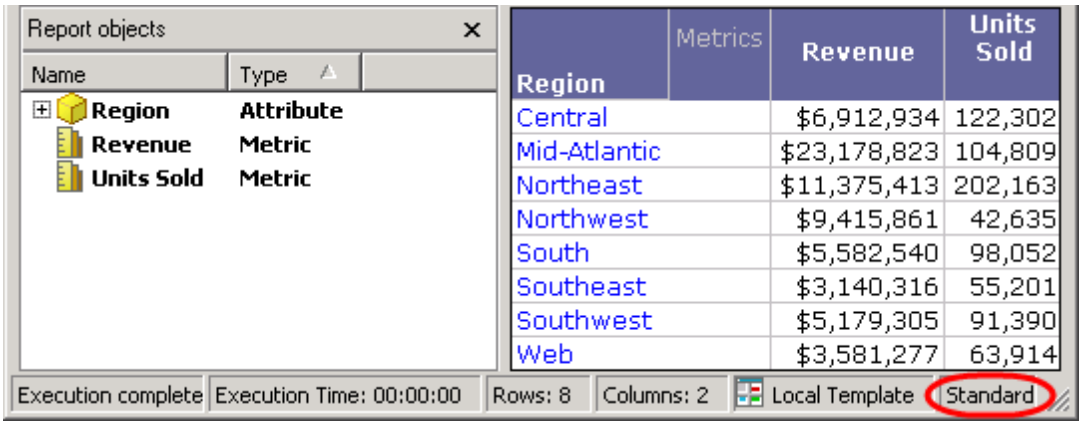
Dynamic aggregation is an OLAP Services feature that allows you to change the level of report aggregation on the fly, while you are reviewing the report results. This feature allows metric values to be aggregated at different levels based on the attributes included on the report without having to re-execute the report against the data warehouse. Dynamic aggregation occurs when you move attributes and metrics between the Report Objects pane and the report grid.

To better understand dynamic aggregation, consider a report that includes two attributes, Region and Category, and two metrics, Revenue and Units Sold. A subset of the report results are shown below.

Region	Category	Metrics	Revenue	Units Sold
Central	Books		$251,279	15,715
	Electronics		$5,307,467	15,278
	Movies		$241,405	15,500
	Music		$1,112,782	75,809
Mid-Atlantic	Books		$210,320	13,023
	Electronics		$22,565,830	65,424
	Movies		$208,237	13,299
	Music		$194,436	13,063
Northeast	Books		$2,048,826	127,017
	Electronics		$8,563,073	24,974
	Movies		$395,246	25,355
	Music		$368,269	24,817

To remove an attribute from the report without using dynamic aggregation, right-clicking it (the Category attribute is used for this example) and select **Remove from Report**. The data for Revenue and Units Sold is aggregated to the Region level, resulting in the report below.

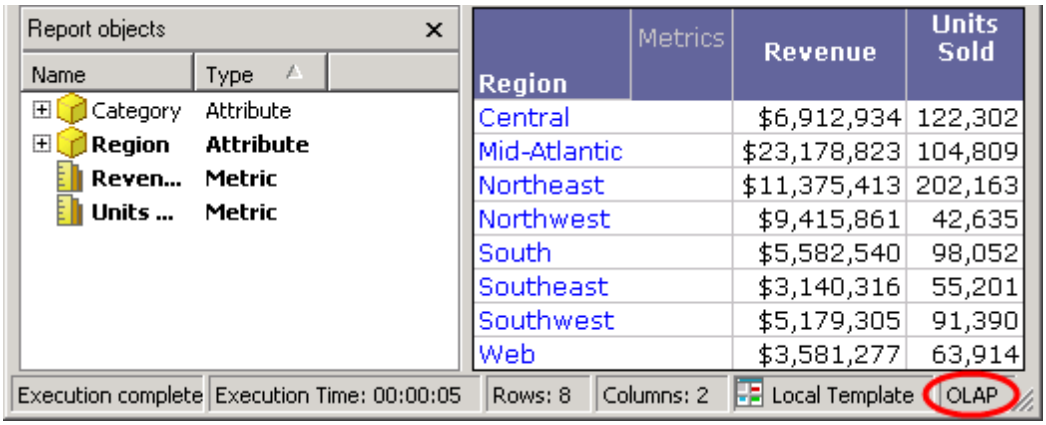
Copyright © 2024 All Rights Reserved



Notice that the Category attribute is not in the Report Objects pane, and the report remains a **Standard** report as indicated in the bottom right corner.

When you remove an attribute or metric from the report, a message is displayed that indicates the manipulation causes the report to be re-executed. This is because standard metric aggregation needs to be re-executed against the data warehouse, and therefore requires new report SQL to be generated.

Now consider the same report, but this time you use dynamic aggregation to return a different view of the data. Instead of removing Category from the report, Category is moved to the Report Objects pane, as shown in the image below.



Although the report data is exactly the same as the previous report with standard aggregation, dynamic aggregation has the following differences:

Copyright © 2024 All Rights Reserved

- **No re-execution and no new SQL required**: Dynamic aggregation does not require re-execution against the data warehouse. Aggregating data within the report can improve the performance of your system by reducing the load on the data warehouse. Dynamic aggregation also improves performance by returning the new report results as soon as the attribute or metric is moved to Report Objects, instead of having to wait for the results to be returned from the data warehouse.
- **Attributes remain in Report Objects**: Category is now in the Report Objects pane and is no longer displayed with bold formatting. Since Category is part of the report definition, it can be used to define the view of the report. For example, you can build a view filter to return Revenue only for the electronics category. A standard report would have to use the report filter to filter on an attribute not on the report, which would also require re-execution against the data warehouse.

  With attributes available in the Report Objects pane, you can easily add them back onto the report grid.
- **OLAP report**: The report is marked as OLAP (as indicated in the bottom right corner of the Report Editor) because of the dynamic aggregation performed.

This section discusses the following aspects of dynamic aggregation:

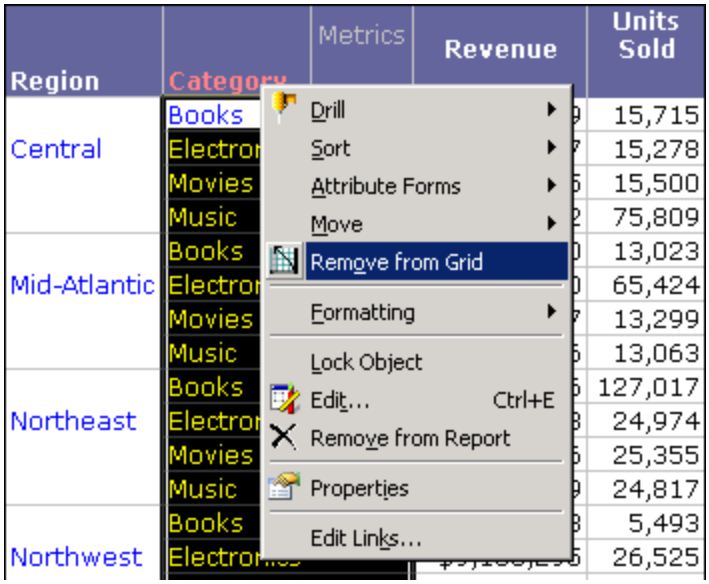
- *Using dynamic aggregation, page 325*
- *Functions used in dynamic aggregation, page 327*
- *View filter effect on dynamic aggregation, page 343*

Using dynamic aggregation

To change the aggregation level of a report, you must either remove an attribute from a report entirely, or use dynamic aggregation to remove the attribute from the report grid only. With dynamic aggregation, the attribute is still a part of the report definition and is included in the Report Objects.

Copyright © 2024 All Rights Reserved

To use dynamic aggregation, right-click an attribute on a report grid and select **Remove from Grid**, as shown below.

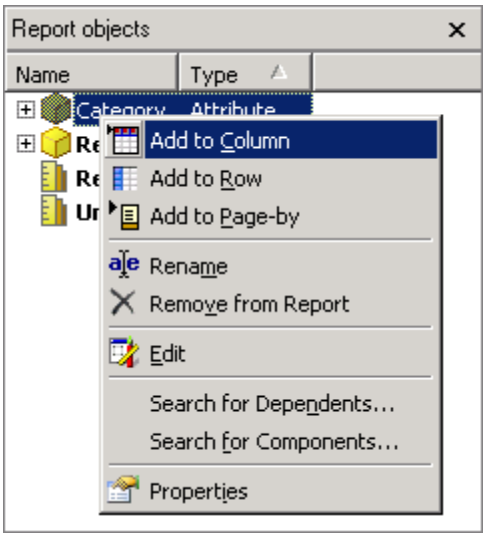


Be careful not to select **Remove from Report**. If you select this option, the attribute is completely removed from the report definition, and the report is re-executed against the data warehouse.

You need the Use Report Objects Window (Developer) and/or the Web Use Report Objects Window (Web) privileges to use dynamic aggregation. These privileges are part of OLAP Services.

Another benefit of using dynamic aggregation is that the attributes removed from the report grid can be easily included back onto the report grid or page-by area. To do this, right-click an attribute in the Report Objects pane and select **Add to Column**, **Add to Row**, or **Add to Page-by** to add an attribute from the Report Objects pane onto the report. The options to move an attribute to the report are shown in the image below.

Copyright © 2024 All Rights Reserved



You can also move attribute forms between the Report Objects pane and the report grid. If you only move an attribute form and not the attribute itself, dynamic aggregation is not triggered. For example, if the attribute forms Last Name and First Name for the attribute Customer are displayed on a report, you can move First Name to the Report Objects pane without triggering dynamic aggregation. The same First Name attribute form can be moved back to the report grid without triggering dynamic aggregation.

An attribute must have at least one attribute form displayed to be on the report grid.

Functions used in dynamic aggregation

The dynamic aggregation feature of a metric dictates what function to use when the Analytical Engine must dynamically aggregate the metric. You can modify the dynamic aggregation function from the **Subtotals / Aggregation** tab of the Metric Editor. A metric has a default aggregation function if the metric is defined with certain functions, as described below:

- A metric defined with one or more of the functions Sum, Count, Minimum, Maximum, and Product has a default aggregation function. For more information on metrics that have a default dynamic aggregation function,

Copyright © 2024 All Rights Reserved

see *Metrics with default dynamic aggregation functions, page 328*.

- A metric defined with any other functions than the ones listed above does not have a default aggregation function. No aggregation function is used as the default and the dynamic aggregation of one of these metrics returns a null value. The null value is replaced with dashes (--) to indicate that the metric cannot be calculated at the higher level. For more information on metrics that do not have a default dynamic aggregation function, see *Metrics that are not dynamically aggregated by default, page 330*.

  The exception is a shortcut metric defined in a report, document, or Visual Insight quick dashboard created in MicroStrategy Web. Dynamic aggregation and subtotals are calculated correctly for these shortcut metrics, even if they do not contain the functions listed above. For instructions to create shortcut metrics, which are a type of derived metric, see the MicroStrategy Web Help.

By default, the level of the dynamic aggregation is defined by the metric that is being aggregated. You can define the Subtotal Dimensionality Use VLDB property so that the dynamic aggregation uses the level of the metric's dynamic aggregation function instead. For a more detailed description of this VLDB property, along with an example of its use, see the System Administration Guide.

Metrics with default dynamic aggregation functions

A metric defined with functions such as Sum or Product can be dynamically aggregated because these functions can be calculated at a higher level than the initial calculation. Any metric defined with one of the following functions has a default dynamic aggregation function that allows the data for the metric to be aggregated at different levels on the fly:

Copyright © 2024 All Rights Reserved

- **Sum or Count:** Sum is used as the aggregation function.

  If the Count function is set to count distinct entries, it cannot use Sum as its dynamic aggregation function and returns null values instead. The exception is for a count distinct function in a shortcut metric defined in a report, document, or Visual Insight quick dashboard created in MicroStrategy Web. For instructions to create a shortcut metric, which is a type of derived metric, see the MicroStrategy Web Help.

- **Minimum:** Minimum (Min) is used as the aggregation function.
- **Maximum**: Maximum (Max) is used as the aggregation function.
- **Product**: Product is used as the aggregation function.

For example, consider the **Dynamic Aggregation - Region - Employee** report from the MicroStrategy Tutorial project. To focus only on those metrics that have default dynamic aggregation functions, move the metrics Standard Deviation of Revenue and Count Distinct (Items Sold) to the Report Objects pane (for steps, see *Using dynamic aggregation, page 325*). The report results for the Central and Mid-Atlantic regions are shown below.

Region	Employee		Metrics	Revenue	Max Revenue	Min Revenue
Central	Ellerkamp	Nancy		$1,169,245	$53,460.00	$102.00
	Gale	Loren		$2,262,146	$76,740.00	$199.20
	Torrison	Mary		$2,364,993	$76,800.00	$195.30
	Zemlicka	George		$1,116,549	$45,240.00	$82.20
	Subtotals			**$6,912,934**	**$76,800.00**	**$82.20**
Mid-Atlantic	Bernstein	Lawrence		$5,295,910	$232,500.00	$109.95
	Brown	Vernon		$1,803,732	$93,300.00	$20.25
	Corcoran	Peter		$1,709,388	$62,400.00	$17.40
	Folks	Adrienne		$5,708,091	$278,520.00	$117.45
	Hollywood	Robert		$5,272,618	$232,920.00	$106.35
	Ingles	Walter		$1,013,388	$46,800.00	$13.00
	Smith	Thomas		$1,183,056	$58,800.00	$11.00
	Young	Sarah		$1,192,641	$47,520.00	$16.00
	Subtotals			**$23,178,823**	**$278,520.00**	**$11.00**

Copyright © 2024 All Rights Reserved

A custom subtotal has been included on this report to display subtotals for the different revenue calculations. For information on creating custom subtotals, see the Advanced Reporting Help.

The Revenue, Max Revenue, and Min Revenue metrics use total, maximum, and minimum subtotals respectively. These are the same default dynamic aggregation functions that are used for these three metrics because they are built with the Sum, Max, and Min functions respectively. You can verify that this is true by moving Employee to the Report Objects pane, which triggers dynamic aggregation, causing the metrics to be aggregated at the regional level.

Region / Metrics	Revenue	Max Revenue	Min Revenue
Central	$6,912,934	$76,800.00	$82.20
Mid-Atlantic	$23,178,823	$278,520.00	$11.00
Northeast	$11,375,413	$123,900.00	$42.00
Northwest	$9,415,861	$205,260.00	$39.25
South	$5,582,540	$107,760.00	$183.20
Southeast	$3,140,316	$33,600.00	$42.00
Southwest	$5,179,305	$55,320.00	$40.00
Web	$3,581,277	$112,500.00	$279.60

Notice that the values for the Northeast and Mid-Atlantic regions are the same values as the regional subtotals in the report prior to triggering dynamic aggregation. When the three metrics are aggregated at the regional level, each metric uses its default dynamic aggregation function to perform the calculation.

Metrics that are not dynamically aggregated by default

The ability to aggregate data at a higher level in memory is useful for quick report interaction and analysis. However, due to their aggregation function, some metrics cannot be dynamically aggregated. For certain metric aggregation functions, recalculating the data at the higher level would yield erroneous or null values if using only the data in the report. The metrics would need to be re-executed against the data warehouse to return correct values.

Copyright © 2024 All Rights Reserved

Metrics that cannot be calculated correctly using a dynamic aggregation function have the default dynamic aggregation function set to none. Metrics have their default aggregation function set to none if they are defined with the following functions:

- **Average (Avg)**: Sum of input values divided by number of input values.
- **Count (Distinct=true)**: Number of distinct input values.
- **Geometric mean (Geomean)**: Square root of the product of input values.
- **Median**: Middle value when input values are sorted.
- **Mode**: Most frequently found input value.
- **Standard Deviation (Stdev): Statistical distribution of input values.**
- **Variance (Var)**: Square of the standard deviation of input values.
- **Non-group functions or arithmetic operators**: Metrics defined in this way are called compound metrics.

For information on supporting dynamic aggregation for compound metrics, see *Dynamic aggregation for compound metrics, page 336*. Dynamic aggregation is supported for a shortcut metric defined in a report, document, or Visual Insight quick dashboard created in MicroStrategy Web. Dynamic aggregation and subtotals are calculated correctly for these shortcut metrics, even if they contain the functions listed above. For instructions to create shortcut metrics, which are a type of derived metric, see the MicroStrategy Web Help.

When you use dynamic aggregation at a level the Analytical Engine considers erroneous for a metric defined with one or more of the functions listed above, null values, represented by dashes (--), are displayed on the report. For information on changing the display of null values, see *Changing the display of null values, page 340*.

This section provides the following information related to functions and metrics that do not support dynamic aggregation by default:

Copyright © 2024 All Rights Reserved

- *Example of exceptions to dynamic aggregation, page 332*
- *Returning correct metric values by accessing the data warehouse, page 333*
- *Estimating dynamic aggregation values with different aggregation functions, page 335*
- *Dynamic aggregation for compound metrics, page 336*
- *Changing the display of null values, page 340*

Example of exceptions to dynamic aggregation

This example uses the **Dynamic Aggregation - Region - Employee** report from the MicroStrategy Tutorial project. To look at only those metrics that do not have default dynamic aggregation functions, move the metrics Revenue, Max Revenue, and Min Revenue to the Report Objects pane. To use dynamic aggregation, the Employee attribute is also moved to the Report Objects pane. The report results are shown below.

Region / Metrics	Standard Deviation of Revenue	Count Distinct (Items Sold)
Central	$3,393.51	1,440
Mid-Atlantic	$13,931.67	2,879
Northeast	$6,524.70	2,160
Northwest	$5,070.02	1,080
South	$3,896.12	1,080
Southeast	$248.86	1,440
Southwest	$1,950.34	1,800
Web		360

By default, the Standard Deviation of Revenue and Count Distinct (Items Sold) metrics would return null values because they use the Standard Deviation and Count Distinct functions in their metric definitions, respectively. The default dynamic aggregation functions have instead been set to the functions used for their metric definitions.

Copyright © 2024 All Rights Reserved

Now compare these values to the values returned by executing against the data warehouse instead of using dynamic aggregation to calculate the values from the data within the report.

Region / Metrics	Standard Deviation of Revenue	Count Distinct (Items Sold)
Central	$9,289.44	360
Mid-Atlantic	$23,658.84	360
Northeast	$11,574.77	360
Northwest	$21,280.25	360
South	$9,780.08	360
Southeast	$3,945.78	360
Southwest	$5,484.26	360
Web	$17,419.65	360

You can see that the Count Distinct (Items Sold) value is 360 for the Northeast region, which is far different from the 2,160 value returned for the Northeast region in the report that uses dynamic aggregation. The report above is able to query the data warehouse and show the distinct items sold by all employees in the Northeast region. For example, if Employee A, Employee B, and Employee C all sell one or more wrenches, the item is only counted as one distinct item for the Northeast region.

Dynamic aggregation uses the data available in the report. In this example, all the values for each employee in a given region are simply added together. When the calculation is performed, there is no way of relating which distinct items each employee sold. The calculation results in double-counting distinct items sold by two or more different employees. Rolling up data to a higher level for metrics defined with functions such as Standard Deviation and Average also perform erroneous calculations. For this reason, by default, metrics defined with certain functions return null values instead of erroneous results.

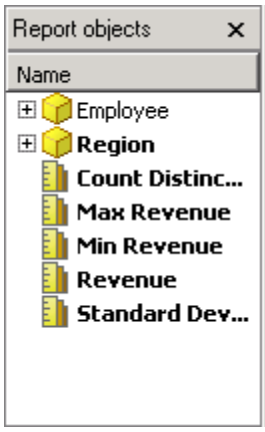
Returning correct metric values by accessing the data warehouse

If using dynamic aggregation in a report returns null or erroneous data, you can force the report to re-execute against the data warehouse instead of

Copyright © 2024 All Rights Reserved

triggering dynamic aggregation.

For example, move the Employee attribute to the Report Objects pane in the **Dynamic Aggregation - Region - Employee** report in the MicroStrategy Tutorial project. If you have not changed how the report displays null values, you get the results shown below.



Region	Metrics	Revenue	Standard Deviation of Revenue	Max Revenue	Min Revenue	Count Distinct (Items Sold)
Central		$6,912,934	--	$76,800.00	$82.20	--
Mid-Atlantic		$23,178,823	--	$278,520.00	$11.00	--
Northeast		$11,375,413	--	$123,900.00	$42.00	--
Northwest		$9,415,861	--	$205,260.00	$39.25	--
South		$5,582,540	--	$107,760.00	$183.20	--
Southeast		$3,140,316	--	$33,600.00	$42.00	--
Southwest		$5,179,305	--	$55,320.00	$40.00	--
Web		$3,581,277	--	$112,500.00	$279.60	--

Notice that Employee is still a part of the Report Objects pane.

To execute the report against the data warehouse, right-click the Employee attribute and select **Remove from Report**. The report is re-executed against the data warehouse, and returns the data shown below.

Region	Metrics	Revenue	Standard Deviation of Revenue	Max Revenue	Min Revenue	Count Distinct (Items Sold)
Central		$6,912,934	$9,289.44	$76,800.00	$82.20	360
Mid-Atlantic		$23,178,823	$23,658.84	$278,520.00	$11.00	360
Northeast		$11,375,413	$11,574.77	$123,900.00	$42.00	360
Northwest		$9,415,861	$21,280.25	$205,260.00	$39.25	360
South		$5,582,540	$9,780.08	$107,760.00	$183.20	360
Southeast		$3,140,316	$3,945.78	$33,600.00	$42.00	360
Southwest		$5,179,305	$5,484.26	$55,320.00	$40.00	360
Web		$3,581,277	$17,419.65	$112,500.00	$279.60	360

The report is re-executed against the warehouse without using dynamic aggregation only if all the attributes in the Report objects pane are on the report.

Copyright © 2024 All Rights Reserved

Note that taking Employee off the report changes the data definition of the report, instead of the view definition. While you get the correct results, you are no longer taking advantage of dynamic aggregation to perform the calculations. The SQL must be regenerated and executed against the data warehouse to retrieve the results.

Estimating dynamic aggregation values with different aggregation functions

If you use a function that does not support dynamic aggregation, you can use a different function to estimate the values using dynamic aggregation. This tactic depends greatly on the function used and the amount of error that is allowable for your report results. For steps on how to change the default aggregation function, see *Changing the default dynamic aggregation function, page 340*.

The **Dynamic Aggregation - Region - Employee** from the MicroStrategy Tutorial project is used as the base report to illustrate this tactic. Move all the attributes and metrics to the Report Objects pane except for the attribute Region and the metric Standard Deviation of Revenue. When you run this report, null values are displayed for the metric because the metric's function does not support dynamic aggregation.

You can estimate these values by using the Average function as the dynamic aggregation function. This estimation is possible because the Standard Deviation and Average formulas are similar in this situation.

You can change the dynamic aggregation function in the Subtotals / Aggregation tab of the Metric Editor. The report below shows the results for the metric using the Average dynamic aggregation function on the left, compared to the Standard Deviation function on the right, run against the data warehouse.

Copyright © 2024 All Rights Reserved

Region	Metrics	Standard Deviation of Revenue AVG
Central		$8,670.24
Mid-Atlantic		$18,995.93
Northeast		$9,333.97
Northwest		$20,831.90
South		$9,087.91
Southeast		$3,941.30
Southwest		$5,105.98
Web		$17,419.65

Region	Metrics	Standard Deviation of Revenue
Central		$9,289.44
Mid-Atlantic		$23,658.84
Northeast		$11,574.77
Northwest		$21,280.25
South		$9,780.08
Southeast		$3,945.78
Southwest		$5,484.26
Web		$17,419.65

Using the average function to perform the dynamic aggregation estimates the metric values to within $1,000 for all but two regions.

The accuracy of data returned when using a different dynamic aggregation function than the function used to define the metric depends on the similarity of the functions. For example, you cannot expect an accurate estimation of values if you use Sum as the dynamic aggregation function for a metric defined with the Standard Deviation function.

You can use any function as the dynamic aggregation function of a metric. However, be aware that not all functions are well suited for dynamic aggregation (see *Metrics that are not dynamically aggregated by default, page 330*).

You can also create your own subtotal to use as the dynamic aggregation function. You cannot directly use a function as the dynamic aggregation function of a metric, you must create a subtotal that uses the function in its definition. For steps to create a subtotal, refer to the Report Designer Help.

Dynamic aggregation for compound metrics

A compound metric is any 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 RunningSum or

Copyright © 2024 All Rights Reserved

Rank. The operators and functions can be applied to facts, attributes, or metrics.

This section covers compound metrics as they are used with dynamic aggregation. For complete information on compound metrics, see the Advanced Reporting Help.

For example, a metric defined with the formula `Sum(Revenue) - Sum(Cost)` is a compound metric because of the minus (-) operator. Another example of a compound metric is one defined with the formula `RunningSum(Revenue)` because of the RunningSum function.

By default, dynamic aggregation causes compound metrics to return null values. One reason for this is that the calculation of the formula is performed before it is dynamically aggregated, which can cause erroneous or null results depending on the formula (see *Functions used in dynamic aggregation, page 327*). Another possible reason is that when a metric has a complex formula for its definition, there is no way to assume the aggregation function that should be performed on the data. Rather than returning possibly erroneous results by default, you can decide how the metric handles dynamic aggregation.

For example, consider the **Dynamic Aggregation** report from the MicroStrategy Tutorial project, which contains the metrics Revenue, Cost, and Profit as well as the attributes Region and Employee. The report is shown below, with Employee off the report grid, but still in the Report Objects pane.

Region / Metrics	Revenue	Cost	Profit
Central	$6,912,934	$5,827,613	$1,085,321
Mid-Atlantic	$23,178,823	$19,171,243	$4,007,581
Northeast	$11,375,413	$9,365,673	$2,009,740
Northwest	$9,415,861	$7,780,870	$1,634,990
South	$5,582,540	$4,602,027	$980,513
Southeast	$3,140,316	$2,650,191	$490,125
Southwest	$5,179,305	$4,379,227	$800,078
Web	$3,581,277	$3,031,885	$549,392

Copyright © 2024 All Rights Reserved

The Profit metric in this case is based on a profit fact stored in the data warehouse. Suppose that your data warehouse only has facts for revenue and cost, but you want to create a metric that calculates profit. One way you can achieve this is by creating a profit metric called Compound Profit that combines the two metrics Revenue and Cost. The metric can be defined as `Sum(Revenue) - Sum(Cost)`.

You can also see a comparison of profit margins by including a Profit Margin metric on the report. The definition for Profit Margin is `Sum(Profit) / Sum(Revenue)`. When you add these two metrics to the Dynamic Aggregation report, the report returns the results shown below.

Region / Metrics	Revenue	Cost	Profit	Compound Profit	Profit Margin
Central	$6,912,934	$5,827,613	$1,085,321	--	--
Mid-Atlantic	$23,178,823	$19,171,243	$4,007,581	--	--
Northeast	$11,375,413	$9,365,673	$2,009,740	--	--
Northwest	$9,415,861	$7,780,870	$1,634,990	--	--
South	$5,582,540	$4,602,027	$980,513	--	--
Southeast	$3,140,316	$2,650,191	$490,125	--	--
Southwest	$5,179,305	$4,379,227	$800,078	--	--
Web	$3,581,277	$3,031,885	$549,392	--	--

With Employee in the Report Objects pane and not on the report grid, the metrics are dynamically aggregated to the regional level. The Profit metric is a simple sum of the Profit fact. This calculation can be aggregated from the Employee level to the Region level. By default, the metrics Compound Profit and Profit Margin do not use a dynamic aggregation function.

For Compound Profit to be dynamically aggregated correctly, you change the dynamic aggregation function to Sum. In this case the Sum function can be used to aggregate the data after the subtraction because the order of operations does not matter in a formula with only sum and subtract. For more information on changing the default dynamic aggregation function, see *Changing the default dynamic aggregation function, page 340*.

For Profit Margin, you cannot choose Sum as the dynamic aggregation function, because the definition of the metric includes a division. If Sum is

Copyright © 2024 All Rights Reserved

chosen, the division is performed first and then these values are added together, which would use the formula `Sum(Profit/Revenue)`. Recall that the definition of the metric is `Sum(Profit) / Sum(Revenue)`, performing the sum aggregations first and then dividing the sums.

To return valid results in this case, you can calculate the subtraction after dynamic aggregation. You can achieve this functionality by defining the compound metric as a smart metric. After you define both of the compound metrics as smart metrics, the correct results are returned, as shown below.

Region / Metrics	Revenue	Cost	Profit	Compound Profit	Profit Margin
Central	$6,912,934	$5,827,613	$1,085,321	$1,085,321	15.70%
Mid-Atlantic	$23,178,823	$19,171,243	$4,007,581	$4,007,581	17.29%
Northeast	$11,375,413	$9,365,673	$2,009,740	$2,009,740	17.67%
Northwest	$9,415,861	$7,780,870	$1,634,990	$1,634,990	17.36%
South	$5,582,540	$4,602,027	$980,513	$980,513	17.56%
Southeast	$3,140,316	$2,650,191	$490,125	$490,125	15.61%
Southwest	$5,179,305	$4,379,227	$800,078	$800,078	15.45%
Web	$3,581,277	$3,031,885	$549,392	$549,392	15.34%

You can define a metric as a smart metric using the procedure below. For an introduction to smart metrics, see *Derived elements in Report Services documents with multiple datasets, page 174*.

To define a metric as a smart metric

1. In MicroStrategy Developer, right-click a metric and select **Edit**. The Metric Editor opens.
2. On the **Subtotals / Aggregation** tab, select the **Allow Smart Metric** check box.
3. Click **Save and Close** to save your changes and close the Metric Editor.

Copyright © 2024 All Rights Reserved

Changing the display of null values

You can change the default display of null values caused by incorrect aggregation for your reports.

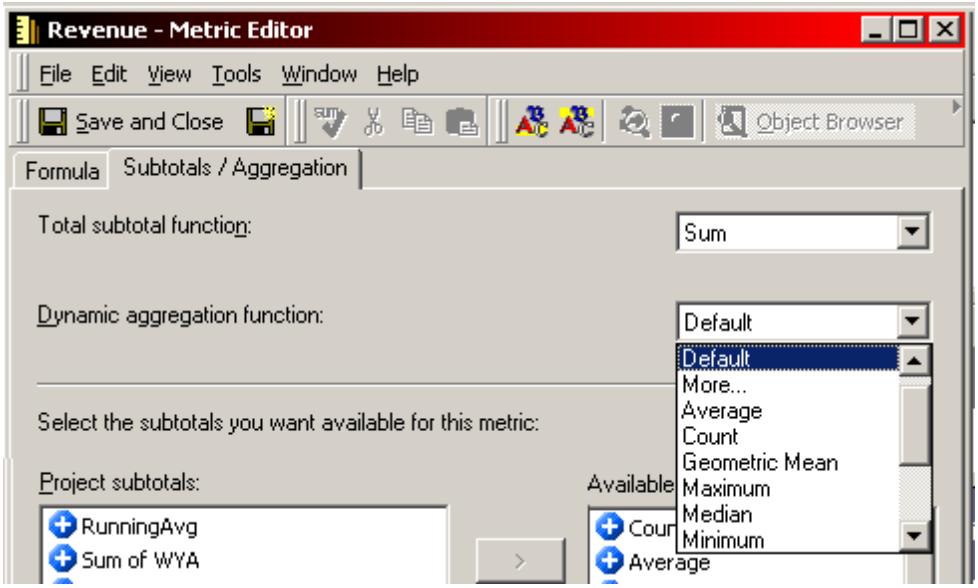
To change the display of aggregation null values

1. Open a report.
2. Choose **Data** > **Report Data Options**. The Report Data Options dialog box opens.
3. Expand the **Display** category, and then select **Null Values**. The Display - Null Values tab opens.
4. Under Aggregation null values, clear **Use Default**.
5. Replace "--" with the symbol you want to use for null values, for example, `00`, `null`, `blank`, and so on.

Changing the default dynamic aggregation function

The dynamic aggregation function is used to aggregate the data whenever the metric is dynamically aggregated, regardless of what function is used in the metric definition. You can change the default function used for dynamic aggregation in the Subtotals / Aggregation tab of the Metric Editor, as shown below.

Copyright © 2024 All Rights Reserved



The validity of the data depends on whether the function can correctly calculate the data within the report. For this reason, it is recommended that you evaluate your report requirements and consider the report results before you make the function selection for dynamic aggregation.

The functions that can be used as the dynamic aggregation function are:

- Average
- Count (Distinct=true)
- Geometric Mean
- Maximum
- Median
- Minimum
- Mode
- Product
- Standard Deviation
- Sum

Copyright © 2024 All Rights Reserved

- Variance
- User-defined function

You cannot directly use a function as the dynamic aggregation function of a metric. You must use or create a subtotal that uses the function in its definition. For information on creating a subtotal, search the online help for "Creating user-defined subtotals".

Metrics defined with certain functions use a default dynamic aggregation function that returns the correct results in most situations. For example, a metric defined with the Sum function uses the Sum function to dynamically aggregate its data. You can change the dynamic aggregation function for this type of metric, but it is recommended that you do not change these default functions as this can cause erroneous or null results for the metrics in a report. For more information on metrics with default dynamic aggregation functions, see *Metrics with default dynamic aggregation functions, page 328*.

Aggregating data from a report with certain functions can return erroneous or null results, and therefore the default dynamic aggregation function is set to none. You can set the default dynamic aggregation function so that these metrics return data instead of null values. For example, if a metric is defined with Standard Deviation, you can change the function used for dynamic aggregation from Default to Standard Deviation.

You can, however, use a function that does not match the function or functions used to define the metric. To see an example that uses this technique, see *Estimating dynamic aggregation values with different aggregation functions, page 335*.

To change the default dynamic aggregation function

1. In MicroStrategy Developer, right-click a metric and select **Edit**. The Metric Editor opens.

Copyright © 2024 All Rights Reserved

2. On the **Subtotals / Aggregation** tab, select a function from the **Dynamic aggregation function** drop-down list.
3. Click **Save and Close**.

View filter effect on dynamic aggregation

Dynamic aggregation can be affected by the use of view filters used in a report. For information on view filter effects on dynamic aggregation and other reporting features, see *View filter effects on reporting features, page 239*.

Copyright © 2024 All Rights Reserved

Efficient Functions for Partitioned Datasets

Copyright © 2024 All Rights Reserved

The table in this appendix lists the functions that work most efficiently and faster with partitioned datasets. Functions that are not listed work on an unpartitioned copy of the data to calculate results, which is less efficient and slower.

For additional information on any of the functions listed, such as how the functions are processed, see the Functions Reference.

Functions for partitioned datasets

Function type	Functions that perform efficiently in partitioned datasets
Basic functions	• Add • Average • Avg (average) • Count • GeoMean (geometric mean) • Greatest • Least • Max (maximum) • Min (minimum) • Multiply • Product • StDevP (standard deviation of a population) • StDev (standard deviation of a sample) • Sum • VarP (variance of a population) • Var (variance of a sample)
Date and time functions	All date and time functions

Copyright © 2024 All Rights Reserved

Function type	Functions that perform efficiently in partitioned datasets
Internal functions	• Banding • BandingC • BandingP • Case • CaseV
Null and Zero functions	• IsNotNull • IsNull • NullToZero • ZeroToNull
String functions	• Concat (concatenate) • ConcatBlank (concatenate plus blank space) • InitCap (initial capitalization) • LeftStr (left string selection) • Length (length of string) • Lower (lower case) • LTrim (left trim) • Position (position of substring) • RightStr (right string selection) • RTrim (right trim) • SubStr (substring selection) • Trim • Upper (upper case)
Arithmetic operators	All arithmetic functions
Comparison operators	• < • <=

Copyright © 2024 All Rights Reserved

Function type	Functions that perform efficiently in partitioned datasets
	• <> • = • > • >= • Begins With • Between • Contains • Ends With • In • Like • Not Begins With • Not Between • Not Contains • Not Ends With • Not In • Not Like
Comparison operators for rank	• *<= • *<> • *= • *>= • *Between • Not*Between
Logical operators	• AND • IF • Not

Copyright © 2024 All Rights Reserved

Function type	Functions that perform efficiently in partitioned datasets
	• Or
Data mining functions	All data mining functions
Financial functions	• Accrint (accrued interest) • Accrintm (accrued interest at maturity) • Coupdaybs (coupon period, beginning to settlement) • Coupdays (coupon period, number of days with settlement) • Coupdaysnc (coupon period, settlement to next coupon) • Coupncd (next date after settlement) • Coupnum (coupon, number payable between settlement and maturity) • Couppcd (coupon date, previous) • Cumipmt (cumulative interest paid) • Cumprinc (cumulative principal paid) • Db (fixed-declining balance (asset depreciation)) • Ddb (double-declining balance (asset depreciation)) • Disc (discount rate for a security) • Dollarde (dollar price, converted from fraction to decimal) • Dollarfr (dollar price, converted from decimal to fraction) • Duration • Effect (effective annual interest rate) • Fv (future value) • Intrate (interest rate) • Ipmt (interest payment) • Mduration (modified duration) • Nominal (nominal annual interest rate)

Copyright © 2024 All Rights Reserved

Function type	Functions that perform efficiently in partitioned datasets
	• Nper (number of investment periods) • Oddfprice (odd-first-period price) • Oddfyield (odd-first-period yield) • Oddlprice (odd-last-period price) • Oddlyield (odd-last-period yield) • Pmt (payment) • Ppmt (principal payment) • Price (price per $100 face value) • Pricedisc (price, discounted) • Pricemat (price at maturity) • PV (present value) • Rate (interest rate per period) • Received (amount received at maturity) • Sln (straight-line depreciation) • Syd (sum of year's digits depreciation) • Tbilleq (T-bill equity) • Tbillprice (T-bill price) • Tbillyield (T-bill yield) • Vdb (variable declining balance) • Yield • Yielddisc (yield on a discounted security) • Yieldmat (yield at maturity)
Mathematical functions	All mathematical functions can be used in partitioned datasets.
Statistical functions	• AvgDev (average deviation)

Copyright © 2024 All Rights Reserved

Function type	Functions that perform efficiently in partitioned datasets
	• BetaDistribution • BinomialDistribution • ChiSquareDistribution • Confidence (confidence interval) • Correlation • Covariance • CritBinomial (criterion binomial) • ExponentialDistribution • Fisher (fisher transformation) • FDistribution (f-probability distribution) • Forecast • ForecastV (forecast, vector input) • GammaDistribution • Growth • GrowthV (growth, vector input) • HypergeometricDistribution • Intercept • InverseBetaDistribution (inverse of the beta distribution) • InverseChiDistribution (inverse of chi-squared distribution) • InverseFisher (inverse of the Fisher transformation) • InverseFDistribution (inverse of F-probability distribution) • InverseGammaDistribution (inverse of gamma distribution) • InverseLognormalDistribution (inverse of lognormal distribution) • InverseNormDistribution (inverse of normal cumulative distribution)

Copyright © 2024 All Rights Reserved

Function type	Functions that perform efficiently in partitioned datasets
	• InverseTDistribution (inverse of T-distribution) • Kurtosis • LognormalDistribution • NegativeBinomialDistribution • NormalDistribution (normal cumulative distribution) • Pearson (Pearson product moment correlation coefficient) • Permut (permutation) • PoissonDistribution • RSquare (square of pearson product moment correlation coefficient) • Skew • Slope (of a linear regression) • Standardize • StandardNormalDistribution (standard normal cumulative distribution) • SteYX (standard error of estimates) • TDistribution • Trend • TrendV (trend, vector input) • WeibullDistribution

Copyright © 2024 All Rights Reserved

BEST PRACTICE FOR MICROSTRATEGY PRIME

Copyright © 2024 All Rights Reserved

MicroStrategy 10.0 introduced the next generation of in-memory analytics, also referred to as in-memory partitioning or PRIME (Parallel Relational In-Memory Engine). In-memory analytics enables you to create in-memory cubes using the Multi-Table Data Import feature.

Before MicroStrategy 10.0, the only way to create an intelligent cube (also known as an ROLAP cube or an OLAP cube) was from a report (which is a single table dataset). Starting in MicroStrategy 10.0, you have the option to use Data Import to create an in-memory dataset (sometimes referred to as an super cube) containing data from multiple tables. For easy reference, the following lists a comparison between Online Analytical Processing (OLAP) and super cubes.

	OLAP	Super
Version	Existed before MicroStrategy 10.0	New in MicroStrategy 10.0
Data Volume	In 9.x, limited to 2 billion rows	Each partition can have up to 2 billion rows
Schema	A single table dataset, pre-joined	Multiple tables, not pre-joined Closer to raw data staged in-memory; Multiple in-memory fact tables, including varying grains, many-to-many relationship tables, and entity-relation model semantics
Definition	User specifies the definition of the cube on a Report Template with Attributes, Metrics, and Filter	User specifies the tables to load into memory; Attributes/Metrics are mapped to these tables
Query Generation	Intelligence Server generates the SQL to execute against the data source (can use Multi-Source) By default, final pass of SQL joins lookup tables to fact table and/or metric temp tables to retrieve attribute descriptions	Queries submitted to the RDBMS are simple SELECT statements against the tables—no joins among tables

Copyright © 2024 All Rights Reserved

	OLAP	Super
Partition	Available	Available
Parallel	**Parallel Data Fetch Option** set at Intelligent Cube level	**Maximum Parallel Queries Per Report** set at Project level

In-memory cubes are perfect for use when creating visually-rich and interactive dashboard applications. Most in-memory analytics dashboard applications are built using the MicroStrategy Report Services Document interface. In-memory cubes can also be used as a dataset within Visual Insight dashboards.

Prerequisites

Administrators should verify that all environment settings are tuned to accommodate the cube to be published. The settings are dependent on data volume. For more information about the environment settings, see the following sections:

- *Hardware configuration settings*
- *Intelligence Server level settings*
- *Project configuration*

See the Tech Note on MSIFileTables for more information , (The information in the knowledge base article is applicable to Intelligence Server 10.x.)

Hardware configuration settings

Turn Off NUMA. For optimum performance with PRIME in-memory cubes, you must turn off NUMA.

Intelligence Server level settings

The Intelligence Server level settings are:

Copyright © 2024 All Rights Reserved

- *Number of connections by priority*
- *Query execution time*

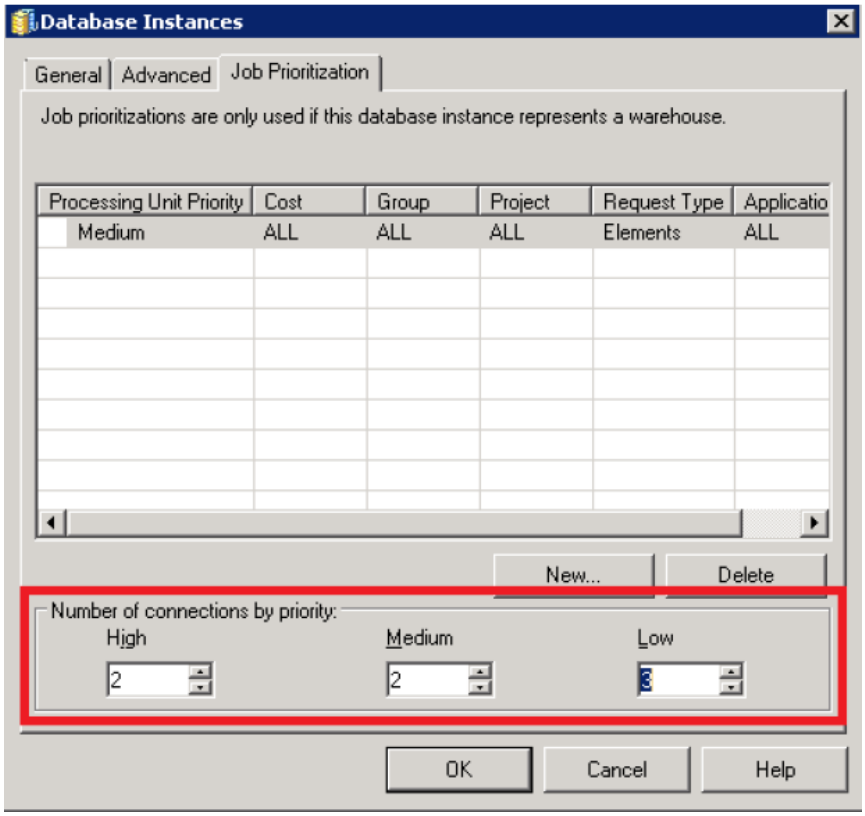
Number of connections by priority

All Data Import cube publishing jobs are currently initiated as low-priority jobs. The **Number of connections by priority** setting controls the number of processes that are available to in-parallel fetch data from external locations and read into memory. This setting specifies the total pool of processes available at the server level for the specific type of Data Import.

For OLAP and MTDI: Database, Hadoop, OLAP, Search index

1. In MicroStrategy Developer, expand **Administration**, expand **Configuration Managers**, and select **Database Instances**.
2. Right-click the database instance for which you want to define job prioritization and choose **Edit**.
3. In the Database Instance editor, select the **Job Prioritization** tab.
4. In the Number of connections by priority section, enter the number of connections in the **Low** field.

Copyright © 2024 All Rights Reserved



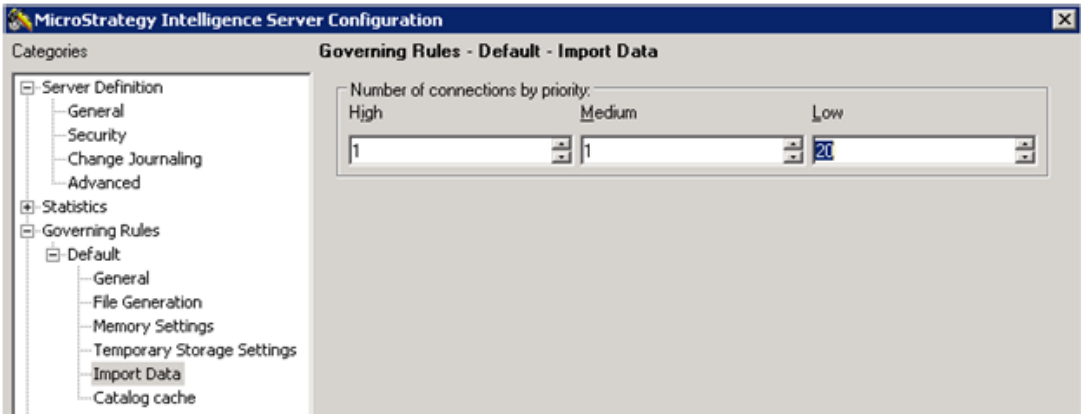
Click **Help** for more information about the Job Prioritization tab.

5. Click **OK**.

For MTDI: File from URL

1. In MicroStrategy Developer, from the **Administration** menu, select **Server**, and then select **Configure MicroStrategy Intelligence Sever**.
2. In the MicroStrategy Intelligence Server Configuration editor, expand **Governing Rules**, expand **Default**, and select **Import Data**.
3. In the Number of connections by priority section, enter the number of connections in the **Low** field.

Copyright © 2024 All Rights Reserved



4. Click **OK**.

Other external sources: Facebook, Google Analytics, Google Big Query, Drop Box, Google Drive, Salesforce, Twitter

Each pool of Data Import processes can have up to 20 threads.

Query execution time

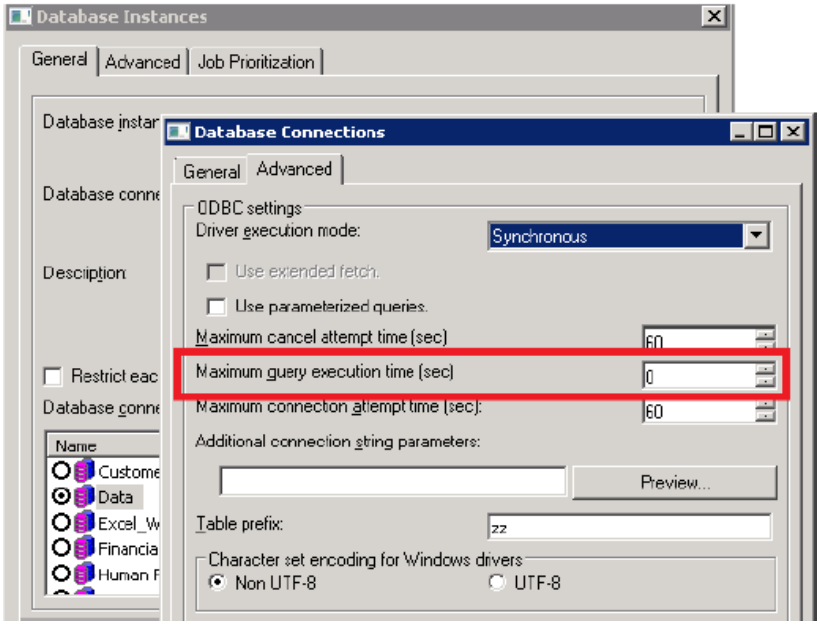
The Query execution time setting needs to be set appropriately to ensure that each process trying to fetch data in-parallel has enough time to fetch data completely. Values of 0 or -1 indicate infinite time (no limit).

For OLAP and MTDI: Database, Hadoop, OLAP, Search index

1. In MicroStrategy Developer, expand **Administration**, expand **Configuration Managers**, and select **Database Instances**.
2. Right-click the database instance for which you want to define job prioritization and choose **Edit**.
3. In the Database Instance editor, select the database connection and click **Modify...**
4. In the Database Connections dialog box, click the **Advanced** tab.

Copyright © 2024 All Rights Reserved

5. Set the **Maximum query execution time (sec)** field. This field defines the maximum amount of time a single pass of SQL can execute on the database.



6. Click **OK** to close the Database Connections dialog box.
7. Click **OK** to close the Database Instance editor.

For other sources

The Query execution timeout is internally set to 0 (infinite).

Project configuration

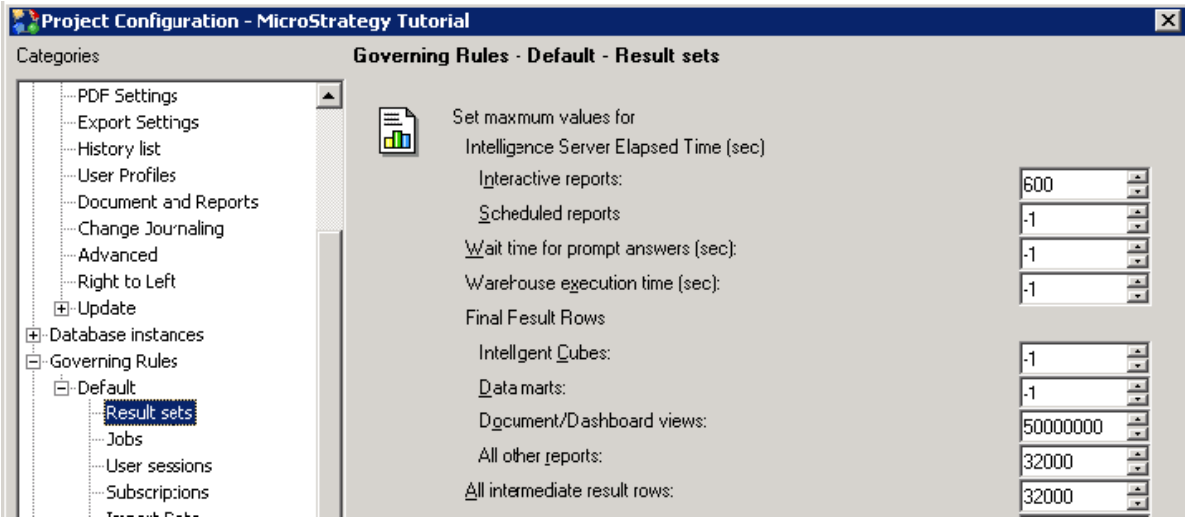
For project configuration, see

- *Result sets*
- *Data import specific*
- *User specific (if needed)*

Copyright © 2024 All Rights Reserved

Result sets

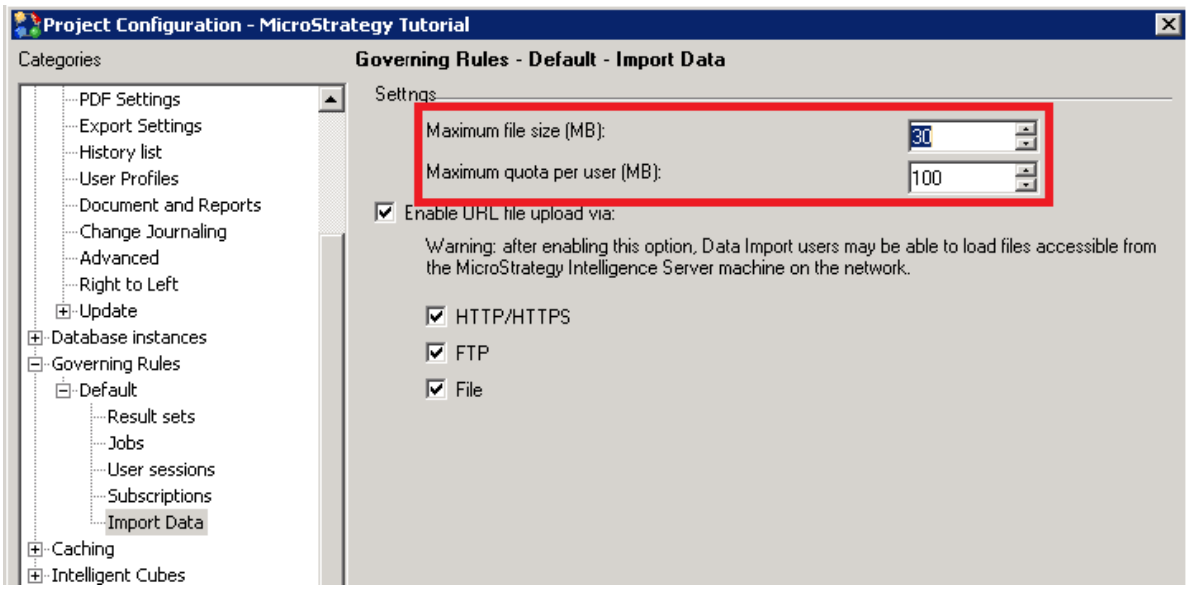
1. In MicroStrategy Developer, right-click on the project and choose **Project Configuration...**
2. In the Project Configuration editor, expand **Governing Rules**, expand **Default**, and choose **Result sets**.
3. Verify that the fields are set appropriately. The following figure shows a sample from a standard project.



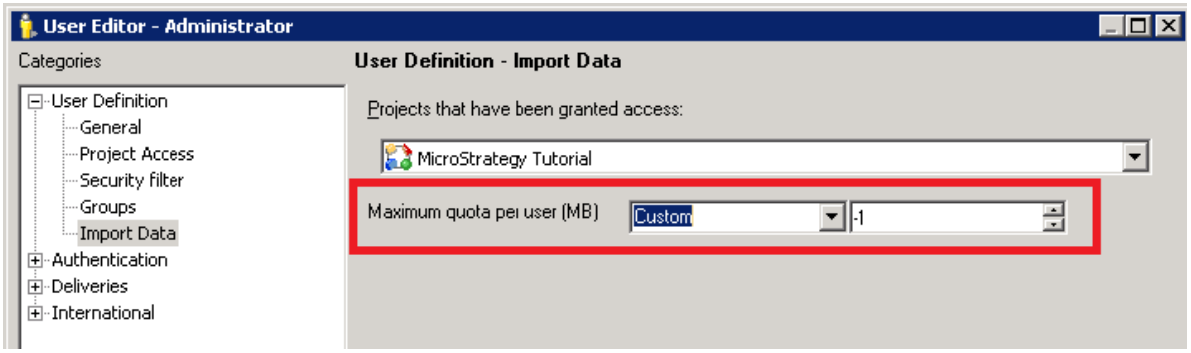
Data import specific

1. In MicroStrategy Developer, right-click on the project and choose **Project Configuration...**
2. In the Project Configuration editor, expand **Governing Rules**, expand **Default**, and choose **Import Data**.
3. Verify that the fields are set appropriately. The following figure shows a sample from a standard project.

Copyright © 2024 All Rights Reserved



User specific (if needed)



Parallel queries

You can improve the speed of a super cube publication with the **Maximum Parallel Queries Per Report** setting.

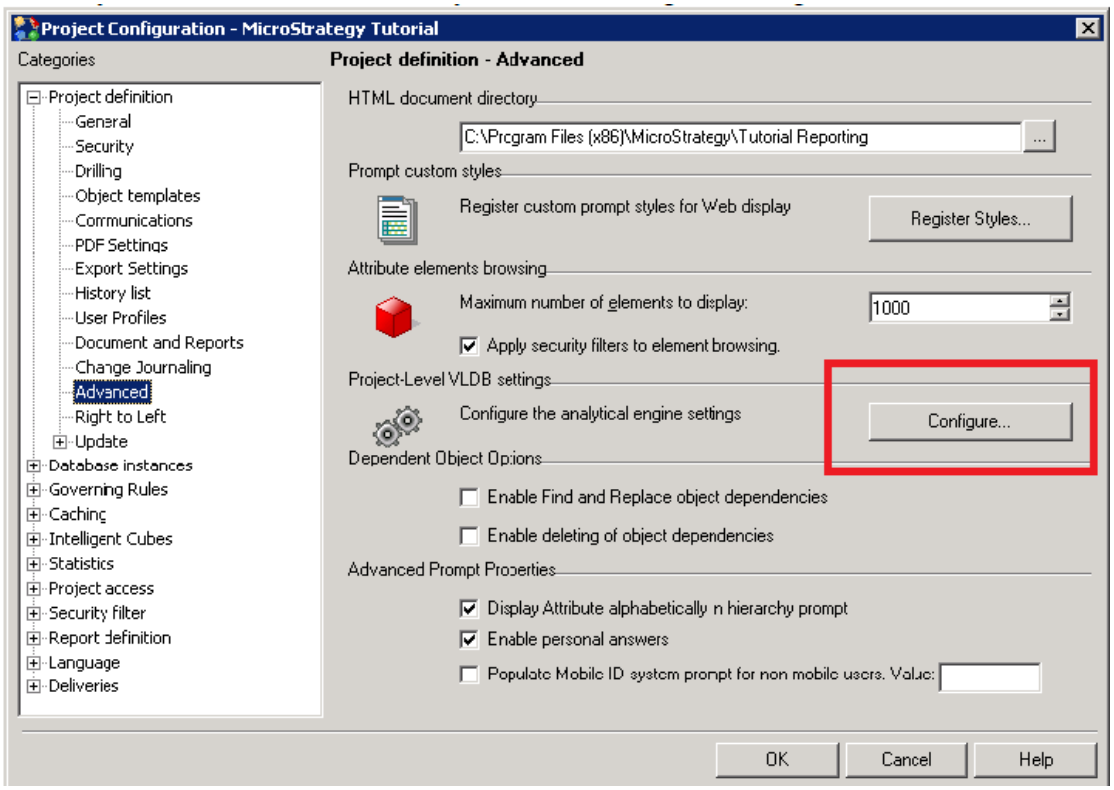
To configure parallel data loading

1. In MicroStrategy Developer, choose **Administration** > **Project Configuration...**.

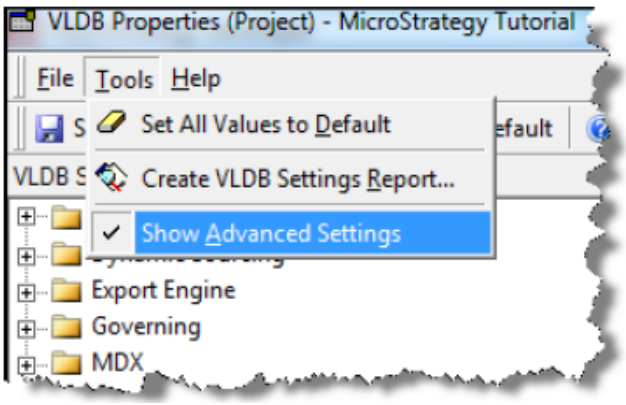
Copyright © 2024 All Rights Reserved

2. In the Project Configuration window, under **Project definition categories**, select **Advanced**.

3. In Project definition - Advanced, under **Project-Level VLDB** settings, click **Configure...**.

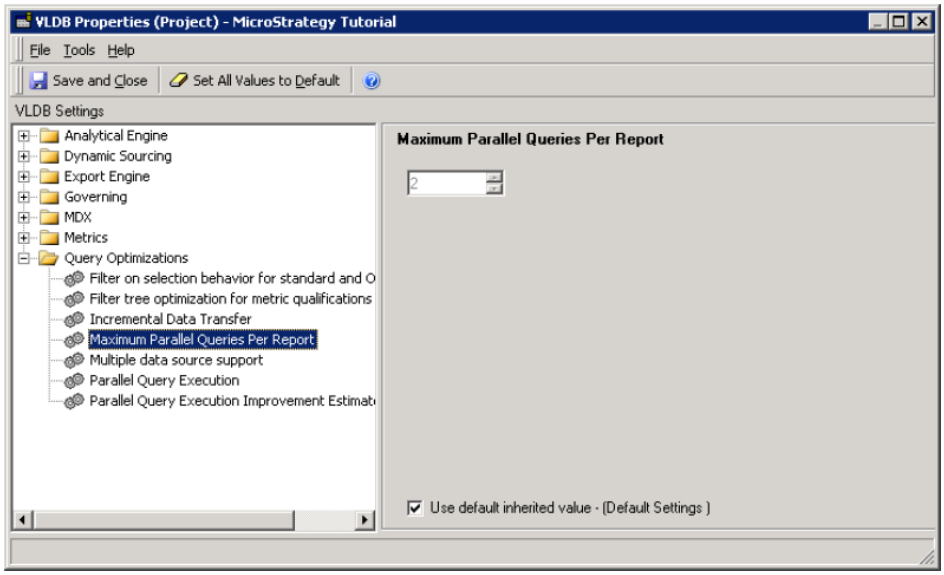


4. In the VLDB Properties window, choose **Tools** > **Show Advanced Settings**.



Copyright © 2024 All Rights Reserved

5. In the VLDB Properties window, under VLDB settings, expand **Query Optimizations**.
6. Under Query Optimizations, select **Maximum Parallel Queries Per Report**.
7. Clear the **Use default inherited value - (Default Setting)** check box.
8. In the **Maximum Parallel Queries Per Report** field, specify the maximum number of parallel queries per report.



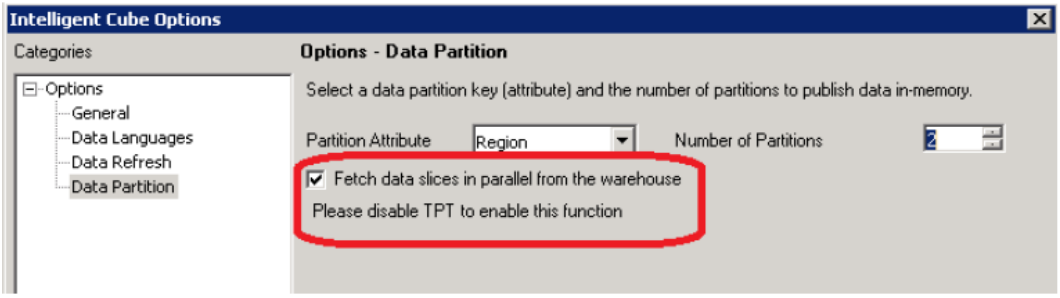
The Network Transfer Rate depends on the theoretical limit between the data source and Intelligence Server. However, if the data source is from a database, the Network Transfer Rate depends on the number of concurrent database threads that can be handled by the database. Each imported table is executed over a single thread. Therefore, to parallelize a big table with multi-table data imports, you may want to build multiple views representing slices to be fetched over an independent connection.

The parallel data fetch option

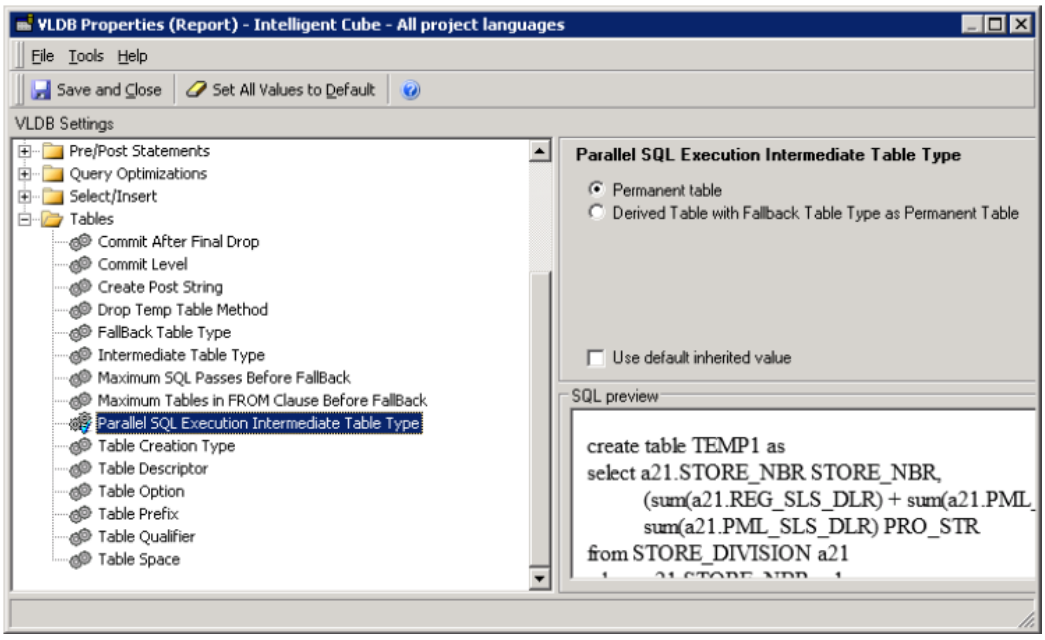
- Increases speed of Publishing for OLAP Cubes built using the MicroStrategy Developer.

Copyright © 2024 All Rights Reserved

- This is an add-on option and not to be confused with Parallel Query Execution option.
- Allows for SQL Select Pass for Metrics (typically the last pass) to be fetched over multiple ODBC connections.



- Users are allowed to switch between **Permanent Table** (for Generic) and **Derived Table** syntax (optimal for Single Select). See below.
- Number of maximum parallel queries for Parallel Data Fetch is governed by **Number of Partitions** setting in the screenshot above.



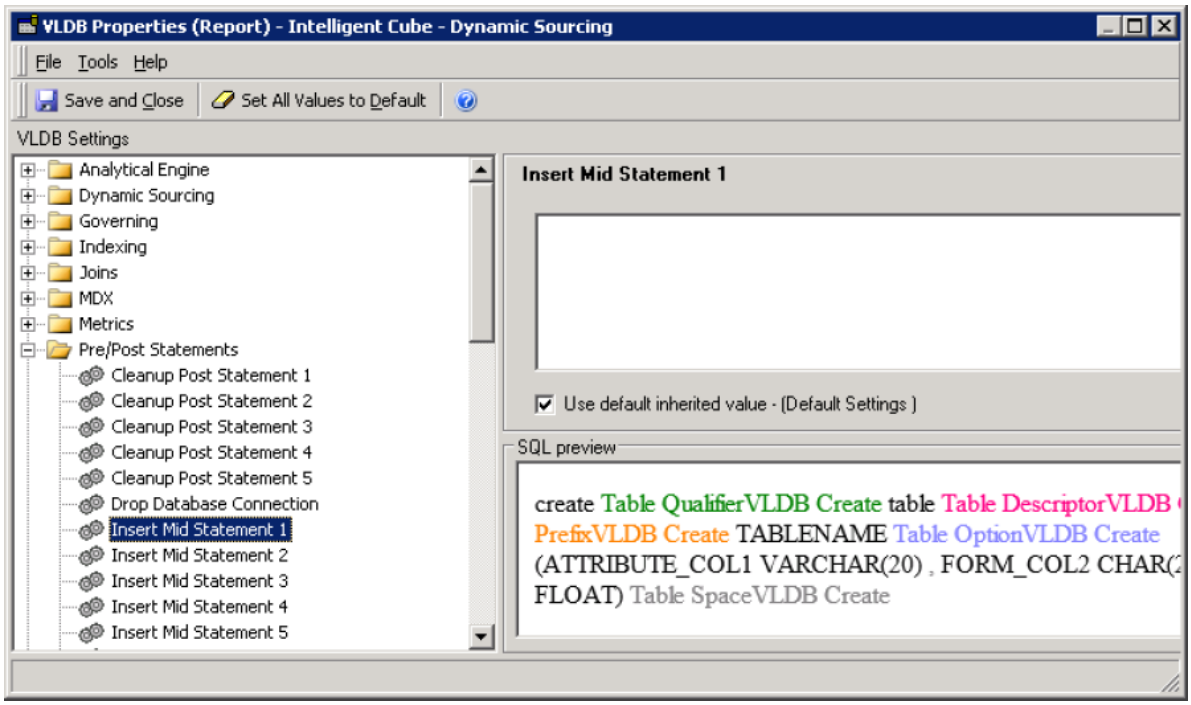
VLDB settings compatibility for OLAP cubes

For some of the VLDB settings, Parallel Data Fetch does not take effect, as described in the following sections:

Copyright © 2024 All Rights Reserved

- *Pre/post statements*
- *Query optimizations*
- *Partition attribute ID form type*

Pre/post statements



Parallel Data Fetch for OLAP Cubes does *not* work if **Insert Mid Statements** is set.

Query optimizations

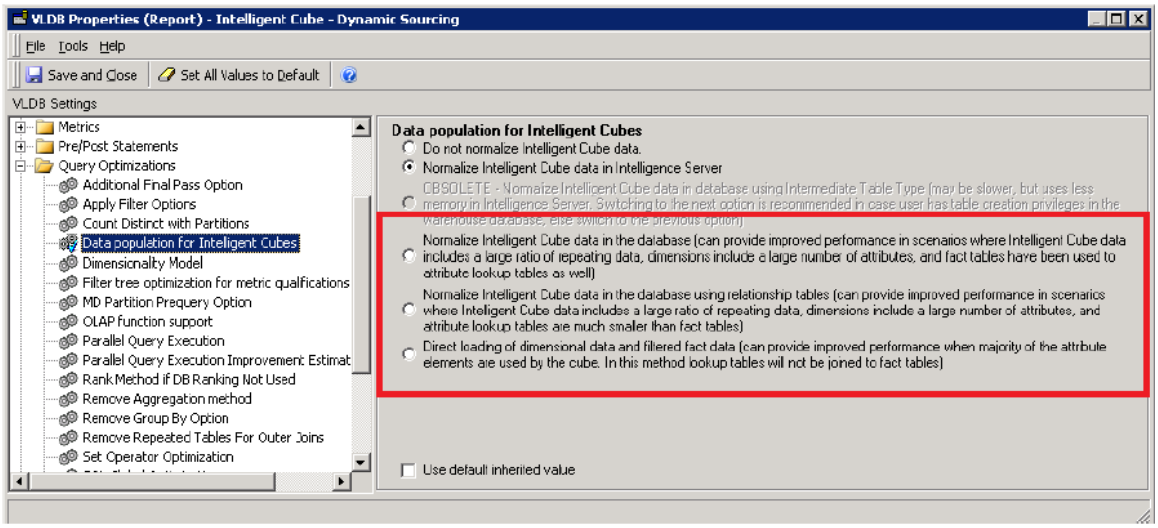
Parallel Data Fetch for OLAP Cubes does *not* work if any of the options below are chosen for the VLDB setting **Data population for Intelligent Cubes:**

- Normalize Intelligent Cube data in the database (can provide improved performance in scenarios where Intelligent Cube data includes a large ratio of repeating data, dimensions include a large number of attributes,

Copyright © 2024 All Rights Reserved

and fact tables have been used to attribute lookup tables as well)

- Normalize Intelligent Cube data in the database using relationship tables (can provide improved performance in scenarios where Intelligent Cube data includes a large ratio of repeating data, dimensions include a large number of attributes, and attribute lookp tables are much small than fact tables)
- Direct loading of dimensional data and filtered fact data (can provide improved performance when majority of the attribute elements are used by the cube. In this method, lookup tables will not be joined to fact tables)



Partition attribute ID form type

If ID form (or first form if ID form is compound) of the Partition attribute is String (or related) data type, Parallel Data Fetch for OLAP cubes does *not* take effect.

Partitioning cubes

Before MicroStrategy 10.0, all datasets were non-partitioned. Starting in MicroStrategy 10.0, partitioning is optional. If you are not partitioning the data, the published cube consists of one table.

Copyright © 2024 All Rights Reserved

A major advantage of MicroStrategy 10.0 cubes is the ability to partition cubes. While OLAP cubes were limited to 2 billion rows, PRIME OLAP cubes can be divided into partitions and each partition can contain up 2 billion rows. This advantage allows PRIME OLAP cubes to increase in capacity and scalability through partitions.

For more information, see the following sections:

- *Number of cube partitions*
- *Selection of partition key (distribution key)*
- *Notes about partitioning*
- *Where to define the partition*

Number of cube partitions

The number of partitions that can be defined for a single cube depends on the number of cores used by Intelligence Server. If Intelligence Server is restricted to a certain number of cores, through CPU affinity, then the PRIME OLAP cube is restricted by the limit. Also note that partitioning is performed on a single attribute.

Selection of partition key (distribution key)

- MicroStrategy PRIME currently supports only one partitioning key/attribute for the entire dataset. All tables that have the partition attribute will have their data distributed along the elements of that attribute.
- MicroStrategy PRIME supports the following data types for partitioning:
  - Numeric
  - STRING/TEXT
  - DATE

  All data is distributed using HASH schemes.
- The partition attribute is typically dictated by specific application needs. Below are some general guidelines for identifying a good partition

Copyright © 2024 All Rights Reserved

attribute.

- Partitioning is effective when each partition holds at least 250K rows.
- Some of the largest fact tables in the application are typically good candidates for partitioning and thus influence the choice of the partition attribute. They need to be partitioned to accommodate large data sizes and to take advantage of the PRIME parallel processing architecture.
- Data should be partitioned in such a way that it allows for the most number of partitions to be involved in any question that is asked of the application. Attributes that are frequently used for filtering or selections do not make for good partition attributes, as they tend to push the analysis towards specific sets of partitions thus minimizing the benefits of parallel processing.
- The partition attribute should allow for near uniform distribution of data across the partitions, so that the workload on each partition is evenly distributed.
- Columns on which some of the larger tables in the application are joined make for good partition attributes.
- Typically, the number of partitions should be equal to **half the number of logical cores** available to the PRIME server. This maximizes CPU usage to offer the best possible performance during cube publishing. Setting the number of partitions larger than the total number of CPUs will hinder performance.
- Each partition can hold a maximum of 2 billion rows. Define the number of partitions accordingly.
- The minimum number of partitions is dictated by the number of rows in the largest table divided by 2 billion, since each partition can hold up to 2 billion records. The maximum number of partitions is dictated by the number of cores on the box. The number of partitions should typically be

Copyright © 2024 All Rights Reserved

between the minimum and maximum, and closer to half the number of logical cores.

- In some cases, it is possible that a single column does not meet these criteria, in which case either the dataset/application is not a good fit for partitioning or a new column needs to be added to the largest table. Such an approach can generally be applied to partition only the single largest fact table in the dataset.

Notes about partitioning

- Partitioning limits the types of aggregations that can be quickly performed on the raw data. Functions that can be handled include distributive functions (such as SUM, MIN, MAX, COUNT, PRODUCT), or semi-distributive functions (such as STD DEV, VARIANCE) that can be re-written using distributive functions.
- Scalar functions (such as Add, Greatest Date/Time Functions, String manipulation functions) are supported.
- DISTINCT COUNTs on the partition attribute are supported.
- Derived metrics using any of the MicroStrategy 250+ functions are supported.
- For non-distributive functions, you may encounter high CPU and memory consumption.
- MicroStrategy 10.2 supports automatic in-memory partitioning for super cubes.

Where to define the partition

MicroStrategy 10.0 supports one partitioning key (attribute) for the entire dataset.

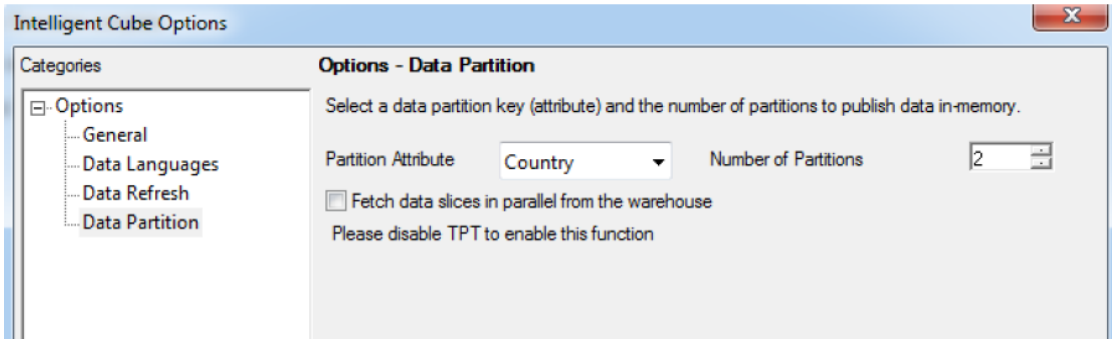
The minimum number of partitions is dictated by the number of rows in the largest table divided by 2 billion, since each partition can hold up to 2 billion

Copyright © 2024 All Rights Reserved

records. The maximum number of partitions is dictated by the number of cores used by Intelligence Server.

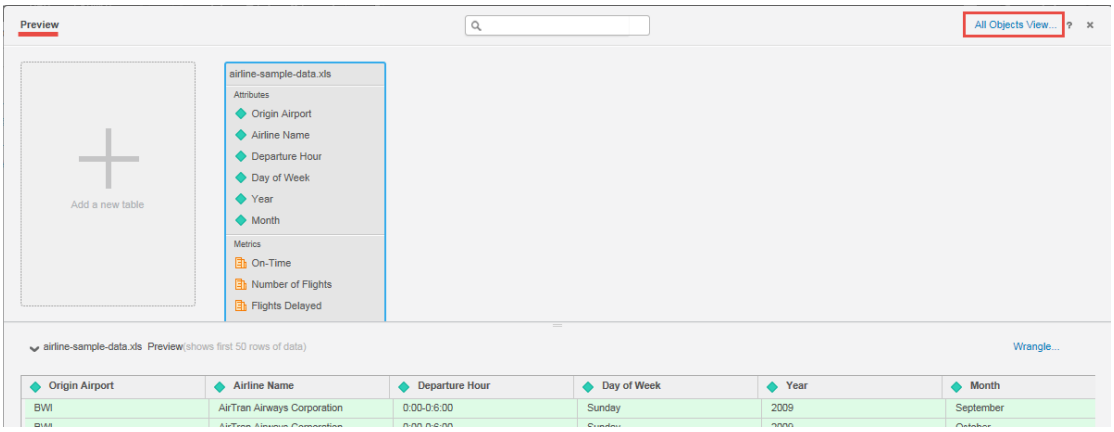
Defining the partition for OLAP Cubes

1. In MicroStrategy Developer, right-click on the intelligent cube and select **Edit**.
2. In the Intelligent Cube Editor, choose **Data** > **Configure Intelligent Cube...**
3. In the Intelligent Cube Options dialog box, expand **Options** and select **Data Partition**.



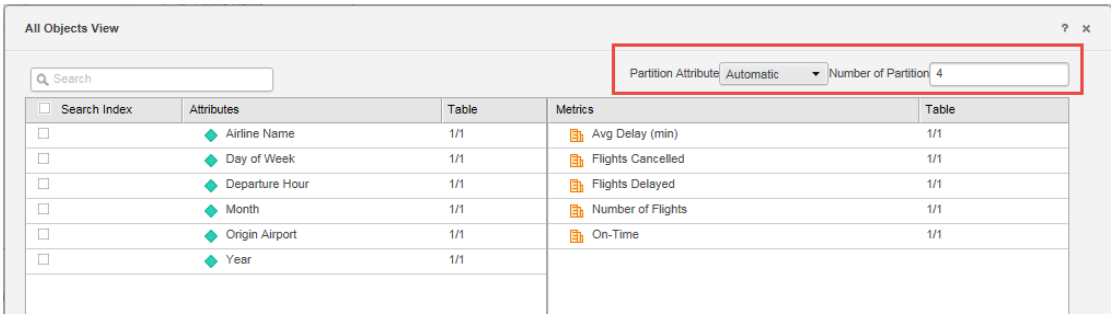
Defining the partition for Super Cubes

1. Select the data set and click **Prepare Data**.
2. Click **All Objects View**.



Copyright © 2024 All Rights Reserved

By default, the **Partition Attribute** pull-down menu is set to **Automatic**, which allows MicroStrategy Web to set the number of partitions.



By choosing **Automatic**, if the largest table is >= 1M rows, MicroStrategy will choose the attribute with highest cardinality attribute in the largest table and do the partitioning accordingly. Otherwise, there will be no partition for this data import cube.

To improve performance, MicroStrategy suggests modifying the default value (4) to be the half of the number of Intelligence Server CPU cores.

Sizing

Ensure that the Intelligence Server has the capacity to support all PRIME cubes in memory. Additionally, ensure that the server meets all the system requirements, and has enough capacity for both hard drive space and RAM.

As a general estimation of memory consumption, RAM can consume up to three times as much as the MSI Table size.

Use the following table as a reference.

Warehouse	MSI Table	
datatype	**datatype**	**Bytes per row**
FLOAT	FLOAT	4

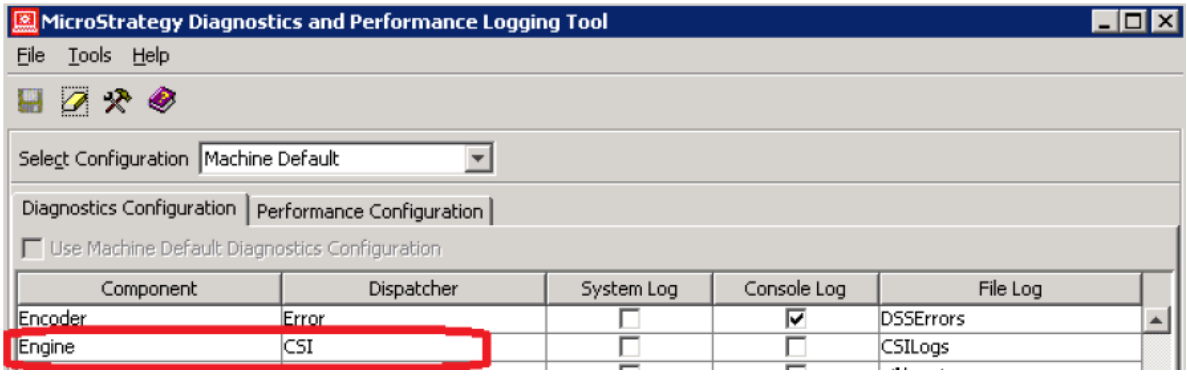
Copyright © 2024 All Rights Reserved

Warehouse	MSI Table	
INTEGER	INTEGER	4
SMALLINT	SHORT	2
DECIMAL (10,0)	DOUBLE	8
DECIMAL (21,3)	Big Decimal	21
DATE 'YYYY/MM/DD'	DATE(short-short-short)	6
CHAR / VARCHAR / NCHAR / NVARCHAR (m)	UTF8_STRING(m)	X*(m+1) If it is Linux, X = 1 or more
CHAR / VARCHAR / NCHAR / NVARCHAR (m)	STRING(m)	Y*(m+1) If it is Windows/AIX, Y = 2 or more

In the case where the cube has multiple tables, the MSIFile Table size of each cube can be combined to estimate the peak memory requirement.

In-memory partitioning generally results in more memory requirement as compared to no partitioning.

For a better understanding of the PRIME cube structure, enable Engine -> CSI logs before publishing cubes.



Copyright © 2024 All Rights Reserved

Cube incremental refresh

For OLAP cubes, MicroStrategy 10.0 functionality is the same as version 9.5.1 and earlier.

For super cubes, MicroStrategy 10.0 provides a flexible way to update all or some of the imported tables at different schedules.

In MicroStrategy Web, right-click the super cube, point to **Schedule**, and select **Show Advanced Options**.

More information about advanced options is available here.

Data source

All MicroStrategy Certified Data Sources are compatible with PRIME Cubes.

If sourcing from File, a multi-table data import is the better approach

Network Browser allows files to be loaded from the file systems that Intelligence Server can access.

Copyright © 2024 All Rights Reserved

Document / dashboard best practice

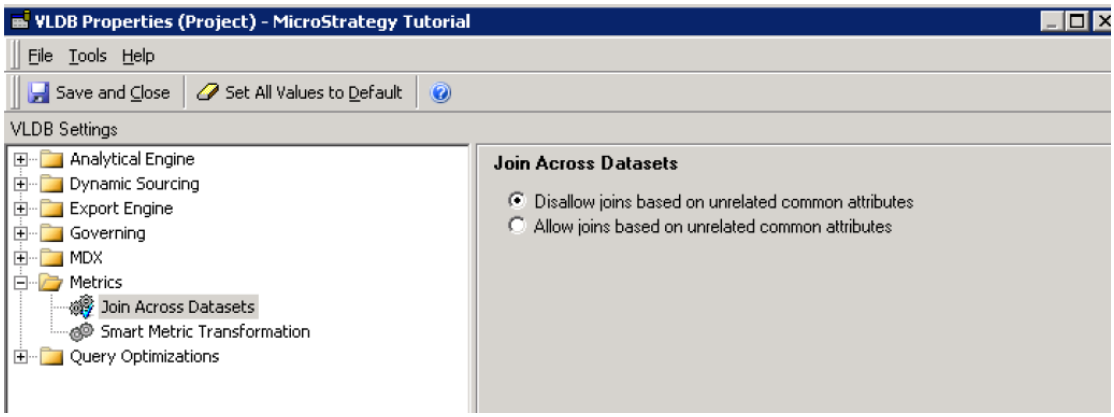
Document or dashboard reusability with PRIME cube

Dashboards have replaced dataset functionality. This allows better reconciliation and reusability of dashboard definitions when switching between an OLAP and super cube.

By default, joins on unrelated attributes across datasets differ between documents and dashboards:

Documents follow a Project level setting

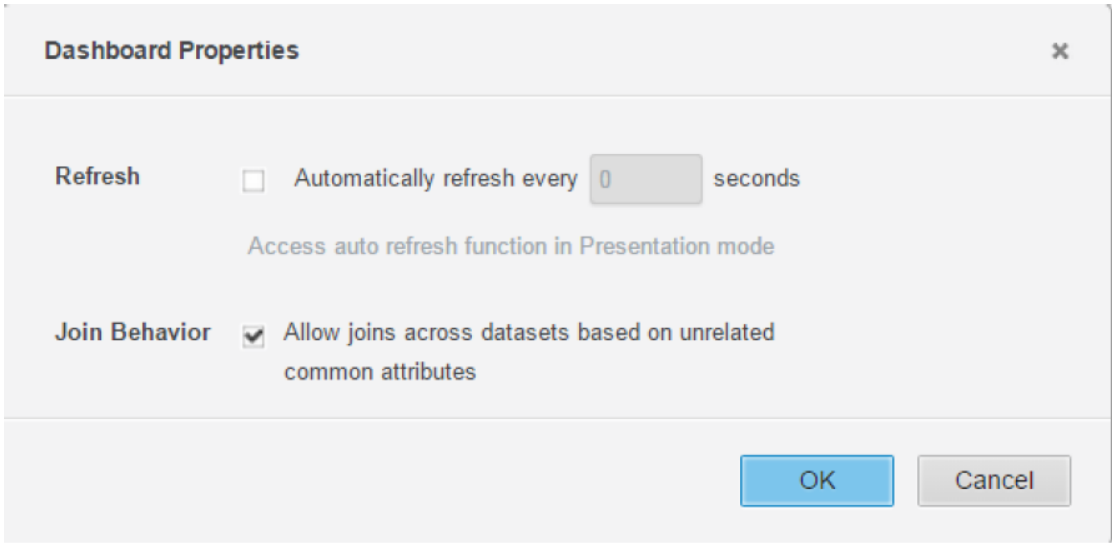
1. In MicroStrategy Developer, right-click on the project and select **Project Configuration...**
2. In the Project Configuration editor, expand **Project definition**, and choose **Advanced**.
3. Under Project-Level VLDB settings, click **Configure...**
4. In the VLDB Properties dialog box, expand **Metrics**, and choose **Join Across Datasets**.



Copyright © 2024 All Rights Reserved

Dashboards have individual settings.

1. In a MicroStrategy Web dashboard, from the **File** menu, choose **Dashboard Properties**.
2. Select/clear the **Join Behavior** option - **Allow joins across datasets based on unrelated common attributes**.

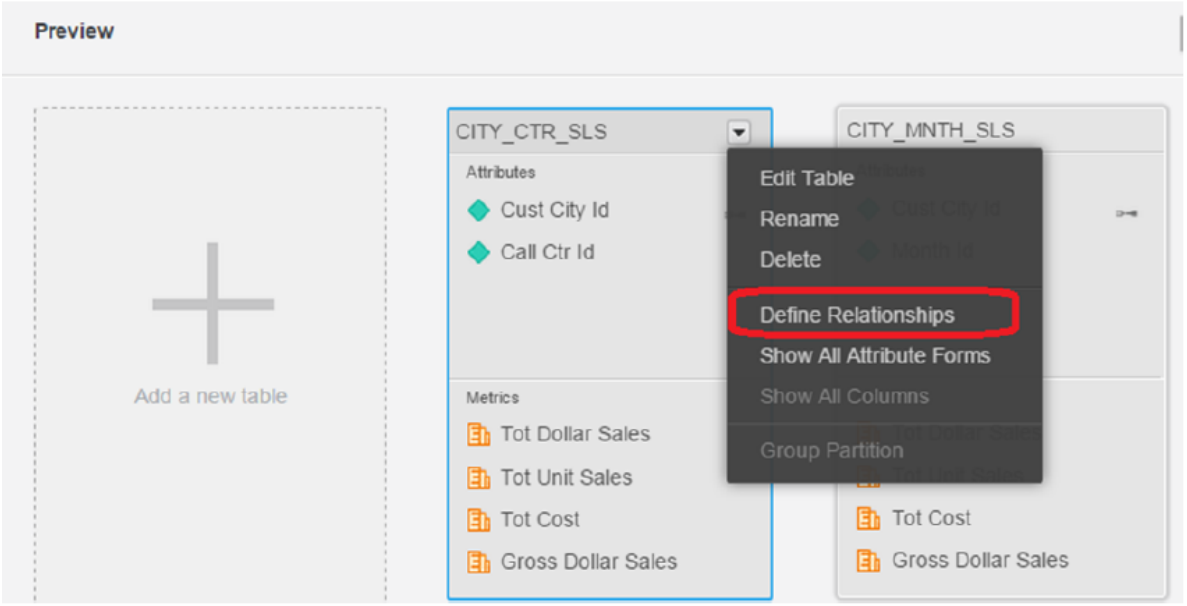


Relationships with Multi-Table Data Import

Generally, to increase the speed of aggregation and filtering, a one-to-many relationship is best.

In the MTDI workflow, auto relationship detection (using the first 1000 rows) can sometimes cause unnecessary overhead during cube publishing and introduce unwanted relationships. Be sure to check the relationships detected for each imported table to ensure that the relationships are not unwanted.

Copyright © 2024 All Rights Reserved

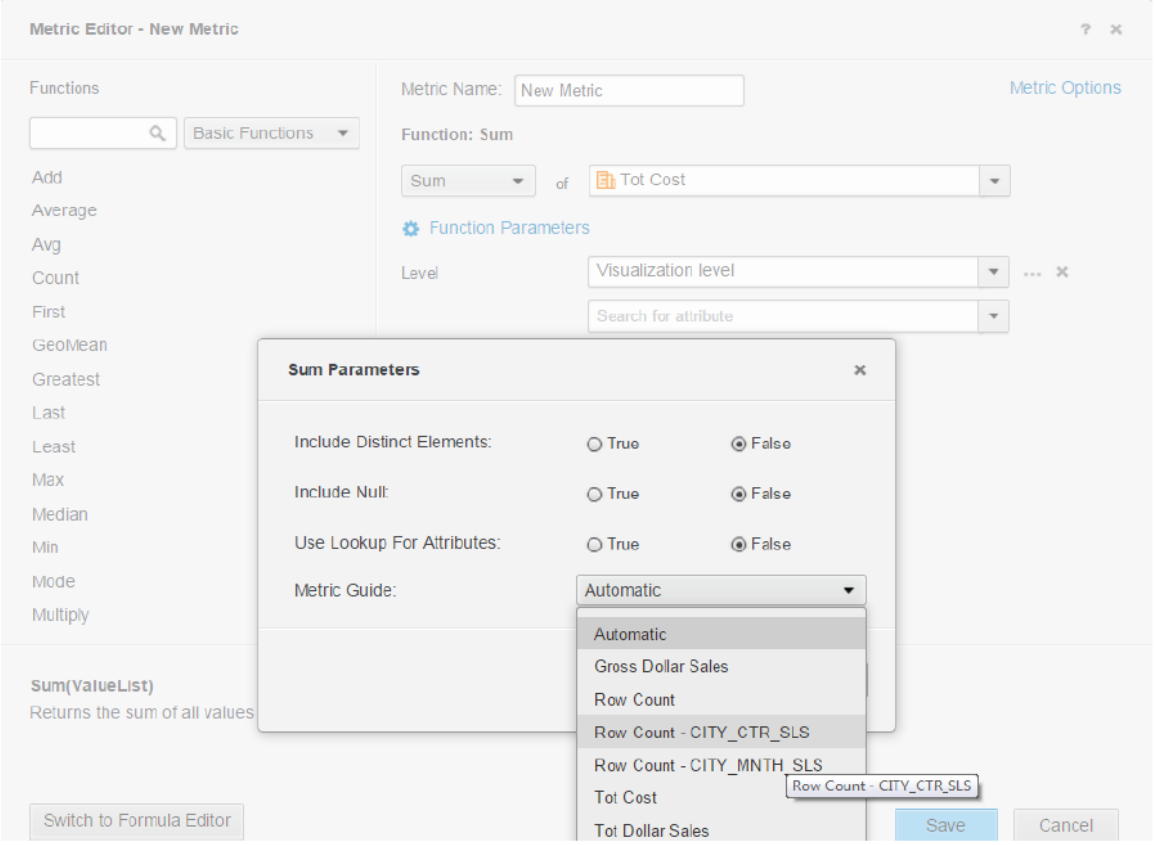


Relationships are global in nature. When defined on one table, they create a relationship table with composite information from all tables.

Metric guide

- Available under **Function Parameters** when Use Lookup for *Attribute* = *False*.
- Some metrics can be calculated in multiple ways. This setting allows the designer to suggest which uploaded table should be considered for calculating a certain metric. In special circumstances, it can also help resolve performance issues introduced by what may be considered in-efficient joins.

Copyright © 2024 All Rights Reserved



- **Row Count – *#Imported Table Name#*** is the best way for a designer to control which table a particular metric gets evaluated from.

Use PRIME cube as dataset vs view report as dataset

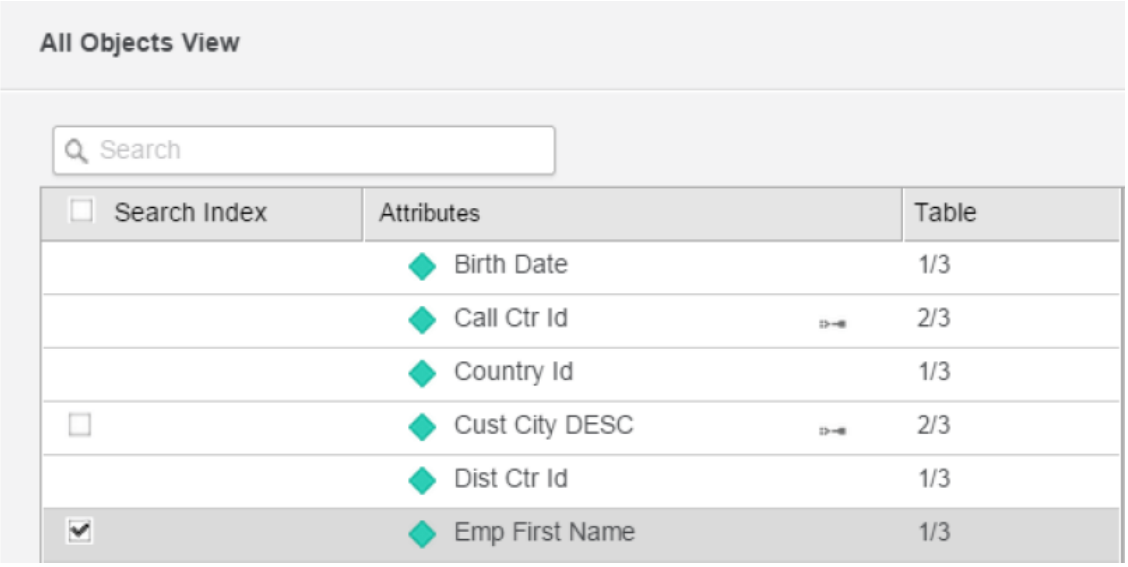
- Working Set/View/Normal Reports, used as datasets in documents, currently cannot support multi pass analytics for document grids.
- Benefit of using View reports currently is allowing customers to drag and drop more default derived metrics into **different** dashboards that can be built and saved in view reports. We are looking to reduce that gap for Cubes. Another benefit is to be able to use a much smaller dataset by filtering data, which can be personalized by using prompts in View reports.

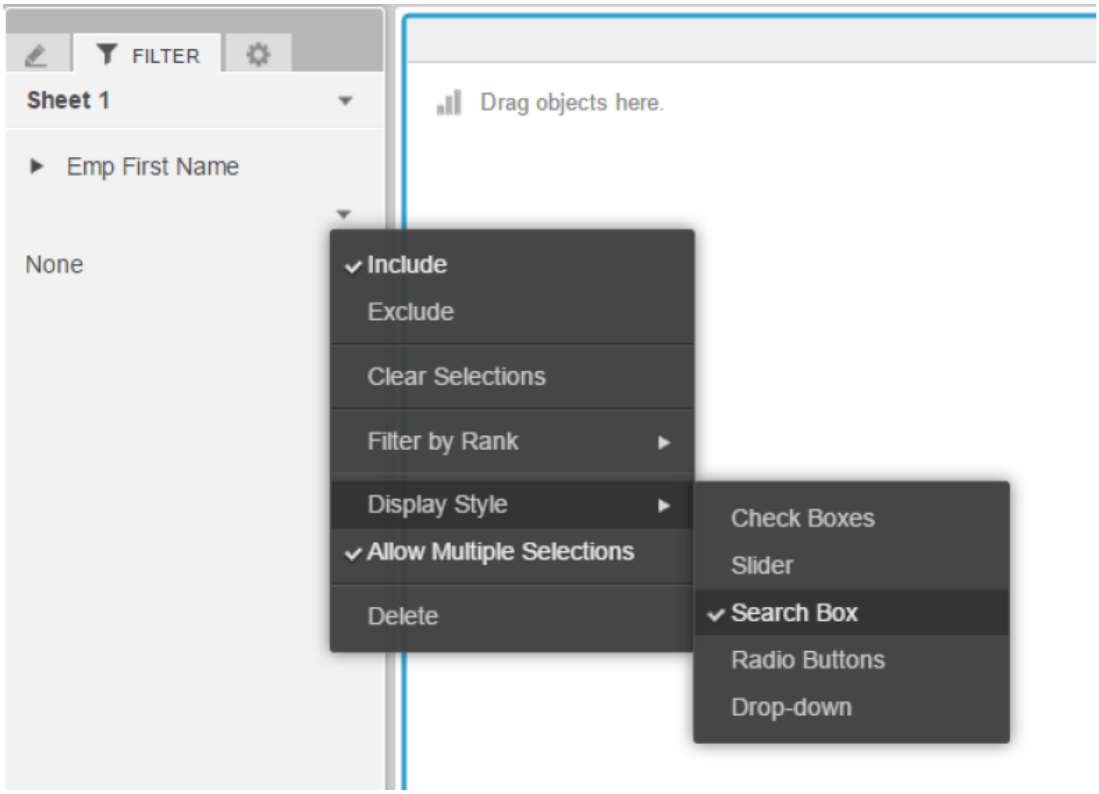
Derived metrics in View Reports would be aggregated two times: once from Cube to View Report and then from View Report to document grid.

Copyright © 2024 All Rights Reserved

Super-fast search-style selectors

For super cubes, consider using Search Index to speed up Search Box style search on text fields. The Selector as Search Box is available in both documents and dashboards.





Copyright © 2024 All Rights Reserved

Concurrency

Memory consumption / CPU utilization during PRIME cube publication concurrency

Expect peak memory usage to be up to three times the raw data size.

For Intelligence Server to stay responsive, a maximum of 50% of the logical CPU cores should normally engaged in populating a cube into memory after raw data is fetched.

Web or mobile access

Considerations for accessing dashboards/documents from Web or Mobile:

- No new suggestion for PRIME OLAP cubes. All considerations for MicroStrategy 9.4.1 cubes still apply.

Copyright © 2024 All Rights Reserved