This Software and Documentation are provided “as is” and without express or limited warranty of any kind by either MicroStrategy, Inc. or anyone who has been involved in the creation, production, or distribution of the Software or Documentation, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Software and Documentation is with you. Should the Software or Documentation prove defective, you (and not MicroStrategy, Inc. or anyone else who has been involved with the creation, production, or distribution of the Software or Documentation) assume the entire cost of any necessary servicing, repair, or correction. Some states do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

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

Copyright © 2000, MicroStrategy Incorporated. All rights reserved.

The information contained in this manual (the Documentation) and the Software are copyrighted and all rights are reserved by MicroStrategy, Inc. MicroStrategy, Inc. reserves the right to make periodic modifications to the Software or the Documentation without obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of the Software or Documentation without prior written consent of an authorized representative of MicroStrategy, Inc. are prohibited. U.S. Government Restricted Rights. It is acknowledged that the Software and Documentation were developed at private expense, that no part is public domain, and that the Software and Documentation are Commercial Computer Software provided with RESTRICTED RIGHTS under Federal Acquisition Regulations and agency supplements to them. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFAR 252.227-7013 et. seq. or subparagraphs (c)(1) and (2) of the Commercial Computer Software—Restricted Rights at FAR 52.227-19, as applicable. Contractor is MicroStrategy, Inc., 8000 Towers Crescent Drive, Vienna, Virginia 22182. Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.


09330800   Published August 2000
Contents

Project Designer Overview 13
What is the role of the Project Designer? 13
What is a project? 14
   Logical data modeling 14
   Physical data modeling 14
   Ideal skills and experience for the Project Designer 15
   Performance and implementation issues 15
   Maintenance and production 16
   Using this manual 16

Section I Project Designer Concepts 17

Chapter 1 Data Modeling 19
   Introduction to data modeling 19
   Learning the components of the logical data model 19
      Similarities between entity relationships and data modeling for MicroStrategy 20
   Structuring the data model 21
   Rules for data modeling 22
      Designing the Fact Table 23
   To learn about... 23

Chapter 2 Attributes 25
   What is an attribute? 25
   Types of attributes 27
      Simple 27
      Implicit Attributes 28
      Derived Attributes 28
   Elements and forms 29
      Attribute elements 29
      Attribute forms 31
      Form groups 32
      Form expressions 33
   Compound keys 34
   Attribute relationships 34
      One-to-one relationships 35
      One-to-many relationships 36
      Many-to-many relationships 37
      Joint-child relationships 37
   To learn about... 39
Chapter 3  Facts 41
What is a fact? 41
Types of facts 42
  Simple facts 42
  Implicit facts 42
  Derived facts 42
Defining facts 43
  Fact definition 44
  Column alias 46
Level extensions 46
  Table relation 47
  Fact relation 47
  Degradation 47
  Cross product 48
  Disallow 48
Heterogeneous fact column names 49
To learn about... 49

Chapter 4  Hierarchies 51
What is a hierarchy? 51
  Hierarchy structure 52
Types of hierarchies 53
  System hierarchy 53
  User hierarchies 54
Hierarchy display 54
  Locked hierarchy 55
  Limited hierarchy 55
  Filtered Hierarchy 55
  Entry point 56
Hierarchy browsing 56
  Drilling 58
To learn about... 59

Chapter 5  Partitions 61
What is partition mapping? 61
Types of partitioning 61
  Warehouse partition mapping 61
  Metadata partition mapping 62
To learn about... 63

Chapter 6  Projects 65
What is a project? 65
From warehouse to project 66
Project sources 66
  Connection Modes 67
Project source tools and projects 67
Project types 67
   MicroStrategy Desktop 68
   MicroStrategy Agent 6.x and DSS Agent 5.x 68
Duplicating projects 68
Warehouse Catalog 69
   Creating a database instance 69
To learn about... 69

Chapter 7 Transformations 71
What is a transformation? 71
Creating a transformation 71
   Member attributes 72
   Member attribute expression 72
   Transformation mapping type 72
To learn about... 73

Chapter 8 Putting It All Together 75
Development sequence 75
Inputs to modeling task 76
   Interviews 76
   Existing systems and reverse engineering 76
   Functional models 77
Approach to developing a data model 77
   Identifying entities 77
   Using patterns 78
Developing and using generic models 78
   Adapting generic models from other applications 78
   Developing a generic model 78
   When there is no generic model 79
   Using VMall as a sample project 79
Bottom-up modeling 79
Top-down modeling 80

Section II Project Designer Interfaces 81

Chapter 9 Attribute Editor 83
Attribute Editor layout 84
   Forms tab 84
   Children tab 89
   Parents tab 89
   Display tab 89
Section III  How Do I...?  147

Chapter 22  Attributes  149
Add an attribute form to an attribute  149
Create a column alias  150
Create a compound key for an attribute  151
Create a derived attribute  151
Create a form group  152
Create a joint child attribute  153
Create a new attribute  153
Create a new form category  154
Create an attribute form and expression  155
Create an attribute in the Attribute Creation Wizard  156
Define a column for SQL generation (attributes)  158
Define attribute children  158
Define attribute parents  159
Delete a compound attribute  159
Delete a form group  160
Manually map an expression to source tables  160
Modify an attribute  161
Modify an attribute form  162
Modify form display properties  162
Remove forms from the attribute browsing display  163
Remove forms from the attribute report display  163
Set attribute display properties  164
View attribute properties  165

Chapter 23  Facts  167
Add a fact expression to an existing fact  167
Change the mapping method for a fact expression  168
Create a fact extension based on a fact entry level  168
Create a fact in the Fact Creation Wizard  169
Create a formula for a fact expression  170
Create a fact and fact expression  171
Define a column for SQL generation (facts)  172
Delete a fact extension  173
Disallow a fact entry level  174
Extend a fact from its current fact to a new fact  175
Lower a fact entry level  176
Map a fact expression manually to source tables  177
Modify a column alias  178
Modify a fact expression  179
Modify a fact extension  179
Unmap an expression from a source table  180
View fact properties  181

Chapter 24  Hierarchies  183
Add a user hierarchy to the system hierarchy  183
Add attributes to a hierarchy  183
Add browsing attributes to a hierarchy  184
Add hierarchy attribute filters  184
Create a hierarchy  185
Define attribute relationships within a hierarchy  185
Delete a hierarchy  186
Remove attribute filters  187
Remove attributes from a hierarchy  187
Set hierarchy display properties  188
Set hierarchy for drilling  188
Set system hierarchy display in Data Explorer  189
Unlock hierarchy display  190
View data explorer in the desktop  190
View hierarchy properties  190

Chapter 25  Partitions  193
Add a data slice to a metadata partition  193
Create a metadata partition  194
Modify a data slice  194
Set partition mapping logical size  195

Chapter 26  Project Schema  197
Print the project schema  197

Chapter 27  Projects  199
Create a project  199
Create attributes with the Project Creation Assistant  199
Create facts with the Project Creation Assistant  201
Create a project with the Project Creation Assistant  202
Create a project source  203
Display hidden objects  203
Display the system hierarchy in the Data Explorer  204
Display the Data Explorer  204
Modify a project source  204
Remove a fact from a project  205
Remove a hierarchy from a project  205
Remove an attribute from a project  205
Select tables from the Warehouse Catalog using the Project Creation Wizard  206
Chapter 28  Tables 207
Set logical table size 207
View table structure 207

Chapter 29  Transformations 209
Add a new member attribute to a transformation 209
Create a member attribute 210
Create a transformation 210
Delete a transformation 211
Modify a transformation 211
Modify a transformation member attribute 212
Remove an attribute member 213

Chapter 30  Warehouse Catalog 215
Change the database instance for a project 215
Change the project warehouse 215
Change the project warehouse when the table name space is different 216
Edit the database instance for a project 216
Edit warehouse catalog tables 217
Set warehouse catalog browser options 217
Set the warehouse catalog read mode for a project 218
View a table definition in the warehouse catalog 218

Section IV  Appendixes 219

Appendix A  Pass-through expressions 221
  String Patterns 221
  Argument Types 221
  Upgrading 222
  Syntax 222

Appendix B  Attribute Cardinality 223
  Determining cardinality 223

Appendix C  SQL Engine Logic 227

Appendix D  Warehouse Catalog SQL 229
  Customizing Catalog SQL Statements 229
  Structure of Catalog Table SQL 230
  Structure of the full catalog SQL 231
  Default Warehouse Catalog SQL 231
Appendix E  Query Engine Logic 237

Appendix F  Drill Map Editor 239

Drill map associations  239
Drill path types  240
Custom drill paths  241

Drill Map Browser Interface  242
Drill Map Browser layout  243

Drill Map Editor Interface  244
Drill Map Editor layout  245

Drill Map Associations Dialog  248
Drill Map Associations dialog layout  248

Drill Map Browser: How do I...  249
Add a new drill map  249
Associate a new default drill map to the project  249
Create a new folder  249
Cut, copy, and paste a drill map  250
Delete a drill map  250
Edit a drill map  250
Log in to the Drill Map Browser  251
Rename a drill map  251
View a drill map’s properties  251

Drill Map Editor: How do I...  252
Access the Drill Map Associations dialog  252
Add a custom drill path  252
Add a hierarchy drill path  253
Add a map drill path  253
Cut, copy, and paste a drill path  254
Delete a drill path  254
Modify a drill path  254
Modify a drill path’s importance  255
Move a drill path up or down in the drill path list  255

Drill Map Associations Dialog: How do I...  256
Associate an object to a drill map  256
Resolve a conflict during object association  256

Section V  Index 259
Project Designer Overview

Topics covered in this overview include:
• What is the role of the Project Designer?
• What is a project?

What is the role of the Project Designer?

Your role as the Project Designer is to plan the data model for the rendering of a logical data model. The logical data model is the arrangement of information that makes sense to the end user, as opposed to the physical data model, which arranges information that makes sense from a database system standpoint. This approach prepares for the design of the physical data model and the eventual table structure in the database. You establish a data model for the project by making business requirements the cornerstone of your design. MicroStrategy products generate reports based on the business rules enforced by your design. You also research and organize data that becomes the foundation for the physical data model and the data warehouse.

As the Project Designer, you are responsible for the design, implementation, and creation of projects. You use a data model to determine how you build your projects. You work with the Administrator to determine who has access to the projects and the types of security access. You also work with the Report Designer and enhance the project as the needs of the Report Designer change. You are the contact for users who need changes made to the projects, and you work with the Administrator to make changes.

The data modeling process gets input from several sources: system users, business specialists, and strategic planners in addition to you, the data modeler. These inputs factor into the diagramming of business data in logical and physical data models. Information is tagged according to hierarchy and data type. It is also organized by relationships to optimize data retrieval once the design of the data model is translated into a working database system.
What is a project?

A project is the means through which you can access MicroStrategy Desktop functionality. A project is the highest-level intersection of a data warehouse, metadata repository, and user community. A typical project contains reports, filters, metrics, and functions. As the Project Designer, you create the projects that users access to run reports.

Projects are where you create and modify objects that the Report Designer can incorporate into future reports. When you create projects, you place each project within a project source. You can create a project source by using the Project Source Manager.

In MicroStrategy Desktop, a project can easily be created through the Project Creation Assistant. The assistant takes you through the creation of a full project from the name definition and database connections to the definition of attributes.

Logical data modeling

Developing a logical model is the first step in the data modeling effort. The logical model depicts business data in a visible structure easily understood by the end user. Similar to the entity relationship diagram but not as detailed in its table structure, the logical model lays the groundwork for the physical model.

Physical data modeling

The physical data model is based on the logical model, except where the physical model already exists and needs to be reverse-engineered to produce the logical model. The physical model, however, is a more detailed graphical explanation of business data. It puts the logical model’s organization in terms that make sense from a system perspective. Tables in the physical model contain highly organized information that interacts with system processes so that meaningful reports can be run and marketing trends can be discerned.

The logical and physical data models are merely different versions of each other. Both eventually contribute to the larger sense of the physical data model—the data warehouse. The logical model is only concerned with logical objects of the business model, such as Day, Item, Store, or Account. Several physical models can be derived from the same logical model based on how the data representing those logical objects is to be stored in the database, whether in the same table, separate tables, duplicated across several tables, or another table arrangement.
Ideal skills and experience for the Project Designer

The skills necessary to successfully fulfill the responsibilities of the Project Designer center on a thorough understanding of business requirements. You also need prior experience in data modeling and a grasp of common database concepts. These skills allow the Project Designer to plan effectively for the interaction between data and business rules.

Performance and implementation issues

Performance considerations figure prominently in the physical data modeling process and are provided for at implementation. You as the Project Designer and data modeler must recognize that your responsibility is to advocate adherence to business rules without respect to performance issues.

Performance issues are best judged against three points of a trade-off triangle: query performance, data warehouse maintenance, and user requirements. Heavy emphasis of one point takes away from the performance of the other two points.

Increasing throughput rates is the duty of the database designer. You and the physical database designer work in counter-balance to ensure that the model supports system design and user requirements, as well as the overall effectiveness of the system.
Maintenance and production

Once your data model is complete and data is loaded into the data warehouse, it is a good idea to periodically backup the metadata to guard against data corruption.

Despite the design of the data model, the desktop allows only one user to manipulate a schema object (attribute, fact, and the like) at a time. If more than one user attempts to modify the same object concurrently, an error message appears warning that the desktop cannot move or save an object changed in the metadata. The object becomes available to other users when the current session is finished.

Using this manual

This manual is a resource for every task you have to complete. This manual is divided into the following sections:

• **Concepts**. This section provides you with the key concepts about the objects you are working with.

• **Interface**. Refer to this section for interface-specific information. This section explains the interfaces you are using to create objects such as attributes, facts, hierarchies, partition mappings, and transformations.

• **How do I**. This is where procedural information is found. This section provides the answers to the How do I? questions you encounter while using MicroStrategy Desktop.
SECTION

Project Designer Concepts

• Data Modeling
• Attributes
• Facts
• Hierarchies
• Partitions
• Projects
• Transformations
• Putting It All Together
Data Modeling topics include:

- Introduction to data modeling
- Learning the components of the logical data model
- Structuring the data model

Introduction to data modeling

Data modeling is a design process. A data model graphically depicts the flow and structure of data in a business environment. It can also be defined as a logical diagram of an organization’s primary lines of business. When you create the data model, include only the requirements that must be incorporated into a MicroStrategy Desktop or any project. You can use the data model to design the logical schema.

You can also build the data model by deriving it from the physical database. This is customarily done when a system is already in place and needs to be reverse-engineered, usually to allow for new functionality. In most cases, the data model is created first, with logical and physical databases derived from it. Ultimately, the data model defines data relationships to the MicroStrategy 7 product. It maps the communication links between the logical and physical databases, where information can travel freely back and forth in support of queries generated through SQL. Its aim is to enable data to be compiled to build intelligence to understand business.

Learning the components of the logical data model

The logical data model is a conceptual representation of how data looks to the end user. A planning tool, it is the starting point that helps you visualize the flow of information in a business context. It also provides a structural reference for the Database Designer as he builds the physical schema.
The logical data model is composed of the following parts that categorize information:

- **Attributes** are logical objects that represent entities in the business model. Attributes must be distinct and act to group but not share elements. This allows users to define the level of aggregation at the attribute level. For example, City may be an attribute while New York City, San Francisco, and Boston are elements within that attribute. Attributes are objects you place in the Group By section of an SQL statement, or anything you want on a report. Attributes represent levels of aggregation and can be used for filtering and building reports.

- **Hierarchies** are groupings of attributes that are ordered to reflect their relationship with other attributes. The best design for a hierarchy is to organize or group attributes into logical business areas. For example, the attributes City, State, and Store are grouped to form the Geography hierarchy.

  Hierarchies must be separate and distinct from one another and include at least one attribute, but they can share attributes. Hierarchies are conceptual and not explicitly found in the data warehouse, though one can be inferred. Hierarchies provide a logical way of showing relationships between attributes.

- **Facts** are atomic level data components, as well as schema level objects, that represent calculations performed in the data warehouse to obtain some number. Facts can be thought of as business measurements, data, or variables that are typically numerical and suitable for aggregate functions in SQL. For example, Sales, Gross Profit, and Net Profit are facts that are used to build metrics.

For a more detailed explanation of these data model components, see the following topics:

- Attributes
- Hierarchies
- Facts

**Similarities between entity relationships and data modeling for MicroStrategy 7**

Data models and entity relationship diagrams (ERDs) share many similarities. Both define relationships between data items using common relationship terminology, graphically provide insight as to how a business is run, and track the flow of information through the business process.
They diverge, however, at the tabular level. As you examine the structural difference between data modeling and ERDs, notice that the ERD presents an easily identifiable table structure, while data modeling presents no visible table structure because it does not use tables in its portrayal of information. The ERD structure allows for easy recognition of its tables; whereas data modeling conceals its table structure beneath the names of its hierarchies. The data model is purely logical and is a conceptual representation of the business abstracted from the physical storage method.

Typically, ERDs feature tables against which reports are run. Data modeling, on the other hand, lays out the specific report topics that the system runs.

**Structuring the data model**

To structure the multi-dimensional data model, first determine the parts of the data model by analyzing your business requirements. You need only list the entities to be included in a project. These include attributes, facts, relationships, and hierarchies. Below is a list of questions to consider when examining data:

1. How much history is available from the operational systems?
2. What are the characteristics of all key columns in the source systems?
3. Is data audited on a frequent basis?
   - When audited, how many corrections are made?
4. How sparse is the data?
   - How much does the data compress as it is aggregated to various levels?
5. What are the attributes?
   - Using your database, make a list of all attributes you want included in the project.
   - Gather any related attributes into logical groups.
   - Determine the cardinality (number of elements) of each attribute in the system (high or low). A high cardinality can cause the system to run slow. For more information see the following appendix:

Appendix B: Cardinality Attributes

6. What are the facts?
   - Using your database, make a list of all facts (things that can be calculated, like sales and profit).
   - Determine the level of detail at which each fact is tracked.
7. Determine relationships between attributes.
   ◊ Specify the type of relationships between related attributes, for example, one-to-many or many-to-many.
   ◊ Specify which attributes are parents and which are children. These are dependent on the type of relationships between attributes. For instance, in a one-to-many relationship, one parent can have many children as in the case of one market having many stores. Hence, market is the parent and each of the many stores represents a child.

8. Determine hierarchies.
   ◊ Place attributes into hierarchically arranged groupings. In this case, hierarchies can be thought of as logical arrangements of business areas. Attributes flow from them.

The most important things to consider are the requirements and the types of reports that are needed. The final products are composed of attributes, attributes related to each other within hierarchies, and facts aggregated at particular attribute levels.

Make a list of all obvious attributes, facts and qualities and group them into logical hierarchies. Base the groupings on business requirements and data relationships. These are included later in the data warehouse.

Next, specify all attribute relationships within the same hierarchies. Attributes that belong in the same hierarchy can be arranged in one or more hierarchies such that all relationships are one-to-many, in a parent-child structure. A sign that two attributes need to be in separate hierarchies is the case of a many-to-many relationship between attributes that can only be resolved through the fact table.

You may find that as you identify attributes for the data model that relationships become apparent.

### Rules for data modeling

Keep the following rules in mind before you start the design of your data model:

- There is no limit to the number of facts or attributes in the data model.
- The data model must have at least one hierarchy.
- Each hierarchy must have at least one attribute.
- There is no limit to the number of elements in an attribute.
- An attribute associated with multiple hierarchies is referred to as a joint child.

The following diagram is an example of a simple data model:
Designing the Fact Table

Facts have two basic traits: They must be numeric and they must be calculable. Facts exist at the intersection of hierarchies and they store numerical data that can be aggregated. Ultimately, fact columns are used to define business metrics for reports. Using gathered requirements, identify facts and their respective hierarchies necessary to fulfill reporting requirements. For each fact, determine at what level of detail the fact is stored or calculated. Group each of the facts by the level of detail at which it is stored or calculated.

Use the logical grouping of facts to design the fact tables. Determine a numeric as opposed to a text ID for the fact table to increase system performance. Each table must have a primary key. Also, each fact table must have at least one attribute from a hierarchy if any analysis is to be done including that hierarchy.

To learn about...

...data warehouse-related tasks you can perform using MicroStrategy Desktop, please see the following topic in the How do I...? section:

Warehouse Catalog

...the Warehouse Catalog, please see the following topic in the Interfaces section:

Warehouse Catalog Editor
Attributes topics include:
• What is an attribute?
• Types of attributes
• Elements and forms
• Compound keys
• Attribute relationships

What is an attribute?

An attribute is a schema object included in a data model, for example, City, State, and Region. You can group related attributes such as these into a common hierarchy, like Location. When attributes are in the same hierarchy they must be related to each other, whereas attributes in different hierarchies cannot be related or share any elements.

You can use attributes to define the level of detail for a report. The lowest level attribute you include in a report is the lowest level of detail reported. A high level report, such as the Region level, includes the Region attribute, but it lacks the detail of a similar report which included the lower level attributes City and State.

Attributes are groups of related data, such as in table columns. The attribute acts like a column header, and the data that appears in the following table are elements. Elements define and make up the attribute. For example, the elements New York, Baltimore, and Boston are grouped under the attribute City.

The following diagram displays attributes and elements as they show in a table.
For SQL users:

In a SQL statement, you find attributes in the Select and Group By sections:

```
Select Store_ID, Date, sum(Sales)
From Store_Fact
Group By Store_ID, Date
```

Attributes are defined by these properties:

- **element**: unique value (a row) of an attribute
- **form**: identifier or descriptor of an attribute such as an abbreviation or URL
- **expression**: criterion defining what an attribute form does with columns, such as combining columns
- **relationship**: allows interaction of data and shows how data is related within a project

The following diagram displays how the attribute properties are related. Elements are not shown because they represent the actual data held in an attribute.
Types of attributes

Simply put, attributes act like holders of information. For example, a customer attribute holds information on the customer such as Name and Address, which are forms. The types of attributes are:

- Simple
- Implicit
- Derived

Simple attributes access data through columns you include when creating attributes. Implicit and derived attributes do not actually hold any data. These attributes create virtual data by combining or using columns to generate the data.

**Simple**

A simple attribute is made up of one or more expressions. With a simple attribute definition, you can define an attribute as a column, constant, or simple expression.
**Implicit Attributes**

An implicit attribute is a virtual or constant attribute that does not physically exist in the database because it is created at the application level. The implicit attribute has its expression, for example, criteria, defined as a constant value, though nothing is saved in a column. You create temporary columns in the database with a value of “1” for every row, which simplifies COUNT limitations. So, in the Attribute Editor, you enter only a “1” in the expression to create a count.

Implicit attributes are useful in analyzing and retrieving information. When analyzing data, you can use implicit attributes to create a COUNT to keep track of how many rows are returned. You can use implicit attributes when building metrics, where you can sum the column holding the constant to create a COUNT. Any constant is acceptable, for example, RushOrder=’Yes’. For more information on creating metrics, see the Report Designer guide.

**Derived Attributes**

A derived attribute has its value determined by an expression which combines two or more columns in a database to create a new column. For example, you can create a derived attribute to calculate age or anniversaries. By using two columns (date of birth and current date), you can place the difference between the two in a third column, to hold the age that has been derived from the two columns. Calculations and functions used in the expression can assist in deriving data from the database by producing SQL with database-specific syntax.

When creating derived attributes based on dates, you can use an **ApplySimple** pass-through function. By including this function in the expression, you can add database-specific syntax which allows SQL to go directly to the database. Because each relational database requires different syntax, you need to verify this with the requirements of the database you use. For more information on pass through functions, see the following Appendix:

**Appendix A: Pass-Through Expressions**
Elements and forms

Attributes are made up of elements and forms. Forms define the attributes, and elements represent the content.

Attribute elements

Attribute elements are the unique values or contents of an attribute. For example, if City is the attribute, then Chicago and Miami are elements of City. This is because they are both instances of the attribute City. You can have multiple elements for an attribute, but you must have at least one.

Elements must be considered when determining relationships between attributes. The elements are not included in the data model because they are the physical data in the database. You can use elements to build filters in reports which return precise data. By knowing and understanding the elements of an attribute, you can better design your data model and project. As shown in the following data model, the attribute Division has multiple attribute elements, for example Men’s Clothing, Shoes, and Sporting Goods.
Product hierarchy attributes and elements

Another example displays the elements and data for the Store attribute. Each attribute element is a row in an attribute lookup table as shown in the following example:
Elements and forms

Elements are typically referred to by their most descriptive form. For example, when you include “Atlanta” in a report, you are referring to the element that corresponds to the Store_Name, which is “Atlanta.”

Attribute forms

Attribute forms are identifiers or descriptors of an attribute. Just as an element is a distinct occurrence of an attribute, a form defines the attribute. For example, if Chicago is an element of the attribute City, then a form for the Chicago store is a URL, such as www.chicago.com, or an abbreviation such as CH.

Each attribute form provides details which identify and describe an attribute. The Store_ID is a unique numeric identifier for each store, while Store_Name holds the actual store name. Attribute forms for the Store attribute can include ID, numbers, descriptive names, short abbreviated names, URLs, and so on. In MicroStrategy 7.x, you can assign a maximum of 32 forms per attribute.

A simple lookup table with three columns holds separate forms, Store_ID, Store_Name, and Store_Long_Name. The attribute forms represent the columns in the table below:

```
<table>
<thead>
<tr>
<th>Store_ID</th>
<th>Store_Name</th>
<th>Store_Long_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Atlanta</td>
<td>Atlanta, Georgia</td>
</tr>
<tr>
<td>2</td>
<td>Miami</td>
<td>Miami, Florida</td>
</tr>
<tr>
<td>3</td>
<td>Boston</td>
<td>Boston, Massachusetts</td>
</tr>
<tr>
<td>4</td>
<td>New York</td>
<td>New York, New York</td>
</tr>
<tr>
<td>5</td>
<td>Albany</td>
<td>Albany, New York</td>
</tr>
</tbody>
</table>
```

Lookup_Store and attribute form data

Elements are typically referred to by their most descriptive form. For example, when you include “Atlanta” in a report, you are referring to the element that corresponds to the Store_Name, which is “Atlanta.”

Attribute forms

Attribute forms are identifiers or descriptors of an attribute. Just as an element is a distinct occurrence of an attribute, a form defines the attribute. For example, if Chicago is an element of the attribute City, then a form for the Chicago store is a URL, such as www.chicago.com, or an abbreviation such as CH.

Each attribute form provides details which identify and describe an attribute. The Store_ID is a unique numeric identifier for each store, while Store_Name holds the actual store name. Attribute forms for the Store attribute can include ID, numbers, descriptive names, short abbreviated names, URLs, and so on. In MicroStrategy 7.x, you can assign a maximum of 32 forms per attribute.

A simple lookup table with three columns holds separate forms, Store_ID, Store_Name, and Store_Long_Name. The attribute forms represent the columns in the table below:

```
<table>
<thead>
<tr>
<th>Store_ID</th>
<th>Store_Name</th>
<th>Store_Long_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Atlanta</td>
<td>Atlanta, Georgia</td>
</tr>
<tr>
<td>2</td>
<td>Miami</td>
<td>Miami, Florida</td>
</tr>
<tr>
<td>3</td>
<td>Boston</td>
<td>Boston, Massachusetts</td>
</tr>
<tr>
<td>4</td>
<td>New York</td>
<td>New York, New York</td>
</tr>
<tr>
<td>5</td>
<td>Albany</td>
<td>Albany, New York</td>
</tr>
</tbody>
</table>
```
In this example, the Lookup_Store table records all of the attribute form data for the Store attribute.

Attributes must contain at least one form and can have an unlimited number of forms from different tables. The forms you create must have a reference to a Lookup Table and can include multiple expressions. You can choose a lookup table in the Attribute Editor from a list of tables existing in the project.

**Attribute form properties**

When you create forms in the Attribute Editor, you must select properties for each form. These properties affect the display of the forms.

- **Form categories**-this property helps to categorize the types of forms. The standard options are ID, Desc, and None. You can create new form categories in the Attribute Editor.

- **Format type**-this property controls how the form is displayed. For example, you can choose Date for dates or Picture for displaying images correctly.

- **Default sort**-this property controls how the form is sorted when included in a report. You can choose from Ascending, Descending, or None.

**Form groups**

A form group is a grouping of attribute forms that have something in common. You can create form groups to join forms that you want related. By grouping forms, you can create a uniquely defined attribute which groups two or more columns under a single attribute. When you create a form group, the included forms are joined together and act as one. See the example below of the Customer form group.

<table>
<thead>
<tr>
<th>State</th>
<th>Last_Name</th>
<th>First_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VA</td>
<td>Brown</td>
<td>Frank</td>
</tr>
<tr>
<td>DC</td>
<td>Jameson</td>
<td>Greg</td>
</tr>
<tr>
<td>MD</td>
<td>Clifford</td>
<td>Jim</td>
</tr>
</tbody>
</table>

Customer attribute

This form group joins the forms State, Last_Name, and First_Name to identify the attribute Customer.
When you create form groups, all form categories must be the same, set to one ID and/or one DESC. When forms are grouped they are being joined, and the categories must be the same. The Format types of the ID and DESC can be different.

Form expressions

An expression defines what a form does with the attribute columns, such as adding or subtracting. Each attribute form must have one or more expressions declared. Though you can have multiple expressions, a form cannot have two different expressions in the same source table. A source table can only be used once in an attribute form.

You can create expressions using attribute columns and/or mathematical operators, for example, +, -, /, *. Only implicit attributes do not include a column in the expression, since they only use the constants you declare. For more information on creating form expressions, see the following How do I...? topic:

Attributes

A VMall solution: The Store attribute

This is an example using the attribute Store and its corresponding elements and forms:
**Compound keys**

A compound key attribute is an attribute whose primary key is made up by the combination of two or more columns. The multiple columns are joined, typically in a many-to-many relationship, with other attributes.

When creating new attributes in the Attribute Editor, you create forms and expressions. To create a compound key, you create multiple attribute forms, group them together, and set the form group as the key for this attribute. In the Attribute Creation Wizard, when you pick more than one column during the ID selection step, MicroStrategy Desktop will create a form group for you and set it as the key. This is made simple because MicroStrategy Desktop assumes form groups exist when you create compound keys. For more information on creating compound keys see the following How Do I...? topic:

Attributes

**Attribute relationships**

Attribute relationships relate and define the relational data model. Without relationships there is no interaction of data, and therefore no structure. The attribute relationships give meaning to the data by describing how data is related within a project. Through relationships, attributes can act as either child or parent in a relationship. The following types of relationships are explained below:

- one-to-one (1:1)
- one-to-many (1:M)
- many-to-many (M:M)
- joint-child relationships

Attributes relate to each other based on business rules and models you create. For example, the attributes Store, Region, and Country relate to one another in the following ways:

- One store has only one address, and one address can only belong to one store. This is a 1:1 relationship.
- One region can have more than one store, but each store can only have one region. This is a 1:M relationship.
- Countries can have many time zones, and a time zone can cross many countries. This is a M:M relationship.
An attribute relationship has two parts: parent and child. A child must always have a parent, but a parent can have multiple children. The parent attribute is in a higher level in the hierarchy than the child, and groups together the child attributes.

Parents and children

In the example above, you see two parents (Class, Color) and one child (Item). Class and Color can each have one or more Items, but each Item can only have one Class and one Color.

**Rule:** Attributes cannot be both child and parent in a relationship with the same attribute. But, an attribute can be the parent in one relationship and a child in a different relationship.

One-to-one relationships

A one-to-one relationship exists when there is only one parent element to every child element. The parent can only have one child, and the child can only have one parent.

A VMall solution: Store and Region

In the case of the attributes SKU and Item, there can only be one SKU per Item, and each Item can only have one SKU.

The diagram below shows the relationship between the SKU and Item attributes. The line descending from Item to SKU denotes that Item is the parent of SKU; the straight non-forked connection denotes the “one” ends of the relationship:
One-to-many relationships

The most common type of relationship between attributes is the one-to-many relationship. A one-to-many relationship is a type of parent-child relationship in which each child has only one parent, but each parent can have multiple children. One attribute is a child of another attribute if every element of the child has exactly one related element in the parent.

A VMall solution: Country and Region

The relationship between Country and Region is an example of a one-to-many attribute relationship. Each region has one country. It is not possible for a Region to be in more than one country, but each Country can have more than one Region.

The diagram below shows the relationship between the Country and Region attributes. The line descending from Country to Region denotes that Country is the parent of Region; the forked connection denotes the “many” end of the relationship:
Many-to-many relationships

A many-to-many relationship exists when there are multiple parent elements to multiple child elements. Each parent can have multiple children and each child can have multiple parents.

A VMall solution: Catalog and Item

An example of a many-to-many relationship is that of attributes Catalog and Item, where each Item can appear in multiple Catalogs and each Catalog can contain multiple Items.

The following diagram shows the relationship between the Catalog and Item attributes. A Catalog can have many Items, and Items can be included in many different Catalogs. The line descending from Catalog to Item denotes that Catalog is the parent of Item; the forked ends denote the “many” ends of the relationship:

Joint-child relationships

Joint-child relationships are non-aggregatable and have a special type of child relationship. A joint-child has a set of child attributes that are together considered a single child of the attribute. A joint-child relationship exists at the intersection of independent attributes and behaves much like an attribute.

A joint-child relationship is a combination of child attributes. Joint-child relationships have ID columns which exist in a base table. Examples of joint-child relationships are flags such as promotion (a function of item, date, and store), loan status (a function of customer and date), or temperature (a function of time and location).

Consider the attribute relationships between weather, time, and location. The first observation might be that weather happens at a discrete time and location. The weather in one location in the morning might be different than the weather in the same location in the evening. One location might have sunshine at a certain time while a location close by has rain and sleet.
Weather must have a time and location to exist. Weather cannot simply exist in a location, because it varies with time and season. And it cannot exist at just one time because it varies with location. Essentially, weather does not independently relate to time or location, but to a combination of these attributes.

Weather occurs in different combinations all the time. At any one time, there are many types of weather happening all around the world. Therefore, there is a many-to-many relationship between attributes Weather and Time. Since weather happens all over the world, there are no places on Earth in which “no weather” occurs. Furthermore, all places have more than one type of weather, at least for the purpose of this example. Therefore, attributes Weather and Location have a many-to-many relationship.

Some sort of relationship exists between the combination of these attributes (Time and Location) and Weather. In fact, every time/location combination has one, and only one, type of weather.

**Note:** An attribute with multiple children is not automatically a joint-child. Only an attribute that is the parent of a joint set of attributes has a joint-child.

---

**A VMall solution: Weather**

In this example of a joint-child relationship, there is a one-to-one relationship between Time and Location. There is also a one-to-many relationship between Weather and the combination of Time and Location. Weather is the parent attribute of this joint-child, creating a joint-child relationship.

![Joint-child relationship diagram]

---

Chapter 2  Attributes
To learn about...

...attribute-related tasks you can perform using MicroStrategy Desktop, please see the following topics in the How do I...? section.

Attributes

...the editors and wizards used to create and edit attributes, please see the following topics in the Interfaces section:

• Attribute Creation Wizard
• Attribute Editor
Facts and attributes are the two main building blocks for projects. Facts topics include:

- What is a fact?
- Types of facts
- Defining facts
- Level extensions
- Heterogeneous fact column names

**What is a fact?**

Facts are objects created by and shared between MicroStrategy Desktop users. They relate numeric data values from the data warehouse to the MicroStrategy Desktop reporting environment. The facts you create allow users to access data stored in a data warehouse. Facts form the basis for the majority of analyses and reports users can create with MicroStrategy Desktop.

Facts use physical columns to gain information from the data warehouse. The data warehouse sends data to the fact through the stored columns, as shown in the example below.

```
Data warehouse  Column  Fact
```

From column to fact

Although facts are associated with numeric columns from the data warehouse, they are not themselves data warehouse objects. Like other schema objects such as attributes, facts correspond to physical columns and tables.
Unlike attributes, facts do not actually describe data. Facts are the actual data values stored at a specific fact level. A fact entry level is the lowest attribute level in which a fact is stored.

Facts and attributes are necessary to define projects. In a MicroStrategy Desktop project, facts describe what data is used and attributes describe how that data is stored and organized. As Project Designer, you create projects which include facts and attributes, which the users can include when building metrics.

**Types of facts**

The types of facts are:

- Simple facts
- Implicit facts
- Derived facts

**Simple facts**

A simple fact is made up of one or more fact expressions. With a simple fact definition, you can define a fact as a column, constant, or simple expression.

MicroStrategy Desktop allows the creation of multiple columns with different names to represent the same fact. These are referred to as heterogeneous fact column names.

**Implicit facts**

An implicit fact is a virtual or constant fact that does not physically exist in the database because it is created at the application level. The implicit fact could have its expression, for example, criteria, defined as a constant value, though nothing is saved in a column. An implicit fact is used in a metric definition that does not contain other facts, and it indicates a fact table from which to retrieve data.

**Derived facts**

A derived fact has its value determined by an expression that combines two or more columns in a database to create a new column. For example, you can create a derived fact to generate new bank account numbers for customers of merged banks. By using two columns, such as bank code and
account number, you can merge the two columns, and place the derived number in a third, virtual column. Calculations and functions added to the expression assist in deriving data from the database by producing SQL with database-specific syntax.

When creating derived facts, you can use an **ApplySimple** pass through function. By including this function in the expression, you can add database-specific syntax which allows SQL to go directly to the database. Because each relational database requires different syntax, you need to verify this with the requirements of the database you use. For more information on pass through functions, see the following help topic:

Appendix A - Pass Through Expressions

### Defining facts

Every fact has three parts:

- **Fact Definition**: The fact definition is made up of one or more fact expressions. Every fact must have at least one expression.

- **Column Alias**: The column alias is the part of the fact that stores the column name. The column name is used by MicroStrategy Desktop to generate SQL statements when creating temporary tables related to the fact. Every fact must have a column alias, and MicroStrategy Desktop will select a default column alias depending on the type of fact, unless you create a new column alias.

- **Level Extensions (Advanced)**: Level extensions allow facts stored in the data warehouse at one level to be reported at a different level. You can use extensions to prevent a specific fact from being reported at a certain level, even though it is stored at that level. Level extensions are not commonly applied, but are very effective for special data modeling scenarios, if you are an advanced user.

---

**Note:**

- For a fact to exist in a MicroStrategy Desktop project, both the fact expression and column alias must be defined.

- During project creation, when you select the numeric column used to represent the fact, both the fact definition and column alias are automatically defined for you.
Fact definition

A fact definition contains properties which define a fact and its components. The fact definition can be comprised of one or more fact expressions and basic information about the fact, including the fact name, expression, and the source table it uses.

The example below demonstrates a fact definition including Name, Description, and Expressions. Within each definition there can be multiple expressions.

Fact definition

The fact expressions contained in the definition define how a fact is calculated by MicroStrategy Desktop. Facts can be found in multiple tables in a warehouse schema, and often must be calculated differently from one table to the next.

Note:

- Each fact expression relates to one or more related tables that contain the fact.
- Fact expressions define, for each of the related tables, how the fact is calculated.

Consider the following table, which lists fact definitions for simple facts:
A fact expression can be as simple as a fact column name from the warehouse, or as sophisticated as a formula containing fact columns and numeric constants.

Valid expressions are formulas constructed from attribute columns with or without numeric constants or mathematical operators. The mathematical operators that can be used in an expression are:

- addition (+)
- subtraction (-)
- multiplication (*)
- division (/)

A fact expression must contain one or more columns from a table with some mathematical operations, to return a business-specific calculation for the fact at that table level. Columns used in derived fact expressions must exist together in at least one table. If they do not, they must be modeled as a different fact before being used in any metric expression to replace the fact to be defined. For example, one fact table contains a fact, f1, and another contains a different fact, f2. You cannot assign a new fact (f3) as the sum of f1 and f2, and therefore a new metric cannot be defined as the average of f3. However, you can set up a metric to be the average of the sum of f1 and f2.
**Advanced expressions**

You can define an expression using database functions such as concatenate, by using the pass-through function. To do so, you need to use database-specific syntax, depending on which database you are using. For more information, see the following help topic:

Appendix A - Pass Through Expressions

**Column alias**

A column alias is a short name for a column and defines how a fact is referred to in SQL temporary table statements generated by MicroStrategy Desktop. Unlike expressions contained in the fact definition, a column alias does not define how the fact is calculated.

Every fact must have a column alias that can be used in the SQL statements containing the fact. You do not need to understand SQL, only ensure that an alias is always generated.

**Level extensions**

You can use level extensions to change a fact level, which is a set of attributes that represent the lowest level of detail at which the fact exists in the warehouse. Level extensions define how facts can be extended, lowered, or disallowed to other facts across the schema. By creating a level extension, you are allowing facts or attributes that have been captured at one level to be extended to other, technically unrelated levels, for reporting reasons. Each fact relates to a defined hierarchy based on the attributes present in the fact tables.

Level extensions are not requirements like the fact definition and column alias, and they tend to be used only in special cases. Level extensions can contain the expressions used by MicroStrategy Desktop to calculate fact values for attributes other than those contained in the original fact level.

---

**Rule:** To report a fact at a different level, you must define a level extension.
Before any facts can be resolved, the level extension must be defined, in case it is needed by a report. This resolution is accomplished by one of the methods listed below:

- Table relation
- Fact relation
- Degradation
- Cross-product
- Disallow the fact level

Level extensions are necessary when facts are often stored in the data warehouse at one level and reported at a different level. Every fact is tied to a set of attributes and hierarchies that may or may not satisfy all user-level reporting requirements. When a fact does not relate to all report hierarchies, there is a need for an explicit fact extension.

**Table relation**

A table relation defines a join on tables. When you create a join between a lookup or fact table, you are creating a table relation to extend a fact. A fact extension can be used to relate a fact to an attribute using a fact table.

**Fact relation**

Fact extensions can be defined by fact relation instead of table relation. With a fact relation, the table join is possible on any table that contains the fact. This gives you more flexibility in defining the relations, since the MicroStrategy Engine is responsible for choosing the appropriate table to join.

**Degradation**

Degradation, which lowers a fact level, is the logical opposite of aggregation. In a common retail schema, a fact exists at the Store level, but you can aggregate or sum the data up to the higher level by summing all of the Store level data into one aggregate total. However, when facts exist at a higher level than the report display level, you must specify how the Engine degrades the data to the lower level. When you lower the level at which a fact is reported, you are using degradation. This can be useful when the data stored at a certain level is actually invalid for certain reports.
For example, if your fact is stored at the Yearly level and you want to report the data at the monthly level, create a degradation. You also add an allocation expression to change the definition of the fact in a level extension. In this particular example, you select Month to be the Attribute to degrade to, and then specify that the allocation expression is Year/12. By creating allocation expressions, you are defining how higher-level facts are degraded to lower-level attributes. Allocation expressions are defined by operations you set on attributes and facts in the Level Extension Wizard. For more information on the Level Extension Wizard, see the following topic in the Interfaces section:

Level Extension Wizard.

**Cross product**

You can use a cross product join when a join does not exist, and where you need to force a fact to relate to an attribute by extending the fact. The cross product join allows a single fact value to relate to all elements of an unrelated attribute. This method can produce incorrect data because, in some cases, cross product joins allow reporting of facts that were not reportable using any other method.

When you specify a cross product join to relate a fact to an attribute, you are creating a Cartesian product of the lookup attribute. As this method can be inefficient, MicroStrategy does not recommend using the cross product extension.

**Disallow**

A disallow is like a lock, which you use to prevent a fact from being reported at a specific level. For example, you can select an attribute or set of attributes at which level you do not want the fact reported. Instead of extending to a specific level, the disallow does not allow a fact to be reported at that level.

If a fact is stored at the Month and Year level, but your customer does not want the fact reported at the Month level, you can create a disallow extension for Month. With the disallow in place, if you create a report and attempt to include the fact at the Month level, the MicroStrategy Desktop returns an error indicating the report cannot be run at that level.

You can create a partial or complete disallow to keep a fact from being reported. A partial disallow is when you select some but not all attributes for the disallow.
Heterogeneous fact column names

MicroStrategy Desktop allows you to create heterogeneous fact column names for each fact column. With heterogeneous column names, you can refer to the same fact column with multiple names. To do this, MicroStrategy Desktop allows you to create multiple fact expressions for the same fact, which represent or define the fact using different names. This allows you greater flexibility because you can give each fact column more than one name, which allows you to include the fact in a report more than once. You can use the Fact Editor to create fact expressions.

A VMall solution: Item Sales

Item Sales has two expressions. The first contains the default column

\([\text{Item\_Sls}]\)

the second uses a fact expression to calculate the value:

\([(\text{Number\_Items\_Sold} \times \text{Unit\_Price}) \times 1.05]\)

Both expressions represent the same fact.

To learn about...

...fact-related tasks you can perform using MicroStrategy Desktop, please see the following help topic in the How do I...? section:

Facts

...the editors and wizards used to create and edit facts, please see the following topics in the Interfaces section:

• Fact Creation Wizard
• Fact Editor
CHAPTER 4

Hierarchies

Hierarchies topics include:
• What is a hierarchy?
• Types of hierarchies
• Hierarchy display
• Hierarchy browsing

What is a hierarchy?

Hierarchies are groupings of attributes which can be displayed, ordered or unordered, to reflect their relationships to other attributes. There are two types of hierarchies: user and system. A user hierarchy is unordered, and you can easily change its design to include additional attributes, or limit the user access. The system hierarchy is ordered, and it is created automatically when you create new projects.

The best design for a hierarchy is to organize or group attributes into logical business areas. For example, you can place related attributes into hierarchies by their level.

The example below demonstrates the Location and Customer hierarchies. Within the Location hierarchy, State, City, and Store are organized according to their relationships. The Customer hierarchy also groups together the attributes Company, Contact, and Customer ID.
Hierarchies must be separate and distinct from one another and include at least one attribute, which can be used in other hierarchies. Default hierarchies are user-specified hierarchical paths. Setting these defaults causes the hierarchy to open to a default attribute element immediately when expanded. You can define default hierarchies, though only one default can be set for each hierarchy.

Hierarchy structure

Hierarchies are structures based on relationships you define between attributes. A hierarchy allows you to logically define and order groups of attributes. With a set hierarchy, you can navigate and organize attributes in your project.

When you group attributes together into hierarchies, you are developing a working design of the display and browse functions of the attributes. In the example below, there are two instances of the Region hierarchy. One hierarchy demonstrates Region having multiple States and the States having multiple Stores. This hierarchy allows you to create drilling and browsing options to the lower levels, to view Region, State, and Store on a report. But if you only include Store in the Region hierarchy, as in the second example, then the only options for drilling or browsing are the Region and Store levels.
There are two types of hierarchies:

- **System hierarchy**: The system hierarchy specifies an ordered set of all attributes in the project, but does not define ordering or grouping among attributes. There is only one system hierarchy in each project.

- **User hierarchy**: User hierarchies are named sets of attributes and their relationships, arranged in specific sequences for a logical business organization.

**System hierarchy**

The system hierarchy is the default hierarchy MicroStrategy Desktop sets for you each time you create a project. It contains all of the attributes in the project and is actually part of the definition of the schema. When you first create a project, it contains only the system hierarchy.

The system hierarchy holds information on the relationships between attributes in the project. The system hierarchy cannot be edited, but is updated every time you add or remove children or parents in the attribute editor, or when you define children in the Project Creation Assistant.

The system hierarchy is useful in determining relationships between objects. Attributes from the system hierarchy do not need to be part of an explicitly-defined user hierarchy. Any unassigned attributes remain available to the system as report objects, filter conditions, and components of consolidations.

**Rule**: You can view the system hierarchy only in the Data Explorer.
Data Explorer

The Data Explorer is a tool in the Object Browser that holds the system hierarchy and the user hierarchies. As a tool, it makes the hierarchies available for users to include in new reports. When you create a new project, the system hierarchy for that project is automatically placed in the Data Explorer. User hierarchies, however, are saved to the Hierarchies folder in the Object Browser. You can move user hierarchies to the Data Explorer folder (under Hierarchies folder) in the Object Browser when you want them available for use in element browsing. Moving to and from this folder allows you to keep some hierarchies visible to the user while hiding others.

To learn how to view the Data Explorer, see the following help topic in the How do I...? section:

Hierarchies

User hierarchies

When you create user hierarchies you can define the browse and drill relationships between attributes. You can create these hierarchies in the Hierarchy Editor using one or more attributes from the system hierarchy. Attributes from the system hierarchy are listed for you to choose from.

The user hierarchy is the only type of hierarchy you can define, and you can create unlimited user hierarchies for each project.

Tip: You need to define user hierarchies that correspond to specific areas of your company business model and data warehouse schema.

Hierarchy display

You can perform the following actions to control hierarchy display:

- Lock a hierarchy
- Limit a hierarchy
- Filter a hierarchy
- Set an entry point
Locked hierarchy

A hierarchy is referred to as being “locked” when one or more attributes within that hierarchy has the Element display option set to Locked. Locking a hierarchy prevents viewing elements of the specific attribute and any lower level attributes in the hierarchy. Anything higher in the hierarchy is still visible.

You can lock the hierarchy to restrict viewing elements for security reasons or to better manage lengthy hierarchies. By restricting the view of attribute elements in the Data Explorer, you can prevent the expansion of long attribute element lists that can consume system resources. When you set the element display to locked, a padlock icon displays next to the hierarchy name.

Limited hierarchy

Another way to restrict the viewing of attribute elements in the Data Explorer is to limit the number of elements that display at one time. This method is useful when there are extensive attribute elements in a hierarchy. Instead of loading all attribute elements at once, you can set the limit to five or ten at time. You can then click the arrows to see the next set.

Filtered Hierarchy

You can add filters to a hierarchy to control how data is retrieved and displayed. With a filter you can choose exactly what attributes display. For example, you can filter a hierarchy so that data for only one quarter displays, or data for only a few individual days of one quarter. Filters make data retrieval faster by only allowing specific data to display.

Each attribute in the hierarchy can have multiple filters applied to it. When filtering attributes in a hierarchy, you are limiting the elements of the data returned when you browse the Data Explorer. Whereas setting limits can reduce the number of elements displayed at one time, filters can limit the scope and return more detailed results.

Filters increase efficiency when retrieving data. You can limit user access to parts of a hierarchy when you apply filters to attributes. The filters allow the Data Explorer to display only the criteria you select, and the user is unable to select additional data in the hierarchy.
When adding filters to a hierarchy attribute, you need to make sure that it is relevant to the attribute’s information. MicroStrategy Desktop does not validate that the associated filter makes sense on that attribute. That is the responsibility of the user.

Entry point

An entry point is a shortcut to an attribute element in the Data Explorer. Creating an entry point grants you faster access to the attribute without having to open multiple folders to reach different levels of the hierarchy.

When you create a user hierarchy, the hierarchy, attributes, and their elements display in the Data Explorer. When you set an attribute to be an entry point, you are creating a shorter route to open folders. For example, a typical hierarchy might be Time. When you click on Time, folders for each Year open, such as 1999, 1998, and 1997. When you click on 1999, a folder for each Quarter opens, such as Q1, Q2, Q3, and Q4. If you are seeking Week24, this means you need to open several levels of folders to reach the correct data level, which is Week. If you set the attribute Week as an entry point, the folder Week displays in the Data Explorer at the same level as Time. If an attribute is not set to be an entry point, it displays in its normal hierarchy structure.

If you set a locked attribute as an entry point, it still displays in the hierarchy, but a with padlock icon. You can see the locked entry point, but you are not able to access attributes below that level.

Hierarchy browsing

You can design the user hierarchy browsing for the Data Explorer by applying browse attributes. A browse attribute is the attribute child defined for the hierarchy attribute. When you apply browse attributes to attributes in a hierarchy, you are specifying what levels of detail are visible when browsing the Data Explorer.

Once you choose which attributes to place in a hierarchy, you need to define the relationships between them. These relationships determine how the users can browse the attributes from the Hierarchies folder. For example, if Year, Month, Week, and Day are the attributes that comprise the hierarchy Time, the hierarchy resembles the example below.
Time Hierarchy

For each attribute you select to be a part of the hierarchy, you can assign to it one or more browse attributes. For example, assume that the same attributes have been defined for the Time hierarchy. Some of these attributes have been assigned a browse attribute. For example:

<table>
<thead>
<tr>
<th>Hierarchy Attribute</th>
<th>Browse Attribute(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Month, Week</td>
</tr>
<tr>
<td>Month</td>
<td>Week</td>
</tr>
<tr>
<td>Week</td>
<td>Day, Year</td>
</tr>
<tr>
<td>Day</td>
<td></td>
</tr>
</tbody>
</table>

The addition of these browse attributes allows you to see the Week elements directly from the Year attribute, without having to first view the Month attributes. The new hierarchy resembles the example below.

Hierarchy with browsing
From the Hierarchies folder, it resembles the example below.

![Hierarchies folder structure](image)

You can now view the weeks in 1994, without first having to browse through the months.

**Drilling**

Drilling is a function in MicroStrategy Desktop reports that allows you to browse lower levels of attributes along pre-defined criteria. When you create hierarchies you can specify what level attributes are included in drilling. Depending on the level of the attributes included in the drilling specification, reports that use the hierarchy allow the user to drill down to lower levels of detail. Basically, when the user selects a drilling level, the reports refresh to display that level of detail. The user is seeing the report but is essentially drilling through the hierarchy used in building the report.

To enable drilling, you must select the hierarchy to be used as a drill hierarchy in the Hierarchy Editor. This makes the drill option available when the user adds the hierarchy to a report. If you do not specify the drilling option for a hierarchy, then drilling does not appear in the right-click menu of any report created using the hierarchy.
To learn about...

...hierarchy-related tasks you can perform using MicroStrategy Desktop, please see the following topic in the How do I...? section:
Hierarchies

...the Hierarchy Editor, please see the following topic in the Interfaces section:
Hierarchy Editor
Partitions topics include:

- What is partition mapping?
- Types of partitioning

What is partition mapping?

Partition mapping divides large logical tables into smaller physical tables. Partitions improve query performance by minimizing the number of tables and records within a table that must be read to satisfy queries issued against the warehouse. By distributing usage across multiple tables, partitions improve the speed and efficiency of database queries.

Time is the most common category for partitioning databases. Partitioning by Time limits growth of the database and increases stability. If you partition by Time and need to cross partitions to retrieve data, you can create multiple data slices to filter information from multiple partitions. The data slices act like filters to return only the slice of data you request.

MicroStrategy Desktop allows you to partition a table along any number of attributes.

Types of partitioning

MicroStrategy Desktop supports two types of partitioning:

- Warehouse partition mapping
- Metadata partition mapping

Warehouse partition mapping

Warehouse partition mapping is the mapping of partitions carried out and maintained in the warehouse, at the server level. You can define a Warehouse partition by adding a table with a special structure through the
Warehouse catalog. This special table contains the map for the partition, which is stored in the warehouse. Warehouse partitions divide tables physically along any number of attributes, though this is not visible.

The partition level appears as one table, but is actually multiple smaller tables. Warehouse partitions must be homogenous, so that columns, facts, and attributes exist at all levels. Homogenous partitioning divides data of equal levels, like January and February.

**Tip:** For SQL users:

The SQL engine uses the filtering information to determine the physical tables that need to be retrieved. The SQL engine runs a partition pre-query against the partition mapping table and gets back a list of physical tables to be used. It then generates SQL for each table and applies the partitioning logic.

**Notes:**

- There are no data slices in the warehouse partition.
- MicroStrategy Desktop supports Warehouse partitions on upgraded and newly created projects. These are added using the Warehouse Catalog Browser. See the warehouse Catalog section for more information.

**Metadata partition mapping**

Metadata partition mapping is the mapping of partitions carried out and maintained in the project metadata, at the application level. In application-level partitioning, the relational database is not aware of partitioned tables. MicroStrategy Desktop manages the mapping between the logical table and the physical tables. This design makes it easier for you to specify a flexible partitioning schema. Unlike warehouse partitions, metadata partitions can be homogenous or heterogeneous.

**Data slices**

A data slice acts as a filter that defines what portions of data to place in the partition table. When you create a data slice, you are specifying how a table is partitioned. Based on this data slice, the engine knows which table to pick when generating the SQL.
A data slice holds the parameters that a partition is based upon, for example Month=January. Instead of retrieving data for all months, the server returns only data for January. By creating a data slice with the partition, you can retrieve specific data without time-consuming joins and searches.

It is very important to create a reasonable and valid data slice because MicroStrategy Desktop has no way of verifying its accuracy or relevance. Thus, the information is known only by you, the project designer. Basically, the data slice needs to make sense for the data. A poorly crafted data slice can lead to errors from generating incorrect SQL and retrieving the wrong data.

Data slicing displays and can be modified only for the metadata partitioning. Each partition mapping table must include one or more data slices. In a heterogeneous schema, data slices can exist at different levels and can be composed of different keys.

**Attribute qualifications**

To create data slices, you use attribute qualifications. Attribute qualifications are types of filters that are applied to attributes forms. These qualifications allow you to limit the type and amount of data that is returned for a report. For example, if you create a report that contains the attribute Country but you only want to return results for France, you can create a qualification on the attribute Country and use France as the element that appears on the report.

**To learn about...**

...partition-related tasks you can perform using MicroStrategy Desktop, please see the following help topic in the How do I...? section:

Partitions

...the Partition Editor, please see the following interface topic in the Interfaces section:

Partition Editor
This chapter presents the terms and concepts related to projects. It contains information that will help you understand what projects are, and how you create, upgrade, and duplicate them.

The main topics in this chapter are:

• What is a project?
• From warehouse to project
• Project sources
  ◊ Project source types
  ◊ Project source tools and project
  ◊ Project source creation
• Project Types
  ◊ MicroStrategy 7 Desktop project
  ◊ MicroStrategy 6.x and DSS Agent 5.x
• Creating, upgrading, and duplicating projects
  ◊ Create a project
  ◊ Upgrade a project
  ◊ Duplicate a project
• Warehouse catalog
  ◊ Creating a database instance

What is a project?

A project is the means through which you can access MicroStrategy Desktop functionality. A project is the highest-level intersection of a data warehouse, metadata repository, and user community. A typical project contains reports, filters, metrics, and functions. As Project Designer, you create the projects that users access to run reports.
Projects are where you create and modify objects that the Report Designer can incorporate into future reports. When you create projects, you place each project within a project source. You can create a project source by using the Project Source Manager.

In MicroStrategy Desktop, a project can easily be created through the Project Creation Assistant. The assistant takes you through the creation of a full project from the name definition and database connections to the definition of attributes. For more information, see the Project Creation Assistant Toolkit chapter.

From warehouse to project

MicroStrategy Desktop schema objects such as attributes, facts, and tables are in essence abstractions built on top of database objects such as tables, views, and columns. To start building different objects, the system requires certain basic information about the warehouse such as: What tables are available in it? What columns do they contain, and what are their datatypes? This type of information is stored in the Warehouse Catalog.

Through the Warehouse Catalog Browser, you can select the data warehouse tables to be used in the project. If the Project Creation Assistant menu option is not available, verify the user privileges with the Administrator. At any time after creating a project, you can change the tables that are included in the project using the Warehouse Catalog Browser. The Warehouse Catalog is in the DBMS and holds all table and column information in the warehouse. The Warehouse Catalog Browser is the tool or editor that enables MicroStrategy Desktop to access and modify it for MicroStrategy Desktop projects.

Project sources

A project source is the information which defines a project, including metadata connections and the location of the warehouse. You can have multiple projects associated with each project source.
Connection Modes

The connection modes determine how the projects connect to the metadata for data retrieval. A project source can have one of three connection modes:

- **Server (3-tier):** A 3-tier connection mode connects the project to the metadata via the MicroStrategy Server. You need to know the server name and port number to setup this connection.

- **Direct (2-tier):** A 2-tier connection mode connects the project to the metadata via an Open Database Connectivity (ODBC) data source name (DSN). You need to know which database the project uses.

- **6.x Project:** A project using this connection mode connects to the metadata of prior MicroStrategy products via the ODBC DSN specified in a .dss file that includes project preferences. You must specify the .dss file.

Project source tools and projects

Project sources include administrative tools and projects. Some objects can be shared and reused, such as connections, users, schedules, and database instances.

Project creation script (.pds) files are included with MicroStrategy Desktop. These files control what is included in a new project and how a project is created.

You can create a project source of 3-tier or 2-tier connections using the default project creation scripts (.pds) included in MicroStrategy Desktop. You do not need to create or modify any .pds files. It is recommended that only the Administrator change the .pds files.

Project types

You can create or duplicate existing projects in MicroStrategy Desktop or upgrade existing projects created with an earlier version. Types of projects supported by MicroStrategy Desktop include:

- MicroStrategy Desktop project
- MicroStrategy Agent 6.x
- DSS Agent 5.x
MicroStrategy Desktop

When creating a new project in MicroStrategy Desktop, use the Project Creation Assistant. Through the Assistant, you can create facts and attributes, select warehouse tables, and build relationships. You can make changes to the project through the following editors:

- Attribute Editor
- Fact Editor
- Hierarchy Editor
- Partition Editor
- Transformation Editor
- Table Editor
- Warehouse Catalog Browser

MicroStrategy Agent 6.x and DSS Agent 5.x

MicroStrategy Agent 6.x and DSS Agent 5.x projects are older projects which are currently supported by MicroStrategy Desktop through upgrades. These projects are read-only and can be viewed using MicroStrategy Desktop.

A MicroStrategy Agent 6.x or earlier project can be identified by its project connection mode. The project has a 6.x or earlier project connection mode for its project source setup. Only one 6.x or earlier project can exist within a project source. If you need to view more than one 6.x or earlier project, you must create additional project sources.

You can utilize the new features and functionalities of MicroStrategy Desktop only after you upgrade your existing project to a MicroStrategy Desktop project. This can be easily done using the Project Upgrade Wizard. For more information, see the Upgrade manual.

Duplicating projects

You can duplicate MicroStrategy Desktop projects if they were created in MicroStrategy 7. The duplication feature allows you to quickly create projects based on existing designs. You can also create backups, deploy the the same design to other users, create new projects with new reports, metrics, filters, but with the same schema, and point the project on a server to a different repository. After duplication, you can modify the projects as time allows.
For more information, see the following interface chapter:
Project Duplication Wizard

Warehouse Catalog

When creating a new MicroStrategy Desktop project you can choose the tables you want included in the project. The table selections can be changed anytime using the Warehouse Catalog Browser.

Creating a database instance

A major step in building a project is creating a new warehouse database instance. A database instance is a logical representation of a physical database in the system. When creating a project, the Database Instance Wizard displays to guide you quickly through the process of creating the database instance.

To learn about...

...project-related tasks you can perform using MicroStrategy Desktop, please see the following topic in the How do I...? section:
Projects

...the Project Creation Assistant, please see the following topic in the Interfaces section of this guide:
Project Creation Assistant
Transformations topics include:

- What is a transformation?
- Creating a transformation

What is a transformation?

A transformation is a group of member attributes that share the same kind of analysis on time. Multiple member attributes can define a transformation.

Transformations are based on tables, such as transformation tables that you select to compare values at different times, for example, this year vs. last year sales, or new customers this month vs. last month. By using transformations to track sales for the same day a month or year ago, you can easily recognize and analyze trends.

You can use transformations to define metrics based on the member attributes included. The user only needs to create a single metric for each transformation, and the transformation analysis applies depending on the level of the report where it is used.

Creating a transformation

A typical transformation analyzes sales stored at the daily level. To retrieve the sales information, the user needs to compare the sales of today to the sales of last year for the same date. In this case, you create the transformation Last Year, then add each of the attributes that become the members of this transformation. The member attributes for this example are Day, Month, and Quarter. For each member attribute, you define an expression or column and table where it is located. The user can then employ this transformation to create a metric Sales Last Year. This metric can be used in reports which contains any of the member attributes of the
Transformation. So, if the report includes Day, the transformation is applied at the Day level; if the report includes Month, the transformation is applied at the Month level, and so on.

A transformation manipulates member attributes using relationships to retrieve specific time related information. This is accomplished with a transformation table or transformation expression, which basically creates a relationship that maps the original attribute to the transformation.

Transformations are useful because they enable you to create metrics that return data which has a particular relationship to the current filter attributes.

**Member attributes**

Transformations are defined by the member attributes grouped in them. When you add member attributes to a transformation, you select attributes from the list of attributes already defined in the project.

**Member attribute expression**

After adding member attributes, you create an expression for the transformation. The table you select for the expression must include the attribute(s) you have included. Member attribute expressions retrieve information from the current member attribute and make it available to the transformation. Since transformations are expression-based, they generate SQL using the expressions. To define a transformation expression, you can use columns found on a transformation table, the member attribute lookup table, or any other table where the member attribute is found.

**Transformation mapping type**

When creating a transformation, you can select the mapping type. The transformation mapping type determines the way the transformation is created based on the nature of the data. You choose the mapping type to select how the transformation relationships are mapped: one-to-one or many-to-many.

**One-to-one mapping type**

A typical one-to-one relationship is Last Year Day to This Year Day. This is one-to-one because one day or month in last year maps to a day or month in this year. The one-to-one transformation mapping type is the default for a new transformation.
Many-to-many mapping type

A typical many-to-many relationship is Year to Date. This is many-to-many because for a year there many days or months that can be included for this transformation since it is cumulative.

To learn about...

...transformation-related tasks you can perform using MicroStrategy Desktop, please see the following help topic in the How do I...? section: Transformations

...the Transformation Editor, please see the topic in the Interfaces section: Transformation Editor
Putting it all together topics include:

- Development sequence
- Inputs to modeling task
- Approach to developing a data model
- Developing and using generic models
- Bottom-up modeling
- Top-down modeling

The primary goal of data modeling is to determine customer requirements and to produce a model that meets those needs. Developing a customer-compliant model involves the identification of requirements, design of solutions, and the evaluation of the solutions. Data modeling is an iterative process that also becomes an investigative process wherein additional questions and concerns arise with each draft of the model design.

Getting started usually involves borrowing parts of standard, successful models, although forcing a generic model to fit the needs of a custom project is not recommended. During this process, data is organized in tables by hierarchy, attribute, element, fact, and relationship.

**Development sequence**

In the normal order of designing software applications and data models, the preferred development sequence of functional operations is create, read, update, and delete. All methodologies involve these dependencies in varying levels of sequence flexibility. In short, there are two approaches to starting the design process. The first approach is to design the create function first. The model fills out with data, starting from the top of the hierarchy and working down from there. The advantage here is that it is possible to progressively expand upon data that has already been entered.
The second approach is to design core functions first and build the necessary data structures to support them. This is the preferred method because it allows a more complete model to be built in the early stages of development, and therefore makes it easier to locate critical issues when there is time to take corrective action.

**Inputs to modeling task**

A wealth of inputs is available to help data modelers identify requirements. Interviews with business specialists can provide a valuable resource of information. Existing systems and files, system proposals, functional specifications, mission and job position statements, policy and procedure manuals, and any relevant documentation can be consulted as well.

**Interviews**

While interviews can be a rich resource of information, approach them cautiously. Talk to members of senior management after exhausting other interview candidates. Be wary of interviewing people who are offered up as user advocates, as they may lack the knowledge base to provide the critical information necessary to create an effective data model. Resist the temptation to start the modeling process at the interview stage. A premature start can skew objectivity before other important resources are considered and can render a data model that is not merely a reflection of business requirements, but rather is the business requirement.

**Existing systems and reverse engineering**

Much information can be derived from existing database designs. Becoming familiar with existing systems can provide you with insight on what end users are looking for in the new system. The existing system can also help you develop a list of attributes and relationships that can translate to the new system.

If the current system lacks design documentation, you must resort to reconstructing the data model through a process known as reverse engineering. Personnel who have worked extensively with the system, such as maintenance analysts or programmers, are your best guides to
uncovering system nuances that might otherwise be overlooked. As you prepare to reverse engineer your way to a data model of the current system, keep these steps in mind:

- Represent existing files, segments, record types, tables, or equivalents as entities.
- Identify relationships supported by hard links.
- Identify relationships supported by foreign keys.
- List the attributes for each entity and define each entity and attribute.

**Functional models**

Functional models focus on what a system does. Normally, all functions are identified with the data needed for each. This level of detail reveals data used by the process and therefore what is important to the data model. Having a function model as a precursor to the data model is a good idea, even if it is a one or two level data diagram.

**Approach to developing a data model**

Before starting work on a data model, remember the purpose of the data model in the scheme of the larger picture. Most of the first steps in data modeling start out as pen and paper designs as a means to understanding on the overall task. Entities and their relationships are briefly sketched out.

**Identifying entities**

Part of the investigation responsibilities of the data modeler is to first understand customer requirements. This is done through finding the scope of the project through a customer survey asking what kinds of data to maintain and how that data is related. The result can produce a list of entities important to the conduct of the subject business.

You can then group together elements that the business handles in the same manner. Proposed data model designs will next to be verified with the customer.
Using patterns

Patterns from previous data modeling efforts can often be used as a template to help start the data modeling process on a current project. Generally speaking, there are numerous data modeling designs built on proven structures that are available to the data modeler. One is called the standard structure, the other the generic model.

The standard structure is a reusable part of a data model. It can be used as a building block for specific areas of a total model. On a wider scale, the generic model is the better pattern to employ. It covers the range of a complete enterprise and is available for business activities like accounting, asset management, and human resource management.

Developing and using generic models

Generic models are usually discovered when data modelers review their experience while working on past projects. Normally, no generic model for a particular business activity is available in book form for ready use. Generic models can be inferred, however, from existing data models of similar purpose.

The idea of the generic model is to furnish a template that can be readily customized to suit the application at hand.

Adapting generic models from other applications

Not all business activities have a generic model that can directly cross over into the current application. Such a situation produces the need to derive a generic model from other non-related applications. In essence, you are selecting bits and pieces from another application to fit your specific application. The component parts from several different non-related applications can be combined to form one new data model. Non-conforming or inappropriate information is found during the review cycle.

Developing a generic model

The pitfall of trying to adapt a generic model to a current model is that a force-fit of an inherited problem can result in a model that is difficult to understand. Building a personal repository of generic models through experimentation with alternative solutions is the best approach to having a generic model ready for any given application. The mind and experience of the data modeler is the best resource for generic model development.
When there is no generic model

When the occasional situation arises where a generic model does not work, it is time to develop a new generic model through either the bottom-up or top-down approach.

Using VMall as a sample project

VMall is a sample project that demonstrates the functionality of the MicroStrategy Desktop and offers you, the data modeler, a glimpse of what a finished data modeling project with reporting capability looks like. VMall contains hypothetical data stored in a sample data warehouse against which SQL can be run for instructional purposes. As a sample project, it also provides the opportunity for you to learn the MicroStrategy Desktop without risking corruption to live data.

VMall is a project built on a data model and thus shows how the data model lays the foundation for not only the physical schema but also the project model. The MicroStrategy project is built directly off the data model.

The relevant part of VMall for you is the relationship of the attributes within the model. Look into the Schema Objects folder to see how attributes, facts, and other objects are defined.

Running a report in VMall is the same as running any report through the Desktop. The only difference is that it is a sample project that uses dummy data and a generic model.

Bottom-up modeling

The bottom-up modeling approach is invoked when other generic models do not satisfy task requirements. This approach develops a literal model based on existing data structures and terminology. Subtyping and supertyping (a process involving generalization and specialization) permits the data modeler to move toward other options. By generalizing, you can broaden entity meanings to encompass more functions. Thus, you move up or expand the model from its original structure.
Top-down modeling

The top-down modeling approach allows for breaking a large modeling problem into manageable parts. Start by using a model generic enough to include the primary entities (the parts) of any given business. Then ask a series of questions about what kinds of information can be filled in behind the following entity names:

- Party
- Contract or Agreement
- Product
- Resource
- Event
- Location
- Account

This process is only a starting point as it usually produces a literal and incomplete model. Use other techniques to flesh it out.
Project Designer Interfaces

• Attribute Editor
• Attribute Creation Wizard
• Fact Editor
• Fact Creation Wizard
• Hierarchy Editor
• Level Extension Wizard
• Print Project Schema
• Partition Mapping Editor
• Project Creation Assistant
• Project Duplication Wizard
• Table Editor
• Transformation Editor
• Warehouse Catalog Browser
CHAPTER 9

Attribute Editor

What is it?

Attribute Editor allows you to create and edit attributes.

How do I access it?

You can access the Attribute Editor from the File menu by choosing New, then Attribute.

What can I do with it?

Through the Attribute Editor, you can:
• Add, modify, and delete attribute forms
• Select the form type
• Create or delete form categories
• Define a column alias for the column used for temporary table SQL generation
• Add or modify the alias name and column properties of a form
• Add, modify, or delete a form expression
• Identify the source tables of the columns used in the expression
• Choose to have the system select all available tables as sources of the columns used in the expression
• Define a new parent or child for an attribute
• Define attribute display properties

For information on completing one of the above tasks, refer to the following How do I...? topic:

Attributes
**What should I know before I use it?**

Before you begin using the Attribute Editor, you need to be familiar with:
- attribute forms
- parent and child relationships
- your schema
- compound keys—what they are, and why you use them
- lookup tables—what they are and why you use them
- column names for the temporary tables for SQL generation
- the following terms:
  - byte length
  - bit length
  - precision
  - scale
  - time scale

For more information on the above topics, refer to the following **Concept** topic:
Attributes

**Attribute Editor layout**

The editor has the following tabs:
- Forms
- Children
- Parents
- Display

**Forms tab**

The **Forms** tab allows you to define an attribute and its related properties. On the **Forms** tab, the **Attribute Forms** list displays all forms defined for the attribute and their details such as Form name, Form category, Format type, and Form description. You can double-click the attribute form to change any of these fields in the **Modify Attribute Form** dialog box.
When you highlight an attribute form, the Form expressions and Source tables lists change to display all form expressions and source tables created for the selected attribute form. Double-click the attribute form to modify its properties.

Form expressions

The Form expressions list displays the form expressions and their respective Mapping methods (Automatic or Manual) for each attribute form you select. The automatic mapping method selects tables for each attribute, while the manual mapping method allows you to select specific tables for each attribute. You can change the mapping method on the Modify Form Expression dialog box, which you can open by clicking Modify on the Attribute Editor.

Valid expressions that can represent an attribute form can be columns or operations between columns (Derived Attributes), or involve constant values (Implicit Attributes). If you create an expression which is only a constant value (for example, 1), you are creating an expression that exists on the table, but is not defined physically in the warehouse.

Note: Heterogeneous schema support allows you to make an item present in multiple tables. For example, if the ID of the attribute Item is represented by column Item_Id in one source table and Item_Nbr in another table, the Item attribute can be present in both tables. To do this, create two form expressions for an attribute form, one with the column Item_Id and the other with the column Item_Nbr.

Source tables

The Source tables list displays the source tables for each Attribute form you have selected, indicating in which tables the attribute exists. The listed tables contain the columns used in the selected attribute form. The table chosen as the lookup table displays in bold. You can change the lookup table selection on the Modify Attribute Form dialog box when you click Modify on the Attribute Editor. In the Modify Attribute Form dialog, select a source table from the list and click the Set as Lookup button.

Set as key

From the Forms tab, you can set an attribute form as a key, or group similar attribute forms. When creating an attribute in the Attribute Editor, one attribute form must be set as key, which acts as a primary key. To set
or change the key, select the attribute form on the Forms tab and click Set as Key. The attribute form is in bold when it has been set as key. If you set an attribute group as key, the group name and all attribute forms within it are listed in bold.

**Grouping (Compound attributes)**

An attribute form group is a group of attribute forms that are related to each other and share the same form category (for example, ID, DESC). When you create two attribute forms which are dependent on each other, you can group them together. Select the attribute forms to be grouped and click Group. The Create New Attribute Form Group dialog box opens. Enter a name for the group and select a Form Category from the list, or click Modify to change the existing form categories.

When you choose the Form Category for the attribute group, the Form Category for each attribute form in the new group is changed on the Forms tab. You can see the new group displayed on the Forms tab. The form group name is listed first, followed by the attribute forms, indented below.

**Compound keys**

The compound key is a compound attribute, or group of attributes, which has been set as key. When you create multiple forms with the same category, then you must group them into a form group. When you set this form group as key, you are creating a compound key.

A compound key is needed when an attribute cannot be uniquely identified with just one key. The grouped attributes appear indented in the list and the group name appears directly above the group. Only the attribute name and category information is shown for a group name.

**Ungrouping**

When grouped attributes are selected, the Group button changes to Ungroup. To remove a form attribute from a group, select the form and click Ungroup. The group will remain, but the attribute form is no longer indented. It stands alone.

If you want to ungroup all form attributes, select the form group and click Ungroup. The New Key Form selection dialog box opens and you must select a new form category for the attribute forms in the group. This process applies only when the form group to ungrouped is the key. If the form group to be ungrouped is not the key, the New Key Form dialog will not appear at all. Multiple attribute forms with the same form category must be made into a form group, or the form category changed.
From the Forms tab, you can access the following dialog boxes:

- **Create New Attribute Form**: Allows you to create an attribute form
- Modify Attribute Form: Allows you to update an attribute form
- Create New Form Expression: Allows you to create a new form expression
- Modify Form Expression: Allows you to update a form expression

**Create New Attribute Form**

**Definition tab**

From the Attribute Editor, click New to open the New Attribute Form dialog box. The Form expression and Source tables together define the attribute form. Each form expression you create for the attribute form displays in the Expressions box. You can create new form expressions by clicking New, which opens the Create New Form Expression dialog box.

There are also the following fields:

- **Form name**: Is the Name entered on the Create New Attribute Form dialog box when the form was created. This is a required field.
- **Form description**: Is a helpful quick reference as to the purpose of the form. This is an optional field.
- Form category: Contains the category of the form: ID, Desc, or None. You must pick one from the drop-down list. You can also change the definition of a category or define a new category by clicking Modify.
- **Format type**: Displays formats you choose for the display of the form, such as Number, Text, HTML, and Picture. This is a required field, and the default is Number.
- **Default sort**: Presents sorting options you select (Ascending or Descending) and applies to the attribute form you create, when it is displayed on a report. The order in which the sorting is applied is determined by your selections on the Attribute Editor Display tab (display section). This is an optional field, and the default is None.

**Column Alias tab**

The column alias is the column used in temporary table generation. A default is always created for you, but you can modify this alias by clicking Modify.

On the Column selection dialog box, you can choose a different column or create a new custom column by clicking New. A custom created column exists only for SQL generation on temporary tables. After it serves its purpose, the custom-created column ceases to exist. The Modify button is available only when you select a column you have created.
Modify Attribute Form

From the Attribute Editor, select the Attribute Form you want to change, and click Modify to open the Modify Attribute Form dialog box. The fields are the same as listed in the Create New Attribute Form section above.

Create New Form Expression

From the Create New Attribute Form dialog box, click New to open the Create New Form Expression dialog box. You can select a table from the Source table list to use as the source for the expression. Depending on the table you select, a list of the table columns displays in the Available columns grid.

Form expressions define how the information is retrieved from the source tables. You can create a form expression by double-clicking or dragging columns into the form expression box, and clicking on the mathematical operators to build the expression. As you build the expression, you can click Clear to remove an expression or Validate to query whether the expression is correct as written.

When you create the expression, you can select the Automatic or Manual mapping method. Automatic mapping selects for you all available tables that contain the columns being used in the expression as source tables. Manual mapping means that you will manually select each table that contains the column used in the expression. MicroStrategy recommends you choose Automatic if you are unsure what tables contain the column used in the expression.

The default Mapping method for the first expression in a form is Automatic. The default for each subsequent expression created is Manual. If you define a constant as an expression when creating an Implicit attribute, you must set the mapping to Manual.

If you select the Automatic mapping method when you are adding new tables to the project through the Warehouse Catalog Browser, this attribute will be mapped to those new tables if they contain the columns used to define the expression. This mapping method saves you from manually mapping objects to these new tables.

Modify Form Expression

Click Modify to open the Modify Attribute Form dialog box, select the form expression, and click Modify. The fields are the same as listed in the Create New Form Expression section above.
Children tab

The Children tab is used to define the relationships that an attribute has with its children. In the Attribute children grid, you can change the Relationship type and Relationship table from their respective lists. Click each to view the lists.

The Description box below the Attribute children grid puts the Relationship type in context to make it easier to understand. The blue text is the actual relationship.

From the Children tab you can access the Add Children Attributes dialog box when you click Add. This dialog box allows you to define a new child. The Children candidates box lists the available attributes obtained from the schema hierarchy relationships, defined during project creation. To add new children, select an attribute under Child candidates, and use the arrow button to move it to the Selected children box.

Parents tab

The Parents tab is used to define the relationships that an attribute has with its parents. In the Attribute parents grid, you can change the Relationship type and Relationship table from their respective lists. Click on each to view the lists.

The Description box below the Attribute children grid puts the Relationship type in context to make it easier to understand. The blue text is the actual relationship.

From the Parents tab you can access the Add Parent Attributes dialog box when you click Add. This dialog box allows you to define a new parent. The Parent candidates box lists the available attributes obtained from the schema hierarchy relationships, defined during project creation. To add new parents, select an attribute under Parent candidates, and use the arrow button to move it to the Selected parents box.

Display tab

An attribute can be displayed directly in a report or in the object browsing window. The Display tab allows you to define whether the attribute is displayed in one or both formats.

The Available forms grid lists the forms defined for the selected attribute. Select a form from Available forms and click the arrow buttons to move it to the Report display forms box and/or the Browse forms box. All attribute forms selected for report display will be included in any report.
which uses the related attribute. All attribute forms selected to be included in browsing are displayed in the Data Explorer when the user browses the related attribute.

Use the up and down arrows to order the forms within the attribute. This determines in what order the attribute forms will display in reports and browsing.

**Note:** These display settings can be overridden when the user creates Reports using the attributes.
CHAPTER 10

Attribute Creation Wizard

What is it?
The Attribute Creation Wizard allows you to create multiple attributes quickly and easily.

How do I access it?
You can access the Attribute Creation Wizard by choosing Attribute Creation Wizard from the Schema menu.

What can I do with it?
The Attribute Creation Wizard helps you to quickly:
• Create multiple attributes
• Identify the source tables of the columns used in the expression
• Define a new child for an attribute
• Define attribute display properties
• Define attribute naming conventions
• Define column datatypes relevant to creating attributes
• Create compound attributes

For information on completing one of the above tasks, refer to the following How do I...? topic:
Attributes
Before you begin using the Attribute Creation Wizard, you need to be familiar with:

- Attribute forms
- Lookup tables—what they are and why you use them
- Parent and child relationships
- Your schema
- Compound attribute keys—what they are, and why you use them
- Column names for the temporary tables for SQL generation

For more information on the above topics, refer to the following Concept topic:

Attributes

**Attribute Creation Wizard layout**

The Attribute Creation Wizard is a series of pages which allow you to create multiple attributes. Use the wizard as a quick and easy way of creating more than one attribute at a time. The wizard makes decisions for you to simplify the attribute creation process. To make any modifications to the attributes after you create them in the wizard, use the Attribute Editor.

The Attribute Creation Wizard has the following pages:

- Introduction
- Attribute Creation Rules
- ID Column Selection
- Description Column Selection
- Lookup Table Selection
- Compound Attribute Definition
- Relationship Definition
- Finish

**Introduction**

The Introduction page provides you with a summary of what you can do in the Attribute Creation wizard. You can access the Attribute Creation Rules dialog box from this page by clicking Define Rules.
Attribute Creation Rules

The Attribute Creation Rules page allows you to define rules for creating attributes, such as displaying datatypes, naming new attributes, and setting naming conventions.

The Column data type box allows you to select the column data types available as Attribute ID columns. You can only select the data types for ID columns, and all data types are potential columns for descriptions.

The Attribute name box has rules you can assign to attribute names that are being created. Click the check boxes to see the Attribute name example dynamically change to reflect the rules selected.

Use the Warehouse search box to set naming conventions, which help MicroStrategy Desktop locate your Warehouse objects. Enter text to accompany ID column, Description column, and lookup table (for example, ID, Desc, or Lookup). For example, MicroStrategy Desktop looks for ID when trying to locate an ID column.

---

**Note:** The naming convention fields can be left empty.

---

ID Column Selection

On the ID Column Selection page you can select columns that are the IDs of the new attributes. The Available columns grid lists the columns which are available for creating attributes. Use the arrow buttons to move selected columns to the Attributes grid. The Attributes grid lists the new attribute names and their associated column IDs.

On this page you can also create Compound Attributes, which are attributes whose key is made up of more than one column in the same table. Clicking on the Compound Attributes button changes the page to open the grid for compound attributes. The Add button opens the New Compound Attribute dialog box where you select the columns to be part of the compound attribute. The Remove button allows you to delete the compound attributes.

Description Column Selection

The Description Column Selection page includes a grid with columns for the attributes and their descriptions. Use the list under Description Column Name to select a column to represent the definition for each attribute.
Lookup Table Selection

The Lookup Table Selection page includes a grid with columns for the attributes and their lookup tables. Use the list under Lookup Table Name to select a lookup table for each attribute.

Compound Attribute Definition

The Compound Attribute Definition page displays only if you created compound attributes on the ID Selection page earlier in the wizard.

Use this page to select the ID lookup table in which the ID is located and to add description columns and their lookup tables. The Compound Attributes box lists the compound attributes you have created. Use the ID Lookup Table list to change the lookup table for each attribute.

The Description Columns grid lists the description columns and their lookup tables. Use the list under Lookup Table to select a new lookup table for each description column. The Add button lets you define new description columns. Clicking Add opens the Description Columns dialog box, where you can select multiple columns to act as description columns.

Relationship Definition

The Relationship Definition page allows you to define children for each attribute when you click the Add button. Use the Select Children Attributes dialog box to select the child attributes. The Children grid lists the respective child attributes for each attribute that is selected. The Relationship type list offers options of one-to-one, one-to-many, and many-to-many. Click Remove to delete any selected child attributes.

If there are no children in the data model for the attribute, an error displays: No child candidates found for this attribute. This means that no possible relationship was found based on the ID columns selected for this attribute and the other attributes defined in this wizard.

Finish

The Finish page is the signal that you have successfully created your attributes. This page also gives you the opportunity to review the list of attributes you have created and make any changes before committing.
CHAPTER 11

Fact Editor

What is it?

The Fact Editor allows you to create a new fact or modify existing fact properties. For example, you can identify the column where the fact is stored, or you can view a set of attributes that represent the lowest level of detail at which the fact exists in the warehouse. In addition, you can assign a level extension so you can report a fact at a different level from which it is stored.

How do I access it?

You can access the Fact Editor from the File menu by choosing New then Fact.

What can I do with it?

Through the Fact Editor, you can:

• Create a new fact
• Add or modify a fact expression
• Delete a fact expression
• Select the tables to be mapped to the expression
• Choose to have the system automatically select and map all available tables to the expression
• Define a column alias for the column used for temporary table SQL generation
• Define the column alias for new facts, such as name, datatype, and byte length

For information on completing one of the above tasks, refer to the following How do I...? topic:

Facts
Before you begin using the Fact Editor, you need to be familiar with:

- Your schema
- The column names you want to use for temporary tables
- The following terms:
  - Byte length
  - Bit length
  - Precision
  - Scale
  - Time Scale
- Extensions—what they are, and why you use them

For information on the above items, refer to the following topic in the Concepts section:
Facts

**Fact Editor layout**

The editor has the following tabs:

- Definition
- Column alias
- Extensions

**Definition tab**

The Definition tab allows you to add, modify, or delete a fact expression. In addition, you can map the expression to the selected tables. From this tab you can also access the following dialog boxes:

- **Create a new fact expression**: Allows you to add an expression to the fact definition.
- **Modify a fact expression**: Allows you to modify the selected fact expression.
The formulas that define the fact calculations appear in the **Fact expression** box. The options for the mapping method appear in the **Mapping method** box. When you select a **Source table**, the facts contained in that table display under **Available columns**. The tables with the check box selected contain the highlighted expression.

- When you access the **Fact Editor** from either the **File** menu or **New Object** button in the main toolbar, the **Expressions** tab is blank. Click **New** to add a fact expression to the fact definition.
- When you access the **Fact editor** by right-clicking or double-clicking a fact object from the object viewer, the **Expression** tab is populated with information and all the buttons are fully functional.

**Column alias tab**

Use the **Column alias** tab to create an alias for the column used in the temporary table, accessed during SQL generation. On this tab is also a list of properties for the selected column, including **Name** and **Datatype**.

From this tab you can also access the **Column Editor - Column Selection** dialog box by clicking **Modify**. Use the Column Selection dialog box to change the column used for temporary table SQL generation.

The name of the column used for temporary table SQL generation appears under the **Available columns**. The detailed column properties display under **Column properties**.

**Extensions tab**

The **Extensions** tab lets you create, modify, or delete the fact extensions defined for a particular fact. Clicking **New** opens the **Level Extension Wizard** which helps you to quickly create a new fact extension.

The name of the fact extension and the text that describes the fact extension in terms of its purpose and usage appear under the **Extension name** and **Extension description**. When you highlight an extension name, the extension properties and extended fact level attributes change. Highlight an extension name and click **Modify** to edit the selected fact extension through the **Level Extension Wizard**. Refer to the Level Extension Wizard in chapter 19 of this section for more information.

When you initially access the **Extensions** tab, the only option available to you is the **New** button. Click **New** to open the **Level Extension Wizard**. Once you navigate through the wizard, the tab is populated and all buttons are fully functional.
Fact Editor toolbar and menu options

The following Fact Editor specific toolbar and menu options are available:

<table>
<thead>
<tr>
<th>Location and Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>Open the Create a new fact expression dialog box.</td>
</tr>
<tr>
<td>New Fact Expression</td>
<td>Opens the Level Extension Wizard.</td>
</tr>
<tr>
<td>New Fact Extension</td>
<td>Changes the selected object (expression or extension)</td>
</tr>
<tr>
<td>Modify</td>
<td></td>
</tr>
<tr>
<td>Tools</td>
<td></td>
</tr>
<tr>
<td>Show Fact Entry Level</td>
<td>Opens the Fact Level dialog box.</td>
</tr>
</tbody>
</table>
CHAPTER 12

Fact Creation Wizard

What is it?
The Fact Creation Wizard allows you to create multiple facts quickly and easily.

How do I access it?
You can access the Fact Creation Wizard by choosing Fact Creation Wizard from the Schema menu.

What can I do with it?
Through the Fact Creation Wizard, you can:

• Create multiple facts
• Define fact naming conventions
• Select the column data types to be used as facts

For information on completing one of the above tasks, refer to the following How do I...? topic:

Facts

What should I know before I use it?
Before you begin using the Fact Creation Wizard, you need to be familiar with:

• Your schema, so you know what columns are facts, and what information is in each table
Fact Creation Wizard layout

The Fact Creation Wizard allows you to create multiple facts. The wizard makes decisions for you to simplify the fact creation process. To make any modifications to the facts after you create them in the wizard, use the Fact Editor.

The wizard has the following pages:
- Introduction
- Fact Creation Rules
- Column Selection
- Finish

Introduction

The Introduction page provides you with a summary of what you can do in the Fact Creation wizard. You can access the Fact Creation Rules dialog box from this page by clicking Define Rules.

Fact Creation Rules

The Fact Creation Rules page allows you to define rules for creating facts, such as displaying datatypes and naming new facts.

The Column data type box allows you to select the column data types available as Fact ID columns. You can only select the data types for ID columns, and all data types are potential columns for descriptions.

The Fact name box has rules you can assign to fact names that are being created. Click the check boxes to see the Fact name example dynamically change to reflect the rules selected.

Column Selection

On the Column Selection page you can select columns for use in creating facts. The Available columns grid lists the columns which are available for creating facts. Use the arrow buttons to move selected columns to the Facts grid. The Facts grid lists the new fact names and their associated column names.
Finish

The Finish page is the signal that you have successfully created your facts. This page also gives you the opportunity to review the list of facts you have created and make any changes before committing.
Hierarchies

What is it?
The Hierarchy Editor allows you to modify hierarchies by adding and removing attributes, as well as setting specific options.

How do I access it?
You can access the Hierarchy Editor from the File menu by choosing New then Hierarchy.

What can I do with it?
Through the Hierarchy Editor, you can:

• Create a hierarchy
• Add attributes to a hierarchy
• Remove attributes from a hierarchy
• Define attribute relationships within the hierarchy
• Set attribute locking properties
• Set hierarchy display properties
• Add a user hierarchy to the system hierarchy

For information on completing one of the above tasks, refer to the following How do I...? topic:
Hierarchies
What should I know before I use it?

Before you begin using the Hierarchy Editor, you need to be familiar with the following:

• The attributes in the hierarchy
• The relationships between these attributes
• The following objects:
  ◊ hierarchies
  ◊ filters
  ◊ attributes

For information on the above topics, refer to the following topic in the Concepts section:
Hierarchies

Hierarchy Editor layout

The editor has the following sections:

• Hierarchy Attributes grid
• Browse Attributes tab
• Element Display tab
• Filters tab
Hierarchy Attributes grid

The Hierarchy Attributes grid allows you to add or remove an attribute and its related properties. This grid shows the attribute name, entry point, element display property, and the associated filter.

- **Attribute name**: The name of the attribute.
- **Entry point**: If checked, the selection was made to use the attribute as an entry point in the browse sequence. This can be changed on the Element Display tab.
- **Element display**: If the field is set to Locked, the elements do not display in the Data Explorer. If the display has been set to a limit, the field shows the limit such as Limit 9 or Limit 55. The limit means that the number of elements displayed at one time is restricted to the number entered. The element display can be changed on the Element display tab.
- **Filter**: Indicates a filter has been added to the attribute. This can be changed on the Filters tab.

From this section you can choose the attributes you wish to add to the hierarchy. Click **Add** to open the Select Objects dialog box and select the attributes from the Available objects box.

From this section you can also remove attributes that were previously added to the hierarchy by clicking the **Remove** button.

When you select an item in the grid, the entire row is highlighted. The tabs below the grid are directly related to the highlighted row. For example, if the grid contains the attributes Year, Month, Week, and Day, and only Week is highlighted, changes that are made in the tabs affect only the Week row.

At the bottom of the Hierarchy dialog box is the option **Use as a drill hierarchy**. By selecting this checkbox, you indicate that drilling is allowed on the hierarchy. If this checkbox is not selected, you cannot drill on the hierarchy in a report. When setting the hierarchy as a drill hierarchy, you establish a drill path for a report containing any of the attributes found inside this hierarchy.

Browse Attributes tab

The Browse Attributes tab defines the attribute children that are associated with a hierarchy attribute. For example, if Week and Month are the browse attributes defined for the hierarchy attribute Year, then from the Data Explorer you can navigate to Week and Month from Year. If only
Month is designated as a browse attribute, you can navigate only to Month from Year and cannot view the Week directly from Year. You can order the browse attributes within the attribute by using the up and down arrows. You must highlight an attribute from the Hierarchy Attribute grid before you can use this tab. The browse attributes you select are added to the attribute that is highlighted in the Hierarchy Attribute grid.

From this tab you can also access the Browse Attributes dialog box. When you click Add, the Browse Attributes dialog box displays and allows you to select attributes you wish to add to the hierarchy.

Element Display tab

The Element Display tab defines how the attribute is set; it can either be locked, unlocked, or have a limit set on it.

You must highlight an attribute from the Hierarchy Attribute grid before you can use this tab. With the attribute selected, from the Browse Attributes tab, select either locked, unlocked, or limit for the selected attribute. If Limit is selected, you must type in a number to restrict the number of returns for that selected attribute. Select Use as entry point in browse sequence to tell the Data Explorer where to start browsing the hierarchy.

- **Locked**: No elements of the attribute are shown in the Data Explorer. For example, if the attribute Year is locked, no elements for Year display in the Data Explorer when Year is expanded.

- **Unlocked**: All elements of the attribute are shown in the Data Explorer. For example, if the attribute Year is unlocked, all elements of Year (such as 1994, 1995, and 1996) display in the Data Explorer when Year is expanded.

- **Limit**: Incrementally retrieves the number of elements set for the attribute. For example, if the limit for the attribute Year is set to one, the years 1994, 1995, and 1996 are retrieved incrementally as they are requested.

Filters tab

On the Filters tab you can add filters which define the conditions or criteria for determining the attribute result set. For example, 1997 for Year, or Stores in the Northeast for Stores. Click Add to open the Select Objects dialog box. This dialog box allows you to choose the filters you wish to add to the attribute.
You must highlight an attribute in the **Hierarchy Attribute** grid before you can use this tab. When you select the filter, it is added to the highlighted attribute.

**Note:** The filter(s) added or associated to an attribute need to make sense for that attribute; otherwise, no data is retrieved when you browse the attribute. This is your responsibility as the project designer because you are the only one knowing the nature of the data.
Level Extension Wizard

What is it?
The Level Extension Wizard allows you to quickly create extensions to extend or lower fact levels.

How do I access it?
You can access the Level Extension Wizard from the Fact Editor. On the Extensions tab click New or Modify.

What can I do with it?
Through the Level Extension Wizard, you can:
• Lower the fact level
• Extend the fact level
• Disallow the fact level
For instructions on using the Level Extension Wizard, see the Facts How Do I section.

What should I know before I use it?
Before you begin using the Level Extension Wizard, you need to be familiar with:
• facts in the project
• the project schema
For more information, see the Facts Concept section.
Level Extension Wizard layout

The Level Extension Wizard is made up of screens which allow you to lower, extend, or disallow fact levels.

- Introduction screen
- General Information screen
- Extended Attributes screen
- Join Type screen
- Join Attributes Direction screen
- Allocation screen

**Introduction screen**

The *Introduction* screen provides you with a summary of what you can do in the Level Extension wizard. If you feel you do not need the information contained on the screen, select *Don’t show this window again*.

**General Information screen**

The *General Information* screen is where you choose the type of extension or disallow. In the *Name* box, enter a name for the extension or disallow. The *Description* box is an optional field where you can describe the extension or disallow and its use. Click *Next* to proceed through the wizard.

**Extended Attributes screen**

On the *Extended Attributes* screen you can select attributes for use in creating the extension definition. The *Available attributes* grid lists the attributes which are available for extending a level or disallowing access. Highlight the attributes you want included in the definition and click the arrow buttons to move the selected columns to the *Selected attributes* grid.
Join type screen

The Join Type screen allows you to select the attributes for creating a join. Click in the boxes next to each attribute you want included in the join. Check marks will appear in each box you select. Click Next to proceed through the wizard.

Join Attributes Direction screen

The Join Attributes Direction screen allows you to specify the join direction for the different extension attributes. This screen asks at what level do you want to join the attributes.

You can change the direction of the join by clicking in the Join Against column. When you click on the arrows displayed in the Join Against column, they change direction. A straight arrow builds a join only to the attribute, while the bent arrow builds a join to the attribute and children.

Allocation screen

Use the Allocation screen for entering an allocation expression. You can place operations on attributes and facts to change the definition of a fact in a level extension. The expression box in grayed out, but you can make it active by selecting Specify an allocation expression. Click on each operator to add them to the expression. The Clear button removes the entire expression. When you have entered an expression, click Validate to ensure it works.

Finish screen

The Finish screen is the signal that you have successfully created your level extension definition. This screen also gives you the opportunity to review the options you have created and make any changes before committing.
**What is it?**

Partition mapping is powerful and easy to use. With the Partition Mapping Editor you can create or modify metadata partitions, or view warehouse partitions.

**How do I access it?**

You can access the Partition Editor from the File menu by choosing New then Partition.

**What can I do with it?**

Through the Partition Mapping Editor, you can:

- Create a new metadata partition
- View an existing warehouse partition
- Add tables to an existing partition
- Remove tables from a partition
- Define a data slice
- Add an attribute qualification to a data slice
- Set the partition mapping logical size

For information on completing one of the above tasks, refer to the following How do I...? topic:

Partitions
**What should I know before I use it?**

- What tables have been included in the Warehouse catalog
- The data model
- The schema

For information on the above topics, refer to the following topic in the Concepts section:
Data Modeling

**Partition Mapping Editor layout**

The Partition Mapping Editor has the following screens:

- Metadata Partition Mapping
- Partition Tables Selection
- Data Slice Editor

**Metadata Partition Mapping screen**

From this screen you can add tables to the partition mapping by clicking Add. You can set the Partition mapping logical size by clicking the arrows at the bottom of the screen. The table’s logical size is used by the engine to pick the best, or smallest, table when accessing data.

The Data slice definition box lists the data slices already defined for the partition mapping, if any exist. After adding partition tables, click Define to open the Data slice editor, where you can add new data slices. You must select tables before you can create a data slice. If the Define button is not available, then you need to select tables.

The Partition level attributes box lists the lowest level attributes that represent the partition. These are the attributes you select when creating the data slice in using the Define button. To make any changes to attributes in this box, right-click and select Edit Attribute from the menu.
Partition Tables Selection screen

You can open this screen by clicking Add in the Tables box of the Metadata Partition Mapping screen. On the Partition Tables Selection screen, select the tables you want included in the partition mapping and click the arrow to move them into the partition mapping. Click OK to save the tables, and the screen closes. The selected tables are displayed on the Metadata Partition Mapping screen.

Data Slice Editor screen

After selecting partition tables, you must create a data slice for each table. You can access the Data slice editor when you click Define on the Define Metadata Partition Mapping screen. Use the Object browser to select attributes for the data slice definition. The Location list allows you to access attributes saved in different folders.

You can drag an attribute to the Data Slice Definition box, or double-click on the definition, which will open the Attribute Qualification box. Make selections from the lists described below:

- Choose an attribute: Select or change the attribute.
- **Qualify on:** Select the ID, Desc, or Element for qualifying on the attribute. The default is Elements.

---

**Note:** Once you have decided which attribute you want to qualify, you must choose the attribute form to qualify on. If you choose to qualify on the elements, the Object browser defaults to displaying the selected attributes elements. If you select a different form, such as ID or Desc, the Object browser defaults to displaying hierarchies.

- **Operator:** Depending on whether you are qualifying on ID, Desc, or Element, you will see different options in the Operator list. If you are qualifying on elements, the selections In List or Not in List display. If you are qualifying on ID or Desc, the Operator lists options for comparison operators (for example, greater than, less than). You may choose to compare Values or Expressions, and to enter or select the actual value or expression.
- **Element list:** This option displays only when you are qualifying on elements. Click Add to include specific elements to the qualification. These are form elements for qualifying on the selected attribute.
Print Project Schema

What is it?
Print Project Schema allows you to print the schema according to options you set.

How do I access it?
You can access the Print Project Schema function from the Schema menu.

What can I do with it?
Through the Print Project Schema function, you can:
• Select specific parts of the schema to print
• Adjust the format of the printouts
• Include project, report, template, and filter names
• Insert images
• Preview your work before sending it to print
• Set printer properties

What should I know before I use it?
Before you begin using the Print Schema Project function, you need to be familiar with:
• Your schema, so you know what parts to print
• The final look of your printout, so you can adjust the formatting options accordingly
• Your printer and its capabilities
Print project schema layout

The Print Project Schema function is a dialog box. It allows you to print all or select parts of the project schema in a format of your choosing.

After you access the Print Project Schema from the Schema menu, a dialog appears with a list of seven checkboxes corresponding to a section of the schema available to print. The options are:

- **General/Connection**: Displays the project name, project description, project source information, data warehouse information, connection information, the version of MicroStrategy Desktop, and the date of printing.
- **Selected Warehouse Tables**: Shows table name, prefix, and suffix.
- **Fact Definitions**: Describes the facts used in the project.
- **Attribute Definitions**: Details the attributes included in the project.
- **Table Catalog - Physical View**: Lists table names, columns, and type from the perspective of the database system.
- **Table Catalog - Logical View**: Displays table names, objects, and type from the perspective of the business intelligence or decision support project.
- **Include Project Statistics**: Offers a rundown of the total number of tables and mappings, columns, facts, and attributes.

Click the options you want to include in your printout. You can click OK to go directly to the print dialog.

Page Setup

The Page Setup dialog enables you to adjust the formatting of your printout. It has three tabs, Page, Margin, and Header/Footer.

**Page tab**

On the Page tab, you can set the portrait or landscape orientation of your printout, scale your printout to fit on one page, and choose between the automatic page numbering feature or starting the pagination at a specific number.
**Margin tab**

On the Margin tab, you can center the page vertically or horizontally. You can also set the top, left, right, and bottom margins by the hundredths of an inch. As you enter the margin adjustments, a thumbnail of the print project reflects your activity by showing a red band at the place of your change.

**Header/Footer tab**

On the Header/Footer tab, you can have the most impact on the design of your project schema printout. The **Edit** list allows you to toggle between header and footer as you work on each. The **Line Width** list lets you choose the thickness of the lines that will delineate the header and footer regions from the main body of the printout. Below the list boxes is a toolbar with the following features:

- **Font**: Allows you to change the font, font style, and size of the text presented in your printout. You can also add underlining and strikeout effects, in addition to changing the color of the text.
- **Insert Page Number**: Permits you to display the page number in the header or the footer.
- **Insert Number of Pages**: Discloses how many pages are in the printout. Used with the Insert Page Number feature, you can see the current page in terms of the total number of pages, for instance, page 13 of 21.
- **Insert Date**: Records when the printout was made.
- **Insert Time**: Identifies precisely the time the printout was sent to the printer.
- **Insert Project Name**: Allows you to include the project name in the header or the footer.
- **Insert Report Name**: Grayed out and not available for Print Project Schema.
- **Insert Template Name**: Grayed out and not available for Print Project Schema.
- **Insert Filter Name**: Grayed out and not available for Print Project Schema.
- **Insert Filter Details**: Grayed out and not available for Print Project Schema.
- **Insert Image**: Lets you import graphics of any format supported by your printer.
Below the toolbar are three blank fields: left, center, and right. These fields allow you to justify the placement of information in the header and footer. By first clicking in a field, you establish an insertion point for information added from the toolbar. Clicking an icon on the toolbar inserts the name of the feature in the field where you first clicked.

Once you have finished designing the header or the footer, go to the other one in the Edit list. Your previous design is contained in the drop-down list beneath the three fields.

As you insert elements into the header and footer of your project schema printout, a thumbnail on the right side of the tab screen indicates whether you are working in the header or the footer.

All tabs have the Print, Print Preview, and Options buttons available.

**Print**

This button launches the print dialog where you can select a printer, determine a page range to print, and enter the number of copies to be printed. You can also select what to print from the Print What list. In this instance, the list defaults to Schema Details, the basis for your project schema printout.

The Print Preview and Page Setup buttons are available.

**Print Preview**

Clicking this button presents a sample of what your final printout will look like. This screen has several functions available as buttons presented beneath the title bar. They are:

- **Print**: Takes you to the print dialog where you can send the job to the printer.
- **Setup**: Opens the Page Set Up dialog where you can fine tune the formatting of the job.
- **Previous**: Goes to the previous page in the job.
- **Next**: Goes to the next page in the job.
• **Two-Page:** Displays the print job two pages at a time.

• **Zoom In:** Tightens your view of the screen incrementally. The button grays out when you reach its maximum range.

• **Zoom Out:** Loosens your view of the screen incrementally. The button grays out when you reach its maximum range.

• **Previous Page Field:** Returns your view to the previous section in the print job.

• **Next Page Field:** Advances your view to the next section in the print job.

• **Close:** Terminates the Print Preview function and returns you to the desktop.

**Options**

This button allows you to set the properties for your printer. You can have the printer collate, draw letter-size paper from a particular tray, print at a certain resolution, employ watermarks, and use other settings that add interest to your printout.
CHAPTER 17

Project Creation Assistant

What is it?
The Project Creation Assistant allows you to create new projects.

How do I access it?
You can access the Project Creation Assistant from the Schema menu when you select Create New Project.

Note: For this release, a project can be created only on a Direct (2-Tier) project source. The server is not needed at all to create a new project. You need a Direct project source configured before using the Project Creation Assistant.

What can I do with it?
Through the Project Creation Assistant, you can:

• Name your project
• Select the metadata repository to use for your project
• Select a project creation script
• Select tables from the warehouse catalog
• Define fact and attribute rules
• Create facts
• Create attributes

For information on completing one of the above tasks, refer to the following How do I...? topic:
Projects
What should I know before I use it?

Before you begin using the Project Creation Assistant, you need to know:

• The existing project sources
• A valid username and password for the project source
• The name of the Destination project you want to create
• Location for the Event log
• Location for the Corrupted object log
• Location for the Statistics log
• The Warehouse Database Instance you plan to use
• The password and username for the warehouse
• The database type and version you plan to use
• Which tables you want to use in your new project
• The datatype you want to use to identify and format facts
• What facts you want to include in your project
• The datatype you want to use to identify and format attributes
• What attributes you want to create
• What description column name appears for each attribute
• What a child is
• What child attributes you want for each attribute
• What attribute-child relationship you want to assign

For more information, see the following Concept section:
Projects

Project Creation Assistant Layout

The assistant has the following screens:

• Create project
• Select tables from the warehouse catalog
• Create facts
• Create attributes
Create project

With the General tab of the New Project dialog box, you can name the new project and select the metadata repository that is used for the new project. Or, you can create a new Project Source.

The Advanced tab enables you to select or edit a project creation script. In most cases, the default script is the best one to use and there is no need to make any modifications to this tab.

From this dialog box, you can also access the Login dialog box when you click on the OK button. This login is the gateway to the project source selected on the General tab. If your login is not accepted, you are not authorized to create projects in the specified data source and you cannot proceed past this step in the wizard.

Select tables from the warehouse catalog

The first screen of this wizard allows you to select or edit the project warehouse database instance, or create a new one. If you create a new database instance, you must provide the database type and version.

Click OK to view the Warehouse Catalog Browser. The Cancel button returns you to the Project Creation Assistant welcome screen.

The Warehouse Catalog is divided into two boxes: Tables available in the warehouse and Tables being used in the project. The Tables available in the warehouse list displays all tables in the selected data warehouse. The Tables being used in the project list displays the tables selected for inclusion in the project. The first time you view this screen, the right-hand list is empty because no tables have been selected.

To read the warehouse catalog and populate the left-hand list, click the exclamation point icon in the toolbar. Select the tables under Tables available in the warehouse and click the right button to move them to the Tables being used in the project list.

Click Save and Close to return to the welcome screen of the project creation assistant.

Create facts

The first screen of the Fact Creation Wizard is the welcome screen. From this screen you can access the Fact Creation Rules dialog box by clicking Define Rules. This dialog box allows you to select the column data type and assign rules for fact creation.
Click **Next** to view the **Column Selection** screen. The left-hand box contains a list of column names which you use to create fact columns. Only the columns of the data types specified in the **Fact Creation Rules** dialog box display. The right-hand panel lists the fact columns chosen for the project. The fact names are formatted according to the rules specified in the **Fact Creation Rules** dialog box. Select column names and click the right arrow to create facts.

Click **Next** to view the **Finish** screen. A summary of the facts you created appears in the center of the screen. To add more facts or delete facts you created, click **Back** to return to the appropriate screen in the wizard to make the necessary adjustments. Click **Finish** to complete the Fact Creation Wizard.

### Create attributes

The first screen of the Attribute Creation Wizard is the introduction screen. From this screen you can access the **Attribute Creation Rules** dialog box by clicking **Define Rules**. This dialog box allows you to select the column data type and assign rules for attribute creation.

Click **Next** to view the **ID Column Selection** screen. The left-hand box contains a list of column names from which to choose the attribute columns. Only the columns of the data types specified in the **Attribute Creation Rules** dialog box are displayed. Any columns already selected are recommendations to help you. The right-hand panel lists the attribute columns chosen for the project. Each column you select becomes an attribute. Click the arrow buttons to select columns to be attributes. The attribute names are formatted according to the rules specified in the **Attribute Creation Rules** dialog box.

Click **Compound Attributes** then click **Add** to create a compound attribute. Compound attributes are attributes whose key or ID has a dependency on multiple columns, rather than just one. When choosing columns to build a compound attribute, all columns must exist in at least one table. Each compound attribute you create displays in the Compound Attributes box. You can expand each compound attribute by clicking on the plus (+) sign to view the columns that make up the attribute.

Click **Next** to view the **Description Column Selection** screen. This screen allows you to choose what column name appears when the Description form is selected for an attribute. For example, for the attribute Customer, Customer_Name is mapped to the description form. When the description form of the Customer attribute is chosen for the report, Customer_Name appears in the report.

Click **Next** to view the **Lookup Table Selection** screen. From this screen you can define the lookup table for each attribute.
Click **Next** to view the **Relationship Definition** screen. From this screen you can define the parent-child relationships and the relationship for each attribute. The default relationship is one-to-many.

Click **Next** to view the **Finish** screen. This screen allows you to view the summary of attributes that you created, modified, or removed.

**Exiting the Project Creation Assistant**

You can close the assistant by clicking **OK** on the Project Creation Assistant screen.
**What is it?**

The **Project Duplication Wizard** allows you to make copies of an existing MicroStrategy 7.x project.

**How do I access it?**

You can access the **Project Duplication Wizard** from the **Schema** menu by choosing **Duplicate Project**.

**What can I do with it?**

Through the **Project Duplication Wizard**, you can:

- Create a backup of the project in another repository
- Create a new project with the same schema
- Create a copy of the project for development purposes before releasing to production
- Point the project on a Server to a different repository
- Select the source project location
- Select the location for the duplicated project
- Set process options
- Set event logging and reporting options

For instructions on completing one of these tasks, see the **Project How Do I section**.
**What should I know before I use it?**

Before you begin using the **Project Duplication Wizard**, you need to know:

- Duplication is a 2-tier process (ties directly into metadata repository from the desktop)
- System requirements are the same as described in the Project Upgrade Manual
- The ODBC DSN for the source and destination repository
- New objects created while the duplication is in process will not be copied, if the project is loaded in a server
- Login with administrator privileges in both the destination and the source repository, to be allowed to perform this operation
- Project sources and locations
- Duplicate project destination
- The name for the destination project
- Whether to duplicate only schema objects, or all objects
- Whether to log only errors, or all events
- Log files locations
- A valid username and password for the metadata repository
- The password and username for the warehouse

For more information, see the Project Concepts section.

**Project Duplication Wizard Layout**

The Project Duplication Wizard allows you to back up projects that are currently in development prior to general release or create a new project based on the schema of the original. Use the wizard as a quick and easy way to keep the project in the same or another repository. The wizard assists you in defining the parameters of the project to be duplicated, thus simplifying the project duplication process. To modify a project after you duplicate it in the wizard, use any editor specific to your purpose.
The wizard has the following pages:
• Select the source of a project location
• Select a project to duplicate
• Select a location for the duplicate project
• Create a destination project
• Set process options
• Set viewing options
• Set event logging options
• Review process summary

Introduction

The Introduction page provides you with a summary of what you can do in the Project Duplication wizard. You can choose to remove this page for the future by selecting Don’t show this window again.

Select the Source of a project location

The Source project location page allows you to select a source project from the Available Direct Project Sources list. Or, you can create a new project source by clicking on the New button. For more information on creating a project source see Create a project source in the How Do I section.

Enter a valid Login ID and Password (with administrator privileges) to proceed with the project duplication. If your login is not accepted, then you are not authorized to duplicate projects in the specified project source and you will not be allowed to proceed past this step in the wizard. Ask your Administrator for permission to duplicate projects.

Click Next to select the project source.

Select a project to duplicate

Under Available projects, you can select the project you would like to duplicate. The selected project displays in the Selected project box. Notice the Project Description box, which describes the existing project you have chosen to duplicate. The Project Description is helpful in determining which project to duplicate.

Click Next to select the location for the duplicate project.
Select a location for the duplicate project

You can select the destination for the duplicate project from the **Available Direct Project Sources** list. Or, you can create a new project source by clicking **New**. For more information on creating a project source see Create a project source in the How Do I section.

Select the type of **Authentication** (for example, NT login or other) and enter the Login ID and Password, depending on your selection. You will need administrator privileges to proceed with the duplication. If your login is not accepted, then you may not be authorized to duplicate projects in the specified project source and you will not be allowed to proceed past this step in the wizard.

Click **Next** to enter the destination project name.

Create a destination project

On this page, enter a name for the duplicate project in the **Destination project name** box. You may enter a description of the duplicated project in the **Destination project description** box (optional). The description is helpful for future reference for you and others to determine why the project was created.

Set process options

You can choose to duplicate only the **schema objects** of the project, or **all objects**. If you duplicate only the schema objects, you will need to create other objects later, such as filters, metrics, and reports.

When a user being duplicated already exists in the source destination project, you can choose to add/merge user properties, leave the existing user properties, or overwrite the existing user properties with properties from the source.

Select one of the three options and click **Next** to select viewing options.

Set viewing options

If you want the event log visible, select **View event log concurrently**. By selecting this, the Event types become available for your selection. You can choose to view **Log errors and warnings only**, or **Log all events**.

Click **Next** to select logging options.
Set event logging options

On this page, indicate how and where you would like events logged. By selecting Event log, you indicate that an event log be generated. You can choose to Log errors and warnings only, or Log all events. Click Browse to choose the location for the event log.

You can also choose to have a Corrupted objects log and Statistics log by selecting their respective boxes. Click Browse to choose the location for these files.

If you do not select any log options, no log files will be generated should the product encounter an error.

Click Next to view summary of project duplication.

Review process summary

The summary page displays the selections you made throughout the wizard pages. Click Finish if you approve of the options selected, or click Back to make necessary changes.

Finish project duplication

This page tracks the duplication process as it happens. Each event in the duplication process is listed with its corresponding status. You can use the Pause and Cancel buttons in the Duplicate Controls pane to interrupt the duplication process. Two progress bars indicate the current status of items transferring to the duplicated project and the time elapsed in the duplication process. When the duplication process finishes, click Exit to conclude your activity in the Project Duplication Wizard.

The Finish page signals that you have successfully created your duplicate project. Click OK to return to the Desktop.
CHAPTER 19

Table editor

What is it?
The Table Editor allows you to view objects and columns in each table.

How do I access it?
You can open the Table Editor by double-clicking a table in the desktop Table folder.

What can I do with it?
Through the Table Editor, you can:
• View objects contained in each table
• View physical columns contained in each table
• Calculate table key

What should I know before I use it?
Before you begin using the Table Editor, you need to be familiar with:
• The schema
• The attributes and facts
For information on the above topics, refer to the following topic in the Concepts section:
• Attributes
• Facts
Table Editor layout

You can calculate the table key in the Table Editor when you click the Calculate table key button.

The Table Editor has the following tabs:

- Logical View
- Physical View
- Partition View (only if the table is part of a partition mapping)

Logical View tab

The Logical View tab allows you to view the facts and attributes present in the logical structure of selected table. On the Logical view tab, you can establish that the key specified is the true key for the warehouse table. Use this tab to set the logical size of the table. Double-click objects (for example, facts and attributes) to open their respective editors and make changes.

A key icon displayed by the attribute name on the Logical View tab indicates that the attribute is part of the key for this table. The key is made up of the lowest level attributes. When you double-click an attribute or fact, the respective editor opens.

Physical View tab

The Physical View tab allows you to view the columns contained in the warehouse table. The datatype column displays the datatypes for each column and their size.

Partition View tab

The Partition View tab is present only on tables that are part of a partition mapping. This tab shows the list of partition mappings that the table belongs to. If the table does not belong to a partition mapping, then the tab will not appear.
Transformation Editor

What is it?

The Transformation Editor allows you to create or modify a transformation.

How do I access it?

You can access the Transformation Editor from the File menu by choosing New then Transformation.

What can I do with it?

Through the Transformation Editor, you can:

• Create a new Transformation
• Add or modify a transformation

For more information, see the Transformation How Do I section.

What should I know before I use it?

Before you begin using the Transformation Editor, you need to be familiar with:

• schema
• data
• the attributes in the hierarchy
• the relationships between these attributes

For more information, see the following section:
Transformation Editor layout

The Transformation Editor is a dialog box with the **Member attributes grid** and buttons that allow you to add, remove, and modify transformations. When creating transformations, you create member attributes, select a member table, and define member attribute expressions. The grid holds the member attributes, expression, and member tables for all changes you make to the transformation.

**Add a member attribute**

In the Transformation Editor, click Add to open the Select a member attribute dialog box. In this box, select an attribute that will become a member attribute in the transformation.

**Remove a member attribute**

To remove a member attribute from the transformation, select the attribute in the **Member attributes grid** and click Remove.

**Modify a member attribute expression**

To modify a member attribute expression, select the attribute in the **Member attributes grid** and click Modify. In the Modify an existing member attribute expression dialog box, you can specify the new expression and the table where the columns can be located.

**Define a member attribute expression**

After adding a member attribute, you are prompted to define the member attribute expression. The Define a new member attribute expression dialog box opens. Select a table from the Table list. Under Available columns, select a column and drag it to the Member attribute expression box. Click the operators to insert them into the expression and add any numbers or text as necessary.

The expression you enter is used to retrieve information for the current member attribute of the transformation. To ensure your expression is properly written, click Validate. As long as the expression is valid, you see a green checkmark. If the expression is not valid, you receive an error. Creating a valid expression defines the behavior of the member attribute for each transformation and the table containing the expression.
Select the Transformation mapping type

The Transformation mapping type defines how the transformation is mapped. A transformation can have two different mapping types: one-to-one or many-to-many. The mapping type indicates how to create the transformation, based on the data. For example, a transformation Last Year to This Year is usually a one-to-one relationship, because one day or month in last year maps to a day or month in this year. Conversely, a transformation Year to Date is usually a many-to-many relationship because there are many days or months in a year that can actually be included for this transformation.

After you create the member attributes and expressions, select the Transformation mapping type at the bottom of the Transformation editor. The default for new transformations is one-to-one.
CHAPTER 21

Warehouse Catalog Browser

What is it?

The Warehouse Catalog Browser allows you to change the tables used by the current project.

How do I access it?

You can access the Warehouse Catalog Browser from the Schema menu by choosing Warehouse Catalog.

What can I do with it?

Through the Warehouse Catalog Browser, you can:

• Modify table selection for a project
• Change the database instance for a project
• Change the catalog read mode
• Set prefixes for table names
• Remove prefixes from displayed table names
• Add new table prefixes
• Determine how to obtain table structure
• Set schema objects to map to new tables automatically
• Set to calculate sizes of logical tables automatically

For information on completing one of the above tasks, refer to the following How do I...? topic:

Warehouse
**What should I know before I use it?**

Before you begin using the Warehouse catalog you need to be familiar with:

- Your schema, so you know what columns are facts, and what information is in each table
- How to create a project

For information on the above topics, refer to the following topic in the Concepts section:

Data Modeling

---

**Warehouse Catalog layout**

The Warehouse Catalog Browser is a dialog box that allows you to view tables that have been included in the project, as well as those that are still available in the warehouse.

As you make changes in the Warehouse Catalog, you need to periodically load the updates into the Warehouse Catalog. In the Actions menu, select Read the Warehouse Catalog.

The Warehouse Catalog has the following sections:

- **Tables available in the warehouse:** Displays tables that are located in the warehouse, but have not been included in the project. You can add tables to the project by double-clicking or by selecting them and clicking >.

- **Tables being used in the project:** Displays tables that have been selected to be part of the project. You can remove tables from the project by double-clicking or by selecting the tables and clicking <.

---

**Warehouse Catalog options**

From the Warehouse Catalog Browser, you can choose to view the Warehouse Catalog options dialog box. You can access the Warehouse Catalog options dialog box by clicking Options. On this dialog box you can change the database instance, view table prefixes and structures, and map objects to tables. There are three tabs available: Catalog, View, and Schema.
Catalog tab

On the Catalog tab, you can change the database instance and catalog read mode for a project. To change the database instance, select one from the Database Instance list.

On this tab, you can customize the SQL to read the Warehouse Catalog for every platform except MS Access, which is read through ODBC. Also contained in the Read Method pane is the check box **Ignore current table space when reading from the database catalog and update using new table space.** It allows you to switch between warehouses found in different database name spaces.

Also, you can set the Catalog Read Mode to be Automatic (default) or Manual. The Automatic catalog read mode sets the catalog tables to be read as soon as the catalog browser is loaded. The Manual catalog read mode sets the catalog tables to only be read when the read catalog action is selected.

From this tab you can also access the following interfaces:

- **Database Instance Wizard.** Allows you to create a new database instance. Click **New** under **Database Instance**.

- **Database Instance Editor.** Allows you to make changes to the current database instance. Click **Edit** under **Database Instance**.

For more information, see the following topics in the Interfaces section of the Administrator Guide:

- Database Instance Wizard
- Database Instance Editor

View tab

On the View tab you can set prefixes for table names, remove prefixes from displayed table names, and add new table prefixes.

To display prefixes in table names, select **Display table prefixes in the main dialog.** All prefixes in table names are displayed, even if you add new tables to the project.

You can set MicroStrategy Desktop to automatically add a prefix to tables when you add them to the project. Select the check box and select a prefix from the **Default Prefix list.** If you want to create a new prefix, click **Modify prefix list.** In the **Table Prefixes** dialog box that opens, select a prefix and click **OK.** Or, click **Add** to create a new prefix in the **New Table Prefix** box. You can delete a prefix from the list on the **Table Prefixes** dialog box.
On the **View** tab, you can select when to view table structures. You can choose for MicroStrategy Desktop to view all table structures, including columns and their corresponding data, or just view the structure for selected tables in the warehouse catalog. For improved performance, MicroStrategy recommends selecting to view table structures for selected tables only.

**Schema tab**

On the **Schema** tab, you can select options which guide the warehouse catalog to map schema objects to new tables and calculate logical sizes for new tables.

To map schema objects automatically to new tables, select **Map schema objects to new tables automatically**. Objects in the schema automatically map to tables you add to the project.

To calculate logical table sizes automatically, select **Calculate the logical table sizes automatically**. Each time you add new tables to the project, their logical sizes are calculated for you.

**Ignore table name space setting**

This setting helps you move a project from one warehouse to another. It is a common practice to establish a secondary warehouse with less information than the primary warehouse for development and testing. Before going into production, you can move the project to point to the primary warehouse.

Most database management systems (Oracle, DB2, and others) support the concept of Table Name Space, which is a way of organizing database tables into different storage spaces. This method allows the administrator to repeat the same table name in different table name spaces. For instance, you can have LU_STORE in table name space dbo and another table LU_STORE in another table name space called admin. You now have two tables dbo.LU_STORE and admin.LU_STORE. Table Name Space provides an extra piece of information that uniquely identifies the table.

When you add tables to a project, the Warehouse Catalog Browser saves information to the appropriate Table Name Space. This may cause a problem when you migrate from a warehouse that resides in a certain Table Name Space to another warehouse in a different Table Name Space. The problem is that the Warehouse Catalog Browser interprets the table as already in the project and not found in the new warehouse. This is because the Warehouse Catalog Browser is looking for a table named dbo.LU_STORE and in the new production warehouse the table is actually stored as admin.LU_STORE. This is a common occurrence.
To solve this problem, click the checkbox option **Ignore current table name space when reading from the database catalog and update using new table name space**. You can find it in the Warehouse Catalog Options dialog under the Catalog Reading options section. If you check this option, the Warehouse Catalog Browser ignores the current table name space when it reads the catalog information. Thus, the Warehouse Catalog Browser recognizes the two tables as the same table and saves the new table name space information. This setting allows you to migrate much more easily between warehouses. If the checkbox is left blank, the Warehouse Catalog Browser defaults to identifying the table by both table name space and table name.

For more information, see the following help topics in the **How do I...?** section of this guide:

Change the Project Warehouse

Change the Project Warehouse when the Table Name Space is Different
How Do I...?

- Attributes
- Facts
- Hierarchies
- Partitions
- Print Project Schema
- Projects
- Tables
- Transformations
- Warehouse Catalog
Add an attribute form to an attribute

**Steps**

1. Double-click the attribute to which you wish to add an attribute form. The Attribute Editor opens.
2. On the Forms tab click New. The Create New Attribute Form dialog box opens.
3. Click New. The Create New Form Expression dialog box opens.
4. Select a table from the Source table box.
5. Drag one or more column names from the Available columns list to the Form Expression box.
6. Click any operator to insert it into the form expression.
7. Click Validate to determine if the form expression is valid.
9. Click OK. You are returned to the Create New Attribute Form dialog box
10. Enter a Form name and Description.
11. Under Source tables, select a table name and click Set as Lookup.
12. Select a Form category, Display type, and Default sort from the lists.
13. Click OK
14. Save the attribute.
15. From the Schema menu, choose Update Schema.

**Note**

To see your changes in the project, you must update the schema.
Create a column alias

You can create a new alias for the column used in the temporary table for SQL generation instead of accepting the default column name.

**Steps**

1. Open the attribute in the **Attribute Editor**.
2. From the **Attribute editor**, select a **Form name** and click **Modify**. The **Modify attribute form** dialog box displays.
3. On the **Column Alias** tab, click **Modify**. The **Column Editor-Column Selection** dialog box opens.
4. Click **New**. The **Column Editor-Definition** dialog box opens.
5. Enter a **Column name**.
6. From the **Datatype** list, select a datatype.
7. Depending on which datatype you select, you can define the **Byte length**, **precision**, **scale**, and **length**.
8. Click **OK**. You are returned to the **Column Editor-Column Selection** dialog box.
9. Click **OK**. You are returned to the **Modify attribute form** dialog box.
10. Click **OK**. You are returned to the **Attribute Editor**.
11. Save your changes.
12. From the **Schema** menu, choose **Update Schema**.

**Notes**

- To see your changes in the project, you must update the schema. If you modify multiple attributes, you can update the schema once, after all changes have been made.
- Depending on the datatype you choose, one of the following boxes is visible:

<table>
<thead>
<tr>
<th>Datatype</th>
<th>Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>Float</td>
<td>Bit length</td>
</tr>
<tr>
<td>Integer, Unsigned, Char, VarChar, LongVarChar, Binary, VarBin, LongVarBin</td>
<td>Byte length</td>
</tr>
<tr>
<td>Numeric, Decimal</td>
<td>Precision</td>
</tr>
<tr>
<td>Numeric, Decimal</td>
<td>Scale</td>
</tr>
</tbody>
</table>
Create a compound key for an attribute

**Steps**

1. Open the attribute in the **Attribute Editor**.
2. Select the forms which will make up the compound key.
3. From the **Edit** menu, choose **Group**.
4. A dialog box opens asking you to confirm this action. Click **Yes** to continue.
5. Save your changes.
6. From the **Schema** menu, choose **Update Schema**

**Notes**

- To see your changes in the project, you must update the schema.
- All members of a compound key must be part of the same form category.
- Only one form or form group can be set as a key. If you set a new form as the key, it removes the key from the current form.
- You cannot delete a form group while it is set as a key. You must first set another attribute form as the key.

Create a derived attribute

**Steps**

1. On the **File** menu, point to **New**, then choose **Attribute**. The **Attribute Editor** opens.
2. Under **Source table**, select a table name to view its columns.

<table>
<thead>
<tr>
<th>Datatype</th>
<th>Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time, TimeStamp</td>
<td>Time scale</td>
</tr>
<tr>
<td>Date</td>
<td>[None]</td>
</tr>
<tr>
<td>Double</td>
<td>[None]</td>
</tr>
<tr>
<td>Real</td>
<td>[None]</td>
</tr>
</tbody>
</table>
3. From the **Available columns** list, double-click a column name to place it in the **Form expression** box.

4. Click any operator to complete the expression.

5. Click **Validate** to ensure the expression is valid.

6. Under **Mapping method**, select **Automatic** or **Manual**.

7. Click **OK**. You are returned to the **Create New Attribute Form** dialog box.

8. Enter a **Form name** and **Description** for the derived attribute.

9. In the **Source tables** list, select a table name and click **Set as Lookup** to set the lookup table for the attribute.

10. Select a **Form category**, **Display type**, and **Default sort** from the lists.

11. Click **OK**. You are returned to the **Attribute Editor**.

12. Save the attribute.

13. From the **Schema** menu, choose **Update Schema**.

**Note**

To see your changes in the project, you must update the schema.

### Create a form group

**Steps**

1. Double-click an attribute. The **Attribute Editor** opens.

2. From the Attribute Editor, select the attribute forms under **Form Name** and click **Group**. The **Create an Attribute Group** dialog box displays.

3. Enter a **Name** and **Description** for the new group.

4. Under **Form Category**, select a category from the list or click **Modify**.

5. If you click **Modify**, the **Form Categories** dialog box displays. You can add a new form category by entering a **Form Category Name** and clicking **Add**. Click **OK**.

6. Click **Save and Close**.

7. From the **Schema** menu, click **Update Schema**.
Create a joint child attribute

Steps
1. From the Attribute Editor, click the Children tab.
2. Click Add. The Add Children Attributes dialog box opens.
3. Under Children candidates, select the attributes to become children, and click the right arrow button.
4. Select the Create as Joint Child check box.
5. Click OK.
6. From the Attribute Editor, you can change the child relationship by clicking Relationship type and selecting one from the list.
7. Click Save and Close.

Note: For your changes to take effect, you must first update the schema.

Create a new attribute

Steps
1. From the Folder list, click your project folder.
2. From the File menu, click New, then click Attribute. The Create New Form Expression dialog box opens.
3. From the Source table list, select a table name. Its columns display in Available Columns pane.
4. To create an expression, you can:
   ◊ drag a column name from Available columns to the Fact expression box
   ◊ click any operator to insert it into the form expression
   ◊ enter a constant such as 1 or Rushorder (in double quotes)
5. Click Validate.
7. Click OK.
8. On the Create New Attribute Form dialog box:
   ◊ Under Source tables, select a table and click Set as Lookup. If you choose manual mapping, click the checkboxes of the tables you want mapped.
   ◊ Enter a Name and Description.
   ◊ Under Category used select a form category, or click Modify to create a new one.
   ◊ Under Form format, select a display type and sorting from the lists.
9. Click OK.
10. On the Attribute dialog box, click Save and Close.

Notes:
   • For your changes to take effect, you must first update the schema.
   • When you add an expression to a form, you are only required to select an available column and move it over to the Expression box. You do not have to include any operators or parentheses, or validate the formula.

Create a new form category

When you are defining an attribute, you must create an attribute form, and can choose a form category of type ID, Description, or None. You can also create your own form categories that best describe your attribute.
Steps
1. From the Attribute Editor, select an attribute form and click Modify. The Modify Attribute Form screen opens.
2. Under Form category, click Modify. The Form categories dialog box opens.
3. Enter a Form category name and a description.
4. Click OK.
5. From the Schema menu, click Update Schema.

Note: For your changes to take effect, you must first update the schema

Create an attribute form and expression

Steps
1. Double-click an attribute. The Attribute Editor opens.
2. On the Forms tab click New. The Create New Attribute Form dialog box displays.
3. Click New. The Create New Form Expression dialog box displays.
4. From the Source table list, select a table name. Its columns display in Available Columns pane.
5. To create an expression, you can:
   ◊ drag a column name from Available columns to the Fact expression box
   ◊ click any operator to insert it into the form expression
   ◊ enter a constant such as 1 or Rushorder in double quotes
6. Click Validate.
8. Click OK.
9. On the Create New Attribute Form dialog box:
   ◊ Under Source tables, select a table and click Set as Lookup. If you choose manual mapping, click the checkboxes of the tables you want mapped.
   ◊ Enter a Name and Description.
   ◊ Under Category used select a form category, or click Modify to create a new one.
   ◊ Select the Form category, Display type and Default sort from the lists.
10. Click OK.
11. On the Attribute dialog box, click Save and Close.

Notes:
• For your changes to take effect, you must first update the schema.
• When you add an expression to a form, you are only required to select an available column and move it over to the Expression box. You do not have to include any operators, parentheses, or validate the formula.
• Mapping method defaults to Automatic for the first expression of each form. The system maps the expression to each of the source tables. Manual mapping allows you to select which source tables are related to the expression.

Create an attribute in the Attribute Creation Wizard

Steps
1. From the Schema menu select Attribute Creation Wizard.
2. Click Define Rules to set the column datatypes and rules for naming attributes.
3. Click OK, then Next.
4. On the ID Column Selection dialog box, select the column names to act as IDs for the new attributes and click the > arrow.
To create **Compound Attributes**:

- Click **Compound Attributes** and click **Add**.
- In the **New Compound Attributes** dialog box, enter a **Name** and select multiple columns to build the compound attribute. The columns you select must exist in the same table.
- Click **OK**.

6. Click **Next**. The **Description Column Selection** page appears.

7. Select a description for each attribute in the **Description ColumnName** list.

8. Click **Next**. The **Lookup Table Selection** screen appears.

9. Select a lookup table for each attribute description in the **Lookup Table Name** list.

10. Click **Next**.

11. If you created compound attributes, the **Compound Attribute Definition** page displays. To define the compound attributes:

   - Select a compound attribute name and click **Add**.
   - In the **Description Columns** dialog box, select a column name and click **OK**.
   - Select a table from the **Lookup Table** list.
   - Click **Next**.

12. On the **Relationship Definition** page, select an **Attribute Name** and click **Add** to add children. The **Select Children Attributes** dialog box appears.

13. Select the available child attributes and click **OK**.

14. From the **Relationship type** list, select the type of parent-child relationship.

15. Click **Next**. The **Finish** page opens.

16. Review the attribute definition and click **Back** to make changes or click **Finish** to complete the attributes.

17. From the **Schema** menu, click **Update Schema**.

---

**Notes:**

- For your changes to take effect, you must first update the schema.
- To verify what columns were included in the compound attribute, click the (+) symbol next to the new attribute to display the hierarchy.
Define a column for SQL generation (attributes)

**Steps**

1. From the Folder list, click your project folder.
2. Navigate to your attribute folder. Select the folder to display its contents in the object viewer. Double-click the desired attribute. The Attribute Editor opens.
3. Select the Form name and click Modify. The Modify an attribute form dialog box appears.
4. Click the Column alias tab.
5. Click Modify. The Column Selection dialog box opens.
6. Select a column under Available columns, and click OK.
7. Click OK, then Save and Close.

**Note:** For your changes to take effect, you must first update the schema.

---

Define attribute children

**Steps**

1. From the Attribute Editor, click the Children tab.
2. Click Add. The Add Children Attributes dialog box opens.
3. Under Child candidates, select the attributes to be children, and click the right arrow button.
4. Click OK.
5. From the Relationship type list, select a relationship for each child.
6. From the Relationship table list, select a table.
7. Click Save and Close.
Define attribute parents

Steps

1. From the Attribute Editor, click the Parents tab.
2. Click Add. The Add Parent Attributes dialog box opens.
3. Under Parent candidates, select the attributes to be parents and click the right arrow button.
4. Click OK.
5. Click Save and Close.

Note: For your changes to take effect, you must first update the schema.

Delete a compound attribute

Steps

1. From the Folder list, click your project folder.
2. Navigate to your attribute folder. Select the folder to display its contents in the object viewer. Double-click the desired attribute. The Attribute Editor opens.
3. Select the compound attribute and click Ungroup.
4. Click Save and Close.
5. From the Schema menu, click Update Schema.
**Notes:**

- For your changes to take effect, you must first update the schema.
- All members of a compound key must have the same category.
- Only one form group can be set as a key. If you set a new form group to a key, it will remove the existing key from the other group.

---

**Delete a form group**

**Steps**

1. From the Folder list, click your project folder.
2. Navigate to your attribute folder. Select the folder to display its contents in the object viewer. Double-click the desired attribute. The Attribute Editor opens.
3. Select the form group and click Ungroup.
4. Click Save and Close.
5. From the Schema menu, click Update Schema.

**Notes:**

- For your changes to take effect, you must first update the schema.
- All members of a form group must have the same category.
- Only one form or form group may be set as a key, and you must have a key. When you set a new form to a key, the Attribute Editor removes the existing key from the previous form.

---

**Manually map an expression to source tables**

**MicroStrategy Desktop** automatically maps expressions to source tables. If you want to select the source tables that apply to the expression manually, select **Manual** as the mapping method.

**Steps**

1. From the Folder list, click your project folder.
2. Navigate to your attribute folder. Select the folder to display its contents in the object viewer. Double-click the desired attribute. The Attribute Editor opens.

3. From the Forms tab, click New. The Create a New Attribute Form dialog box opens.

4. Under Form definition, click New. The Create a New Form Expression dialog box opens.

5. Under Source table, select the table that acts as a source for the expression.

6. Under Available columns, double-click a column name to place it in the expression.


8. Click OK.

9. From the Definition tab, under Source tables, select the check boxes for each table where you want the new expression mapped.

10. Under Form name, enter a name for the new form expression.

11. From the lists, you can also enter a description, select a category, display, or sort.

12. Click OK.

13. Click Save and Close.


---

**Note:** For your changes to take effect, you must first update the schema.

---

**Modify an attribute**

**Steps**

1. From the Folder list, click your project folder.

2. Navigate to your attribute folder. Select the folder to display its contents in the object viewer. Double-click the desired attribute. The Attribute Editor opens.

3. Click Modify. The Modify Attribute Form dialog box appears.

4. Make any necessary changes and click OK.
5. From the Schema menu, click Update Schema.

**Note:** For your changes to take effect, you must first update the schema.

---

**Modify an attribute form**

**Steps**

1. From the **Folder** list, click your project folder.
2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Double-click the desired attribute. The **Attribute Editor** opens.
3. Click **Modify**. The **Modify Attribute Form** dialog box appears.
4. Select a Form expression and click **Modify**. The **Modify Form Expression** dialog box appears.
5. Make any necessary changes and click **OK**.
6. From the **Schema** menu, click **Update Schema**.

**Note:** For your changes to take effect, you must first update the schema.

---

**Modify form display properties**

When you initially create the attribute form, you either set the display properties or accept the defaults. By following the steps below, you can modify the properties at any time in the Attribute Editor.

**Steps**

1. From the **Attribute Editor**, click the **Forms** tab.
2. Click **Modify**. The **Modify Attribute Form** dialog box appears.
3. You can change the form category in the **Category used** list or click **Modify** to create a new category.
4. You can change the type of form display when you select from the Display type list.
5. Click Default sort and select from the list of sorting types.
6. Click OK.
7. From the Schema menu, click Update Schema.

**Note:** For your changes to take effect, you must first update the schema.

---

**Remove forms from the attribute browsing display**

**Steps**

1. From the Folder list, click your project folder.
2. Navigate to your attribute folder. Select the folder to display its contents in the object viewer. Double-click the desired attribute. The Attribute Editor opens.
3. Click the Display tab.
4. Under Browse Forms, select the forms you want to remove.
5. With the forms selected, click the (<) arrow to remove the forms from Browse Forms.
6. Click Save and Close.
7. From the Schema menu, click Update Schema.

**Note:** For your changes to take effect, you must first update the schema.

---

**Remove forms from the attribute report display**

**Steps**

1. From the Folder list, click your project folder.
2. Navigate to your attribute folder. Select the folder to display its contents in the object viewer. Double-click the desired attribute. The Attribute Editor opens.

3. Click the Display tab.

4. Under Report Display Forms, select the form you want to remove from the Attribute report display.

5. With the forms selected, click the < arrow button to remove the forms from the Attribute report display.

6. Click Save and Close.

7. From the Schema menu, click Update Schema.

---

**Note:** For your changes to take effect, you must first update the schema.

---

**Set attribute display properties**

**Steps**

1. From the Attribute Editor, click the Display tab.

2. Under Available forms, select the form you want to appear on the Report display or Browse forms.

3. Click > to move the form to Report Display or Browse forms.

4. To change the order of the forms, select a Form name and click the corresponding up or down arrows.

5. Click Save and Close.


---

**Notes:**

- For your changes to take effect, you must first update the schema.
- The default sorting order is Ascending.
View attribute properties

Steps
1. From the Folder list, click your project folder.
2. Highlight the attribute you want to review and right-click.
3. From the right-click menu, select Properties. The Properties dialog box appears.
4. Click OK to close.
Add a fact expression to an existing fact

Steps

1. Locate the existing fact in the Folder List and double-click it. The Fact Editor opens.
2. On the Definition tab, click New. The Create a new fact expression dialog box opens.
3. In the Source table box, select a table name to view its columns.
4. From the Available columns list, double-click the desired column name to move it to the Fact expression box.
6. Click OK, You are returned to the Fact Editor.
7. Save the fact definition.

Notes

• To see your changes in the project, you must update the schema.
• When you create a fact, you are only required to select an available column and move it over to the Fact expression box. You do not have to include any operators or parentheses, or validate the formula.
• Mapping method defaults to Automatic for the first fact or fact expression you create. The system maps the expression to each of the source tables. For subsequent facts, the default is Manual. Manual mapping allows you to select which source tables are related to the expression.
• When adding a fact expression to an existing fact, you must select a Source table different from those selected for existing fact expressions. A fact cannot have two different expressions in the same source table.
Change the mapping method for a fact expression

You can set the mapping method to **Automatic** or **Manual**. By selecting Automatic, the system maps the expression to each of the source tables. The mapping method defaults to **Manual**, so that you can select which source tables are related to the expression.

**Steps**

1. Open the fact in the **Fact Editor**.
2. Select the fact expression and click **Modify**. The **Modify Fact Expression** dialog box opens.
3. Select either **Automatic** or **Manual**.
4. Click **OK**. You are returned to the **Fact Editor**.
5. Save the fact.
6. From the Schema menu, choose **Update Schema**.

**Note**

To see your changes in the project, you must update the schema.

Create a fact extension based on a fact entry level

**Steps**

1. From the **Folder** list, click your project folder. Create a new fact in the Fact Editor.
2. After you have created a new fact, select the **Extension** tab in the Fact Editor.
3. Click **New**. The **Level Extension Wizard** opens to the Welcome page.
4. Click **Next**.
5. From the **General Information** page, enter a **Name** and **Description** for the new extension.
6. Select **Extend the fact entry level**.
7. Click **Next**.
8. From the **Extended Attributes** page, select one or more attributes and click the right arrow.
9. Click **Next**.
10. From the **Extension Type** page, choose to have the tables selected dynamically.

11. Click Next.

12. From the **Fact Selection** page, select a fact to use in the extension.

13. Click Next.

14. From the **Join Type** page, you can choose to dynamically select attributes or select them manually. If you choose to select attributes manually:
   - Click I will select the set of attributes.
   - Select the checkbox for each attribute to be included in the join.

15. Click Next.

16. If you chose to select attributes manually, the **Join Attributes** Direction page opens. Under **Join against**, click the arrows to change the direction of the join.

17. Click Next.

18. From the **Allocation** page, click the **Specify an allocation expression** box to enter an expression. You can apply operations such as +, -, /, and * to facts and attributes.

19. Click Next.

20. From the **Finish** page, review the properties included in the extensions.

21. Click Back to make changes or Finish to complete.

22. From the **Schema** menu, click **Update Schema**.

---

**Note:** To recognize your changes in the project, you must update the schema.

---

**Create a fact in the Fact Creation Wizard**

**Steps**

1. From the **Schema** menu, select **Fact Creation Wizard**.

2. Click **Define Rules** to set the column datatypes and rules for naming facts.
Create a formula for a fact expression

When you add an expression to an existing fact, you are only required to select an available column and move it to the Fact expression box. You do not have to include any operators or parentheses, or validate the formula. If you would like to use these features, complete the following steps.

Steps

1. From the Fact Editor, select the Definition tab and click New. The Create New Fact Expression dialog box opens.
2. Under Source table, select a table name. Its columns display in Available Columns.
3. To create an expression, you can
   ◊ Drag a column name from Available columns to the Fact expression box.
   ◊ Add operators by clicking on them.
   ◊ Enter a constant such as 1 or Rush order (in double quotes).
4. Click Validate.
6. Click OK.
7. From the Schema menu, click Update Schema.
Note:

• To recognize your changes in the project, you must update the schema.
• Mapping method defaults to Automatic. The system maps the expression to each of the source tables. Manual mapping allows you to select which source tables are related to the expression.
• When adding a fact expression to an existing fact, you must select a Source table different from those selected for existing fact expressions. A fact cannot have two different expressions in the same source table.

Create a fact and fact expression

You can create facts and their expressions in the Fact Editor. These steps demonstrate creating facts with the Fact Editor cascading dialogs. You can set the time-saving Fact Editor cascading dialogs in the Tools menu under Desktop Preferences.

Steps

1. From the Folder list, click your project folder.
2. From the File menu, click New, then click Fact. The Create New Fact Expression dialog box opens.
4. To create an expression, you can
   ◊ Drag a column name from Available columns to the Fact expression box.
   ◊ Add operators by clicking on them.
   ◊ Enter a constant such as 1 or Rush order (in double quotes).
5. Click Validate.
7. Click OK, then Save and Close.
8. Under Source table, select a table name to view its columns.
9. From the Available columns grid, double-click the desired column name to move it to the Fact expression text box.
10. Under **Mapping method**, select **Automatic** or **Manual**. **Automatic** is the default for the first fact or fact expression you create.

11. Click OK.

12. Click **Save and Close**.

13. From the **Schema** menu, click **Update Schema**.

---

**Note:**

- To recognize your changes in the project, you must update the schema.
- When you create a fact, you are only required to select an available column and move it over to the **Fact expression** box. You do not have to include any operators or parentheses, or validate the formula.
- Mapping method defaults to **Automatic** for the first fact or fact expression you create. The system maps the expression to each of the source tables. For subsequent facts, the default is **Manual**. Manual mapping allows you to select which source tables are related to the expression.

---

**Define a column for SQL generation (facts)**

You can choose whether you want to accept the default column name as the name of the column used in the temporary table for SQL generation, or you can define your own column. The steps below demonstrate creating your own column.

**Steps**

1. From the **Folder** list, click your project folder.
2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Double-click the desired fact. The **Fact Editor** opens.
3. Select the **Column Alias** tab.
4. Click **Modify**. The **Column Selection** dialog box opens.
5. Select a column and click **OK**, or click **New** to create a new column definition.
6. If you clicked **New**, the **Definition** dialog box opens.
7. Enter a name in the **Column name** box, and select a **Data type** from the list.
8. If applicable, define the precision, scale, and length of the datatype, and click OK.

9. Enter a Byte length or click the arrows to change the number.

10. Click OK. New columns now display in the Column Selection dialog box.

11. Click OK again, then Save and Close.


---

**Note:**
- To recognize your changes in the project, you must update the schema.
- Depending on the datatype you select, one or more of the following boxes is visible. See the What’s This? help (Shift + F1) for more information.

---

<table>
<thead>
<tr>
<th>Datatype</th>
<th>Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>Float</td>
<td>Bit length</td>
</tr>
<tr>
<td>Integer, Unsigned, Char, VarChar, LongVarChar, Binary, VarBin, LongVarBin</td>
<td>Byte length</td>
</tr>
<tr>
<td>Numeric, Decimal</td>
<td>Precision</td>
</tr>
<tr>
<td>Numeric, Decimal</td>
<td>Scale</td>
</tr>
<tr>
<td>Time, TimeStamp</td>
<td>Time scale</td>
</tr>
<tr>
<td>Real</td>
<td>[None]</td>
</tr>
<tr>
<td>Double</td>
<td>[None]</td>
</tr>
<tr>
<td>Date</td>
<td>[None]</td>
</tr>
</tbody>
</table>

---

**Delete a fact extension**

**Steps**

1. From the Folder list, click your project folder.

2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Double-click the desired fact. The Fact Editor opens.
3. From the Extensions tab, select the extension and click Delete.
4. Click Save and Close.
5. From the Schema menu, click Update Schema.

**Note:** To recognize your changes in the project, you must update the schema.

---

**Disallow a fact entry level**

You can completely or partially disallow a fact entry level.

**Steps**

1. From the Folder list, click your project folder. Create a new fact in the Fact Editor.
2. From the Extensions tab click New. The Level Extension Wizard opens.
3. On the General Information page enter the Name and Description.
4. Select Disallow the fact entry level.
5. Click Next.
6. From the Extended Attributes page, select the attribute to be disallowed and click the right arrow.
7. Click Next.
8. From the Finish screen, click Back to make changes, or Finish to complete the disallow.

**Note:** To recognize your changes in the project, you must update the schema.
Extend a fact from its current fact to a new fact

Steps

1. From the Folder list, click your project folder. Create a new fact using Create a new fact and fact expression.
3. On the General Information screen, enter the Name and Description.
4. Select Extend the fact entry level.
5. Click Next.
6. From the Extended Attributes screen, select one or more attributes and click the right arrow button.
7. Click Next.
8. From the Extension Type screen, choose the Select the relationship table dynamically option.
9. Click Next.
10. From the Fact Selection screen, select the fact to use in performing the extension.
11. Click Next.
12. From the Join Type screen, choose the Dynamically select the best set of attributes option.
13. Click Next.
14. From the Allocation screen, you can select the Specify an allocation expression checkbox to enter an expression.
15. Click Next.
16. From the Finish screen, click Finish to complete the level extension.
**Note:**

- When you create a fact, you are only required to select an available column and move it over to the **Fact expression** box. You do not have to include any operators, parentheses, or validate the formula.
- Mapping method defaults to **Automatic** for the first fact or fact expression you create. The system maps the expression to each of the source tables. For subsequent facts, the default is **Manual**. Manual mapping allows you to select which source tables are related to the expression.

**Lower a fact entry level**

You can lower a fact entry level, which is the opposite of extending a fact. By lowering a fact, you make it available at a level lower than where it is stored.

**Steps**

1. From the **Folder** list, click your project folder. Create a new fact in the Fact Editor.
2. From the **Extensions** tab click **New**. The **Level Extension Wizard** opens to the Welcome page. Click **Next**.
3. From the **General Information** screen enter the **Name** and **Description**.
4. Select **Lower the fact entry level**.
5. Click **Next**.
6. From the **Extended Attributes** screen, select one or more attributes and click the right arrow.
7. Click **Next**.
8. From the **Join Type screen**, select the attributes to be included when performing joins.
9. Click **Next**.
10. On the **Join Attributes Direction** screen, specify the join direction by clicking the arrows.
11. Click **Next**.
12. From the **Allocation** screen, you can select the **Specify an allocation expression** checkbox to enter an expression.
13. Click Next.
14. From the Finish screen, click Finish to complete lowering the fact entry level.
15. From the Schema menu, click Update Schema.

Note: To recognize your changes in the project, you must update the schema.

Map a fact expression manually to source tables

MicroStrategy Desktop automatically locates the tables that contain the fact to which you are adding an expression. By default, the system automatically maps the expression to these source tables. If you would like to select the source tables that apply to the expression, you should select Manual as the mapping method.

Steps
1. From the Folder list, click your project folder.
2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Double-click the desired fact. The Fact Editor opens.
3. From the Definition tab, click Modify. The Modify a fact expression dialog box opens.
5. Click OK.
6. In the Fact Editor, select the fact expression. Its source tables will display under Source tables.
7. Select the source tables to map to the fact expression by clicking on the table name or check box.
8. Click Save and Close.
Note:

- To recognize your changes in the project, you must update the schema.
- Mapping method defaults to Automatic for the first fact or fact expression you create. The system maps the expression to each of the source tables. For subsequent facts, the default is Manual. Manual mapping allows you to select which source tables are related to the expression.
- A fact cannot have two different expressions in the same source table.

Modify a column alias

Steps

1. From the Folder list, click your project folder.
2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Double-click the desired fact and the Fact Editor opens.
3. From the Column Alias tab, click Modify. The Column Selection dialog box opens.
4. To choose an existing column name, select one from the Available columns list and click OK.
5. To create a new column name:
   ◊ Click New.
   ◊ In the Definition dialog box, enter a new Column Name and select a Datatype.
   ◊ Click OK.
6. Click OK, then Save and Close.
7. From the Schema menu, click Update Schema.

Note: To recognize your changes in the project, you must update the schema.
Modify a fact expression

Steps

1. From the Folder list, click your project folder.

2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Double-click the desired fact. The Fact Editor opens.

3. From the Definition tab, click Modify. The Modify a fact expression dialog box opens.

4. Modify the expression as desired:
   - To replace a column name used in the expression, select the column under Fact expression, and click Clear.
   - To add a new column name in the Available columns grid, double-click the desired column name to move it to the Fact expression text box.
   - To use the operators, click the desired operators to complete the expression.

5. Click OK, then click Save and Close.


---

Note:

- To recognize your changes in the project, you must update the schema.
- Mapping method defaults to Automatic for the first fact or fact expression you create. The system maps the expression to each of the source tables. For subsequent facts, the default is Manual. Manual mapping allows you to select which source tables are related to the expression.

Modify a fact extension

Steps

1. From the Folder list, click your project folder.

2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Double-click the desired fact. The Fact Editor opens.

3. On the Extensions tab, select the Extension name and click Modify. The Level Extension Wizard opens.
4. Make any necessary changes and click Finish at the end of the wizard.
5. Click Save and Close.

**Note:** To recognize your changes in the project, you must update the schema.

---

**Unmap an expression from a source table**

Source tables are tables that include the fact used in the expression. If you do not want a table to be used as a source for the expression, complete the following steps.

**Steps**

1. From the Folder list, click your project folder.
2. Navigate to your fact folder. Select the folder to display its contents in the object viewer.
3. Double-click the desired fact. The Fact Editor opens.
4. Under the Definition tab, highlight the desired expression.
5. In the Source tables box, clear the check boxes for each table no longer used as the source for a fact expression.
6. Click Save and Close.
7. From the Schema menu, click Update Schema.

**Note:** To recognize your changes in the project, you must update the schema.
View fact properties

Steps
1. From the Folder list, click your project folder.
2. Highlight the fact you want to review and right-click.
3. From the right-click menu, select Properties. The Properties dialog box displays.
4. Click OK to close.
Add a user hierarchy to the system hierarchy

**Steps**

1. In the folder list, locate the user hierarchy you wish to add to the system hierarchy.
2. Drag the user hierarchy to the Data Explorer folder to move it or copy the user hierarchy and paste it in the Data Explorer.

**Notes**

- To see your changes in the project, you must update the schema.
- If the Data Explorer folder is not visible, display it using the Desktop Preferences dialog (from the Tools menu, choose Desktop Preferences).

Add attributes to a hierarchy

**Steps**

1. Open the hierarchy in the Hierarchy Editor.
2. In the Attributes section (top half of the editor), click Add. The Select Objects dialog box opens.
3. From the Available attributes list, select the attributes to add to the hierarchy and click > to add them to the Selected Objects list.
4. Click OK. You are returned to the Hierarchy Editor.
5. Save the hierarchy.
Note

To see your changes in the project, you must update the schema.

Add browsing attributes to a hierarchy

Steps
1. Open the hierarchy in the **Hierarchy Editor**.
2. On the **Browse Attributes** tab, click **Add** (bottom half of the editor). The **Browse Attributes** dialog box opens.
3. From the **Available attributes** list, select the browse attributes to add to the hierarchy and click **>** to add them to the **Selected Attributes** list.
4. Click **OK**. You are returned to the **Hierarchy Editor**.
5. Save the hierarchy.
6. From the **Schema** menu, choose **Update Schema**.

Note

To see your changes in the project, you must update the schema.

Add hierarchy attribute filters

Steps
1. Open the hierarchy in the **Hierarchy Editor**.
2. On the **Filters** tab, click **Add** (bottom half of the editor). The **Select Objects** dialog box opens.
3. From the **Available objects** list, select the browse attributes to add to the hierarchy and click **>** to add them to the **Selected Objects** list.
4. Click **OK**.
5. You are returned to the **Hierarchy Editor**.
6. Select a filter name and click the up or down arrow to change the order of the filters within the attribute.
7. Save the hierarchy.
8. From the **Schema** menu, choose **Update Schema**.
Note
To see your changes in the project, you must update the schema.

Create a hierarchy

Steps
1. From the Folder List, click your project folder.
2. From the File menu, click New, then click Hierarchy.
3. On the Select objects dialog box, select the attributes to include in the hierarchy, and click the right arrow.
4. Click OK. The Hierarchy Editor displays.
5. To create new browse attributes:
   ◇ Select the attribute in the top attribute grid.
   ◇ On the Browse Attributes tab, click Add.
   ◇ The Browse Attributes dialog box displays.
   ◇ Select the attributes for browsing and click the right arrow.
   ◇ Click OK.
6. Select Use this as a Drill Hierarchy to drill down on the browse attributes.
7. Click Save and Close.

Notes:
- To recognize your changes in the project, you must update the schema.
- The Element Display default is Unlocked.

Define attribute relationships within a hierarchy

Attribute relationships allow for browsing from the Data Explorer folder. They are the attributes that you can drill down from a report. Follow the steps below to define attribute relationships within a hierarchy.
Steps

1. From the Folder List, click your project folder.
2. Navigate to your hierarchy folder. Select the folder to display its contents in the object viewer. Double-click the desired hierarchy. The Hierarchy Editor opens.
3. Under Hierarchy attributes, select the attribute you want to define.
4. From the Browse Attributes tab, click Add. The Hierarchy Attributes dialog box opens.
5. Under Available attributes, select the attributes to add as browsing attributes and click the right arrow button. Click OK.
6. To define the order of the browsing attributes within the attribute, select an attribute name and click the up or down arrow.
7. Click Save and Close.

Notes:

• To recognize your changes in the project, you must update the schema.
• The browse attributes that are available from the Hierarchy Attributes dialog box are the attributes you chose to be in the hierarchy. If you want to include an attribute that is not in the list, you must first add that attribute to the Hierarchy Attributes grid.

Delete a hierarchy

You can delete a hierarchy from any project.

Steps

1. From the Folder list, click your project folder.
2. Highlight the hierarchy you want to delete and right-click.
3. From the right-click menu, select Delete.
4. Click Yes.
5. From the Schema menu, click Update Schema.
Remove attribute filters

Steps

1. From the Folder List, click your project folder.
2. Navigate to your hierarchy folder. Select the folder to display its contents in the object viewer. Double-click the desired hierarchy. The Hierarchy Editor opens.
3. Under Hierarchy attributes, select the attribute to add a filter to.
4. Click the Filters tab.
5. Under Attribute filters, select the filter you want to remove and click Remove.
6. Click Save and Close.
7. From the Schema menu, click Update Schema.

Note: To recognize your changes in the project, you must update the schema.

Remove attributes from a hierarchy

Steps

1. From the Folder List, click your project folder.
2. Navigate to your hierarchy folder. Select the folder to display its contents in the object viewer. Double-click the desired hierarchy. The Hierarchy Editor opens.
3. Under Hierarchy attributes, select the attribute to remove and click Remove.
4. Click Save and Close.
5. From the **Schema** menu, click **Update Schema**.

**Note:** To recognize your changes in the project, you must update the schema.

---

### Set hierarchy display properties

**Steps**

1. From the Folder List, click your project folder.
2. Navigate to your hierarchy folder. Select the folder to display its contents in the object viewer. Double-click the desired hierarchy. The **Hierarchy Editor** opens.
3. Under **Hierarchy attributes**, select the attribute to set element display properties for.
4. Click the **Element Display** tab.
5. Select either **Locked**, **Unlocked**, or **Limit**. If you select **Limit**, you must type the limit in the text box.
6. Select **Use as entry point in browse sequence** to make the attribute an entry point in the Data Explorer.
7. Click **Save and Close**.
8. From the **Schema** menu, click **Update Schema**.

**Notes:**

- To recognize your changes in the project, you must update the schema.
- The Element Display default is Unlocked.

---

### Set hierarchy for drilling

To enable drilling, you must change the hierarchy so that drilling is enabled. When you create a hierarchy with drilling, you are making the option available when the user adds the hierarchy to a report.
**Steps**

1. From the **Folder** list, click your project folder.
2. Navigate to your hierarchy folder.
3. Double-click a hierarchy. The **Hierarchy Editor** opens.
4. Select **Use as a drill hierarchy**.
5. Click **Save and Close**.
6. From the **Schema** menu, click **Update Schema**.

---

**Notes:**

- To recognize your changes in the project, you must update the schema.
- If you do not specify the drilling option for the hierarchy, then the drilling option does not display in the right-click menu of any report using the hierarchy.

---

**Set system hierarchy display in Data Explorer**

The system hierarchy displays in the Data Explorer by default each time you create a new project. If you do not want the system hierarchy available in the Data Explorer when you create new projects, you can change the preferences on your desktop.

**Steps**

1. From the **Tools** menu, click **Desktop Preferences**. The **Desktop Preferences** dialog box opens.
2. Under **Data Explorer**, select **Display the system hierarchy in the Data Explorer**.
3. Click **OK**.
4. From the **Schema** menu, click **Update Schema**.

---

**Note**: To recognize your changes in the project, you must update the schema.
Unlock hierarchy display

**Steps**
1. From the Folder List, click your project folder.
2. Double-click the desired hierarchy. The Hierarchy Editor opens.
3. Under Hierarchy attributes, select an attribute and click the Element Display tab.
4. Click Unlocked.
5. Click Save and Close.

---

**Notes:**
- To recognize your changes in the project, you must update the schema.
- The Element Display default is Unlocked.

---

View data explorer in the desktop

The Data Explorer is, by default, visible in the Object Browser. You can change your desktop settings to show or hide the Data Explorer.

**Steps**
1. From the Tools menu click Desktop Preferences. The Desktop Preferences dialog box opens.
2. Click Show the Data Explorer in the desktop.
3. Click OK.

---

View hierarchy properties

**Steps**
1. From the Folder list, click your project folder.
2. Highlight the hierarchy you want to review and right-click.
3. From the right-click menu, select Properties. The Properties dialog box displays.
4. Click **OK** to close.

5. From the **Schema** menu, click **Update Schema**.

**Note:** To recognize your changes in the project, you must update the schema.
Add a data slice to a metadata partition

**Steps**

1. Double-click a metadata partition. The Partition Mapping Editor opens.
2. Select the table to which you wish to add a data slice and click Define. The Data Slice Editor opens.
3. Using the Object Browser, select an attribute and drag it onto the data slice definition. The Attribute Qualification window opens.
4. Click Add.
5. In the Select objects dialog box, select an element from the Available objects list and click > to add it to the Selected Objects list.
6. Click OK. You are returned to the Data Slice Editor.
7. Save the data slice definition. You are returned to the Partition Mapping Editor.
8. Save the partition mapping.
9. From the Schema menu, choose Update Schema and update the schema.

**Note**

To see your changes in the project, you must update the schema.
Create a metadata partition

Steps

1. From the Folder list, click your project folder.
2. From the File menu, click New, then click Partition.
3. On the Partition Table Selection dialog box, select tables from Available tables and click > to include them as a partition.
4. Click OK.
5. On the Partition Mapping Editor, set the Partition mapping logical size using the arrows.
6. Select a table and click Define under Data slice definition.
7. The Data slice editor displays.
   ◊ Under the Object browser, select an attribute and drag it to Data slice definition.
   ◊ Under Attribute qualification, make any changes to the attribute or form and click Add to add elements.
   ◊ From the Select objects dialog box, select one or more IDs and click the right arrow.
8. Click OK, then Save and Close.

Note

To recognize your changes in the project, you must update the schema.

Modify a data slice

Steps

1. Double-click the desired metadata partition. The Metadata Partition Mapping Editor opens.
2. Select a table under Tables defining the partition mapping.
3. Under Data slice definition click Define. The Data Slice Editor displays.
4. In the **Data slice definition** box, you can drag the definition to the left to remove it.

5. If you have more than one attribute in the data slice definition, you can double-click the logical operator to change the definition.

6. Click **Save and Close**.

7. From the **Schema** menu, click **Update Schema**.

**Note**
To recognize your changes in the project, you must update the schema.

## Set partition mapping logical size

**Steps**

1. Double-click a partition. The **Partition Mapping Editor** opens.

2. In the **Partition mapping logical size** box, use the arrows or enter a number directly.

3. Click **Save and Close**.

4. In the **Schema** menu, click **Update Schema**.

**Note**
To recognize your changes in the project, you must update the schema.
Print the project schema

You can print the project schema any time after a project has been set up.

**Steps**

1. From the Schema menu, select Print Project Schema. The Print Project Schema dialog opens.
2. Click the checkboxes corresponding to the print options you want.
3. Click the Setup button to determine formatting options under the Page, Margin, and Header/Footer tabs.
4. Click options on the Page Setup dialog to define printer properties. Click OK or Cancel to return to Page Setup.
5. Click Print Preview to see a final version of the printout before you send it to the printer. If you need to make further adjustments, click Page Setup and make the necessary changes. If you want to print the project schema, you can click Print from the preview window.
6. Click OK. The chosen print options are sent to the printer according to the parameters you set.

**Note**

You must select at least one print option to enable the print feature.
CHAPTER 27

Projects

Create a project

From the Schema menu, choose Create New Project to open the Project Creation Assistant.

The Project Creation Assistant takes a linear approach to creating a new project, selecting tables, creating facts, and creating attributes.

For step-by-step instructions on how to create the different components of a project using the Project Creation Assistant, refer to the following How Do I...? topics:
- Create a project with the Project Creation Assistant
- Select tables from the Warehouse Catalog using the Project Creation Wizard
- Create facts with the Project Creation Assistant
- Create attributes with the Project Creation Assistant

Note

Before you can use your new project in 3-tier mode, you must register it. When you have created a new project and closed the Project Creation Assistant, you can register your project through the MicroStrategy Server Configuration Editor.

Create attributes with the Project Creation Assistant

The Attribute Creation Wizard screens all table columns in the warehouse and applies simple rules to identify and format attributes. As an optional step, you may edit these rules by clicking Define Rules on the first page of the wizard.
Steps

1. Click the arrow for Create Attributes. The Attribute Creation Wizard opens.

2. Click Define Rules.

3. On the Define Rules screen, select the column datatypes to be displayed.

4. Select your attribute name rules and naming conventions.

5. Enter the warehouse search criteria to maintain naming conventions. Click OK, then click Next.

6. On the ID Column Selection screen, select Available Columns to be the identifiers of the new attributes and click the right arrow.

7. If you want to create compound attributes, click Compound Attributes, then click Add.

8. From the New Compound Attribute dialog box, enter a name and select two or more columns to make up the compound attribute. Click OK, then click Next.

9. On the Description Column Selection screen, select the default attribute description and select a description column name from the list. Click Next.

10. On the Lookup Table Selection, make any changes to the default lookup tables for each attribute. Click Next.

11. If you created a compound attribute, the Compound Attribute Definition screen displays.

   ◊ Select a compound attribute and choose an ID Lookup table from the list.

   ◊ Select a compound attribute and click Add to add description columns. In the Description columns dialog box, select the columns and click OK. Click Next.

12. On the Relationship Definition screen, select an attribute and click Add to add children.

13. From the Select Children Attributes dialog box, select an attribute name and click OK.

14. Make any necessary changes in the Relationship type list. Click Next.

15. Click Next. On the Finish screen, review created attributes and click Finish.

16. The Project Creation Assistant returns with a green check mark next to Create Attributes.
Notes

- The default cardinality of the relationship is one-to-many.
- The wizard only displays columns of the data types selected in the Attribute Creation Rules dialog box as possible attributes. If no attributes appear on the ID Column Selection screen, check to ensure the data type selected in the Attribute Creation Rules dialog box matches your attribute type.
- Attributes must have unique names; no two attributes may be named the same.

Create facts with the Project Creation Assistant

The Fact Creation Wizard screens all table columns in the warehouse and applies simple rules to identify and format facts. As an optional step, you may edit these rules by clicking Define Rules on the first screen of the wizard.

Steps

1. Click the right arrow next to Create Facts. The Fact Creation Wizard opens.
2. Click Define Rules to set the column datatypes and rules for naming facts.
3. On the Define Rules screen, select the column datatypes to be displayed.
4. Select the column data types to be displayed and rules to apply to fact names. Click OK.
5. Click Next.
6. On the Column Selection screen, select columns and click the right arrow to move them to the Facts list.
7. Click Next.
8. On the Finish screen, review the facts created and click Finish. The Project Creation Assistant returns with a green check mark next to Create Facts.

Create facts with the Project Creation Assistant 201
Notes

- The fact names are formatted according to the rules specified in the Fact Creation Rules dialog box, which is accessed by clicking the Define Rules button on the first screen of the wizard.
- The wizard only displays columns of the data types selected in the Fact Creation Rules dialog box as possible facts. In many data warehouses, all the facts are limited to only a few data types, such as numeric and date/time. Restricting the display to just those data types makes the job of finding the fact columns easier.

Create a project with the Project Creation Assistant

These instructions correspond to the Create project option on the Project Creation Assistant.

Steps

2. Click the right arrow next to Create Project. The New Project dialog box opens.
3. On the General tab, enter a name for the new project.
4. Select a Project Source from the drop-down list or click New to create a new project source.
5. Click OK. The Login dialog box opens. Click New to create a new project source.
6. Type a valid username and password. Click OK. The Project Creation Assistant returns with a green check mark next to Create Project.

Note

The Login dialog box is the gateway to the data source selected from the New Project dialog box. If you are not authorized to create projects in the data source specified, you will not be able to proceed past this step.
Create a project source

Steps

1. From the Tools menu, choose Project Source Manager. The Project Source Manager dialog box displays.

2. Click Add. The Project Source Manager dialog box displays.

3. Enter a Project source name and select the connection type from the Connection mode list. Depending on the connection mode you choose, the options on the Connection tab may vary.

4. If you choose:
   ◊ Server, enter a Server name and Port number.
   ◊ Direct, select an ODBC DSN from the list. Enter the database authentication Login Id and Password, which can be obtained from the Administrator.
   ◊ 6.x project, enter a DSS file name.

5. Click OK.


Note

To recognize your changes in the project, you must update the schema.

Display hidden objects

In MicroStrategy 7, some folders are hidden by default, but you can make them visible. In most projects, the System Hierarchy and Data Explorer are hidden. The steps below make these folders visible for the current project.

Steps

1. From the Tools menu, select Desktop Preferences.

2. On the Desktop tab, click Browsing Options.

3. Select Display hidden objects.

4. Click OK.
Display the system hierarchy in the Data Explorer

If you do not see the system hierarchy display in the Data Explorer, you can make it visible using the steps below.

**Steps**
1. From the **Tools** menu, choose **Desktop preferences**.
2. From the **Desktop** tab, click **Display the system hierarchy in the Data Explorer**.
3. Click **OK**.

Display the Data Explorer

**Steps**
1. From the **Tools** menu, click **Desktop Preferences**.
2. On the **Desktop** tab, click **Show the Data Explorer in the desktop**.
3. Click **OK**.

Modify a project source

**Steps**
1. From the **Tools** menu, choose **Project Source Configuration**. The **Project Source Manager** dialog box displays.
2. Select a **Project source** and click **Modify**. The **Project Source Setup** dialog box displays. Make any necessary changes.
3. Click **OK**.
4. If you changed the connection mode, you have been disconnected from the project and need to reconnect to see the changes.

**Note**

If you changed any setting except the connection mode, you must update the schema to see the changes. To do that, click **Update Schema** on the **Schema** menu.
Remove a fact from a project

**Steps**
1. From the Folder list, click your project folder.
2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Select the desired fact and right-click.
3. From the menu, click **Delete**.
4. On the **Confirm Delete Object** dialog box, click **Yes**.
5. From the **Schema** menu, click **Update Schema**.

**Note**
To recognize your changes in the project, you must update the schema.

Remove a hierarchy from a project

**Steps**
1. From the Folder list, click your project folder.
2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Select the desired hierarchy and right-click.
3. From the menu, click **Delete**.
4. On the **Confirm Delete Object** dialog box, click **Yes**.
5. From the **Schema** menu, click **Update Schema**.

**Note**
To recognize your changes in the project, you must update the schema.

Remove an attribute from a project

**Steps**
1. From the Folder list, click your project folder.
2. Navigate to your attribute folder. Select the folder to display its contents in the object viewer. Select the desired attribute and right-click.
3. From the menu, click **Delete**.
4. On the Confirm Delete Object dialog box, click Yes.
5. From the Schema menu, click Update Schema.

Note
To recognize your changes in the project, you must update the schema.

Select tables from the Warehouse Catalog using the Project Creation Wizard

The warehouse catalog shows the data warehouse tables that will be used in the project. Each available warehouse is identified by its data source name (DSN).

Steps
1. Click the arrow next to Select Tables from the Warehouse Catalog. The Warehouse Database Instance dialog box opens.
2. In the Project’s Warehouse Database Instance list, select an instance and click OK. Or click New to create a database instance.
3. If you chose New, the Database Instance Wizard displays.
   ◊ Enter Name and Description, and select a Database Type. Click Next.
   ◊ Select a data source and enter a Login and Password. Click Next.
   ◊ On the Completion screen, review the set parameters and click Finish. Click OK.
4. In the Warehouse Catalog dialog box, select available tables and click the right arrow to include them.
5. Click Save and Close. The Project Creation Assistant returns with a green check mark next to Select Tables from the Warehouse Catalog.
Set logical table size

You can set the logical table size in the Table Editor.

**Steps**

1. Double-click a table in the Folder list.
2. Click the *Logical View* tab.
3. In the *Logical size* box, enter a number for the size you want the table.
4. Click *Save and Close*.
5. In the *Schema* menu, click *Update Schema*.

**Note**

To recognize your changes in the project, you must update the schema.

View table structure

You can view the logical and physical structures of each table in the Table Editor.

**Steps**

1. Double-click a table in the Folder list.
2. Click the *Logical View* tab or *Physical View* tab.
3. Click *Close*. 
Add a new member attribute to a transformation

Steps

1. Right-click the transformation to which you wish to add a member attribute and choose Edit. The Transformation Editor opens.
2. Click Add. The Select a member attribute dialog box opens.
3. Select an attribute and click Open. The Define a new member attribute expression dialog box opens.
4. In the Table box, select a source table name.
5. In the Available Columns list, double-click a column name to place it in the Member Attribute Expression box.
6. Create the expression and click Validate to make sure the expression is valid.
7. Click OK. You are returned to the Transformation Editor.
8. Select a Transformation Mapping Type.
9. Save the transformation.

Note

To see your changes in the project, you must update the schema.
Create a member attribute

**Steps**

1. Double-click an existing transformation or create a new one. The Transformation Editor opens.
2. Click Add.
3. On the Select a Member Attribute dialog box, select an attribute and click Open.
4. On the Define a New Member Attribute Expression dialog box, select a table from the list.
5. Double-click one or more columns to place them in the Member Attribute Expression dialog box.
6. Click on any operator to complete the expression.
7. Click Validate to ensure the expression is valid.
8. Click OK.
9. On the Transformation Editor, select the Mapping Type and click Save and Close.

**Note**

To recognize your changes in the project, you must update the schema.

Create a transformation

**Steps**

1. From the Folder list, click your project folder.
2. From the File menu, click New, then click Transformation.
3. On the Select a Member Attribute dialog box, select an Attribute name and click OK.
4. On the **Define a new member attribute expression** dialog box, double-click a column name to place it in the expression box.
5. Click the operators to insert them into the expression and click **Validate**. Click **OK**.
6. If you would like to add an attribute to the transformation, click **Add**.
7. On the **Transformation** dialog box click **Save and Close**.
8. From the **Schema** menu, click **Update Schema**.

**Note**

To recognize your changes in the project, you must update the schema.

---

**Delete a transformation**

**Steps**

1. From the **Folder** list, click your project folder.
2. Highlight the transformation you want to delete and right-click.
3. From the right-click menu, select **Delete**.
4. Click **Yes**.
5. From the **Schema** menu, click **Update Schema**.

**Note**

To recognize your changes in the project, you must update the schema.

---

**Modify a transformation**

**Steps**

1. From the **Folder** list, click your project folder.
2. Navigate to your fact folder. Select the folder to display its contents in the object viewer. Double-click the desired transformation. The Transformation Editor opens.

3. Select a Member attribute and click Modify. The Modify an existing member attribute expression dialog box opens.

4. Make any changes in the expression box and click Validate then OK.

5. From the Schema menu, click Update Schema.

Note
To recognize your changes in the project, you must update the schema.

Modify a transformation member attribute

Steps

1. In the Transformation Editor, select a Member attribute name and click Modify. The Modify an existing member attribute expression dialog box opens.

2. In the Table list, select a source table name.

3. Under Available Columns, double-click a column name to place it in the Member Attribute Expression.

4. Click on the mathematical operators to place them in the expression and click Validate. When you have a valid expression click OK.

5. Select a Transformation Mapping Type.

6. Click Save and Close.

7. From the Schema menu, click Update Schema.

Note
To recognize your changes in the project, you must update the schema.
Remove an attribute member

Steps

1. In the Transformation Editor, select a Member Attribute name and click Remove.
2. Click Save and Close.
3. From the Schema menu, click Update Schema.

Note

To recognize your changes in the project, you must update the schema.
CHAPTER 30

Warehouse Catalog

Change the database instance for a project

Steps
1. From the Schema menu, choose Warehouse Catalog. The Warehouse Catalog opens.
2. From the Tools menu, choose Options. The Warehouse Catalog Options dialog box opens.
3. Select a database instance from the Database Instance box.
4. Click OK. You are returned to the Warehouse Catalog.
5. Click Save and Close.

Note
To see your changes in the project, you must update the schema.

Change the project warehouse

You can switch project warehouses provided that the table names and structures in both warehouses are the same.

Steps
1. From the Schema menu, choose Warehouse Catalog. The Warehouse Catalog opens.
2. From the Tools menu, choose Options. The Warehouse Catalog Options dialog box opens.
3. On the Catalog tab, choose the database instance that points to the other warehouse.
4. Click OK. You are returned to the Warehouse Catalog.
5. On the toolbar, click **Save and Close**.

**Change the project warehouse when the table name space is different**

You can switch the project warehouse even if the warehouses are in different database name spaces. To do this, the table names and structures in both warehouses must be the same.

**Steps**

1. From the **Schema** menu, choose **Warehouse Catalog**. The **Warehouse Catalog** opens.

2. From the **Tools** menu, choose **Options**. The **Warehouse Catalog Options** dialog box opens.

3. On the **Catalog** tab, choose the database instance that points to the other warehouse.

4. Select **Ignore current table name space when reading from the database catalog and update using new table name space**. This causes the **Warehouse Catalog** to ignore the table name space when it compares the current tables in the project to the ones in the new warehouse.

5. Click **OK**. You are returned to the **Warehouse Catalog**.

6. On the toolbar, click **Save and Close**.

7. Open the **Warehouse Catalog** and clear the check box you selected in step 4 to avoid ignoring the Table Name Space in the future.

**Note**

Not all databases support the concept of table name space. Check the documentation for your database to see if this concept applies to your situation.

**Edit the database instance for a project**

You can edit the database instance for each project.

**Steps**

1. In the **Schema** menu, click **Warehouse Catalog**.

2. On the **Warehouse Catalog browser**, click **Options**.
3. On the **Warehouse Catalog Options** dialog box, click **Edit**. The Database Instances dialog box opens.

4. On the **General** tab, make any changes to the database connection and connection type.

5. Click **OK**.

**Note**

To create a new database instance, see the following how do I:

Create a database instance

---

**Edit warehouse catalog tables**

You can add or remove tables in the warehouse catalog for each project.

**Steps**

1. In the **Schema** menu, click **Warehouse Catalog**.
2. On the **Warehouse Catalog** browser, select one or more tables.
3. Click the left arrow to remove tables from the warehouse catalog.
4. Click the right arrow to add tables to the warehouse catalog.
5. Click **Save and Close**.

---

**Set warehouse catalog browser options**

You can load and change options in the Warehouse Catalog.

**Steps**

1. Access the Warehouse Catalog Options dialog from the Warehouse Catalog Browser itself or from the SQL Generation options in the **Schema** menu.
2. Make any changes to the options and settings.
3. Click **OK** to preserve your changes.
Set the warehouse catalog read mode for a project

You can set the warehouse catalog read mode for each project to be manual or automatic.

**Steps**

1. In the Schema menu, click Warehouse Catalog.
2. On the Warehouse Catalog browser, click Options. The Warehouse Catalog Options dialog box opens.
3. In the Catalog Read Mode box, select Automatic or Manual.
4. Click OK.

**Note**

If you select Automatic, the warehouse catalog tables are read as soon as the catalog browser is loaded. If you select Manual, the catalog tables are only read when the Read the Warehouse Catalog option is selected on the Warehouse Catalog Browser.

View a table definition in the warehouse catalog

You can view the definition of a table used in your project through the warehouse catalog.

**Steps**

1. In the Schema menu, click Warehouse Catalog. The Warehouse Catalog browser opens.
2. Select a table being used in the current project.
3. On the Tools menu, click Table Structure.
4. Click Close.

**Note**

By viewing the table structure of a table, you can see its datatype and size.
Appendixes

- Pass-Through Expressions
- Attribute Cardinality
- SQL Engine Logic
- Warehouse Catalog SQL
- Query Engine Logic
Pass-through expressions

Pass-through expressions provide access to functionality that is not standard in MicroStrategy Desktop, but can be obtained through the relational database. Pass-through expressions act as containers for non-standard SQL expressions that MicroStrategy Desktop does not support. MicroStrategy Desktop recognizes these containers as holding information. When you include a pass-through expression in the attribute, fact, or transformation expression box, the SQL engine recognizes custom SQL and treats it as a pass-through expression. The pass-through expression is then sent to the relational database as written.

String Patterns

There are several types of pass-through expressions accepted by MicroStrategy Desktop, but the only one that affects the Project Designer is `ApplySimple`. `ApplySimple` can be used in simple expressions using arithmetic operators. The string pattern for these functions contain parameter markers flagged by # characters. The parameter markers contained within the # flags are replaced in the engine by the actual expressions.

Constants are inserted as they appear, and object names are handled depending upon their type. A string pattern represents custom database-specific functions or other custom SQL. Since the functions are not analyzed or validated by the engine, there is no criteria on what they possibly contain, as long as the result returned is compatible with what the analytical engine expects.

Argument Types

The number of allowable arguments is variable. The engine does not verify arguments until the parameter markers are replaced at parsing. At parsing time, the engine searches for acceptable argument types. Acceptable argument types defined as contacts of the types used in MicroStrategy Desktop, and names of object types that are valid for the predefined functions of the same SQL type.
Upgrading

When upgrading a project, the custom SQL expression needs to be converted. The upgrade utility determines what type of Apply function is used. In the case of the Project Designer, only ApplySimple is applicable. A column is the only argument type considered.

If an expression fails during the first attempt at parsing, the upgrader considers it a custom expression. As a custom expression, the upgrader looks for syntax as shown in the following examples.

Syntax

These are examples of the syntax used for different databases. Refer to your specific database syntax when preparing pass-through expressions. This VMall example shows different expressions for three types of relational databases to determine Customer age.

- SQL Server Warehouse
  
  ApplySimple("datediff(YY,#0,getdate())", Cust_Birthdate)

- Oracle Warehouse
  
  ApplySimple("ROUND(MONTHS_BETWEEN(SYSDATE,#0)/12,0)", Cust_Birthdate)

- DB2 Warehouse
  
  ApplySimple("ROUND((days(current date) - days(#0))/365,0)", Cust_Birthdate)
Attribute Cardinality

Cardinality is the number of distinct elements an attribute has. This can be used for modeling and fine tuning purposes as explained in this appendix.

**Determining cardinality**

1. Determine the primary selection technique for the attribute.

   Consider phone numbers as an example to illustrate the idea of high cardinality attributes. Phone numbers might be selected by last name, first name, and, finally, address. Alternatively, phone numbers might be selected by country code, area code, city code, and number. If there is no meaningful breakdown of the attribute, an artificial one can be instituted. Elements can be grouped by numeric ID values into 100 subsets by grouping on the first two digits. Descriptions can be grouped alphabetically by the first character or group of characters, such as the way encyclopedia books are arranged by the first letter or pair of letters.

2. Design a grouping scheme.

   For all grouping techniques, every effort must be made to keep the cardinality of each group and sub-group at or below 1000 elements. This helps to ensure reasonable performance by limiting the volume of data returned. It presents the user with a manageable quantity of information. It also minimizes the number of groups through which the user must search to locate the desired data.

3. Design the lookup tables for the grouping attributes.

   It is not necessary to create lookup tables for the grouping attributes. However, it is a good idea to do so to maximize the speed with which the group attribute elements are returned. Each group attribute lookup table contains a numeric ID column as a primary key and a brief description column to simplify use. Each sub-group attribute lookup table can also contain the ID column of its parent group attribute. For example, if phone numbers are being grouped, the following tables might be created:
LookupCountryCode Cardinality: ~200
  COUNTRY_CODE_ID Primary Key
  COUNTRY_CODE_DESC Description (name of country)
LookupAreaCode Cardinality for each country: <1000
  AREA_CODE_ID Primary Key
  AREA_CODE_DESC Description of city, state, or region of state
  COUNTRY_CODE_ID Foreign Key
LookupCityCode Cardinality for each area code: <1000
  CITY_CODE_ID Primary Key
  CITY_CODE_DESC Description of city or area of city
  AREA_CODE_ID Foreign Key
LookupPartialNumber Cardinality for each city code: 100
  PARTIAL_NUMBER_ID Primary Key
  PARTIAL_NUMBER_DESC First two digits of the remaining phone number
  CITY_CODE_ID Foreign Key
LookupPhoneNumber Cardinality for each partial number: ~100
  PHONE_NUMBER_ID Primary Key
  PHONE_NUMBER_DESC The full phone number
  PARTIAL_NUMBER_ID Foreign Key

It is important to note that it is not beneficial to include ID columns of grandparents in the lookup tables unless the grouping attributes are used in analyses.

4. Populate the lookup tables.

Start with the lowest level attribute. For instance, in the preceding example, it is phone number. Temporarily duplicate the table, drop the original, recreate the table structure for the original, and repopulate the lookup table with the attribute ID and Description. Then execute the following five steps until the process is complete:

1. Populate the grouping attribute ID column. In the preceding example, it is PARTIAL_NUMBER_ID.
2. Test the cardinality of the attribute after the grouping has been applied. Use an SQL statement such as the following:

   SELECT COUNT(PHONE_NUMBER_ID), PARTIAL_NUMBER_ID
   FROM LookupPhoneNumber
   GROUP BY PARTIAL_NUMBER_ID;

3. Check the resulting counts to ensure that they are reasonable. For the above example, none of them exceeds 100. If any grouping attribute elements contain more than 1000 child elements, consider redesigning the grouping algorithm.

4. Populate the parent attribute ID and description columns where applicable.

5. If the parent attribute has a grouping attribute, repeat the procedure with the parent attribute.


1. Launch MicroStrategy Architect and open the existing project.

2. Proceed to the Warehouse tab.

3. Add the new lookup tables to the list of selected tables.

4. Proceed to the Attributes tab.

5. Select the dimension containing the attribute with high cardinality.

6. Create one new attribute for each grouping attribute, starting with the lowest and proceeding upwards. For each new attribute, specify the lookup table, ID column, description column (where applicable), and child attribute.

7. When all grouping attributes have been defined, go to the Components tab.

8. Modify the hierarchy for the existing attribute to include the parent grouping attributes in order.

9. Save the project and exit MicroStrategy Architect.

You have now created a hierarchical selection method for your high-cardinality attribute.
The SQL Engine uses filters to determine the physical tables that need to be retrieved. The engine uses table joins on a partition-mapping table to obtain this information.

When the SQL engine finds a table for a fact and detects that it is a logical partition-mapping table, it generates a pre-query SQL. This pre-query SQL is generated by creating a report that uses the Custom Group from the partition-mapping table as a template and the original report filter as a filter. From the pre-query, the SQL Engine receives a list of physical tables to be used. The engine generates the SQL for each table. The tables are assumed to have a homogenous logical structure and to apply the partitioning logic.

The SQL Engine also generates the SQL queries for the different physical tables that need to be queried according to the results of pre-query SQL execution. If the partitioning is homogenous, meaning that all the tables have the same key, and there is no heterogeneous column name/definition for attributes/facts across the different tables, the SQL Engine can generate just one SQL. This SQL is incomplete and is used as a template SQL. The Query Engine can fill the holes in the incomplete SQL and cause the complete SQL to be executed.

The tables are heterogeneously partitioned, meaning that the tables have different keys, or the tables have heterogeneous column names/definitions for attributes/facts. In this case the SQL Engine has to generate SQL for all tables that can possibly be selected by the pre-query SQL. The Query Engine then has to select the appropriate tables according to the results of pre-query SQL execution and execute the SQL that corresponds to the different tables.

The SQL Engine also generates any additional SQL needed to handle second aggregation logic.
In all supported warehouse platforms other than Microsoft Access, MicroStrategy 7 uses SQL statements to query the relational database management system (RDBMS) catalog table to obtain warehouse catalog information. This includes tables, columns, and their data types. The catalog SQL statements vary from platform to platform and can be customized according to the characteristics of the specific warehouse.

Customizing Catalog SQL Statements

The MicroStrategy 7 Warehouse Catalog Browser can be configured to read the catalog information in one- or two-pass SQL mode. In two-pass SQL mode, it first reads only the tables from the warehouse. The structure of individual tables is read only when the table is selected. This is the recommended option for interactive warehouse catalog building because no unnecessary catalog information is read from the warehouse, which increases process speed. One-pass SQL mode, on the other hand, reads all the tables and columns in one SQL statement. This option is recommended only if the catalog SQL is well customized to limit the amount of data returned by it.

The two retrieval options use different catalog SQLs, but both can be customized in the Warehouse Catalog Options dialog. In the following text, the name Catalog Table SQL refers to the catalog SQL to retrieve the tables in the warehouse; that is, the first SQL used in a two-pass catalog retrieval. The name Full Catalog SQL refers to the SQL used to read all the tables and columns in one pass. To customize a catalog SQL, you must understand several important concepts.

The table name space

In a typical RDBMS platform, a table name does not uniquely identify it in a particular warehouse database installation. A table name space is a partition of the warehouse installation in which table names are unique. Depending on the type of the RDBMS, this name space can be the name of the warehouse database, the owner of the table, or a combination of both database and owner. In both the Catalog Table SQL and Full Catalog SQL,
a name space provides each table name so that you can avoid mixing up same-named tables from different name spaces. However, the table name space is optional. A customized catalog SQL can omit the name space if duplicate table names do not present a problem in the warehouse database.

**SQL template strings and incomplete catalog SQL**

The default system Catalog SQL can contain certain template strings that can be resolved at run time, or must be completed manually by the user. These templates are listed here:

`#LOGIN_NAME#:` This template is automatically replaced at run time with the login name used to connect to the warehouse. The user can leave this template in the customized SQL if he or she wants the catalog SQL to yield different results depending on the warehouse login used. Otherwise, this template is replaced with the name of the warehouse user who owns the warehouse tables of interest.

`#?Database_Name?, #?Schema_Name?:` This is a catalog SQL that contains a template string enclosed in `#? .... ?#` is an incomplete SQL that must be completed by the user before it can be run. `#?Database_Name?` must be replaced with the name of the warehouse database containing the warehouse tables. `#?Schema_Name?` must be replaced with the name of the schema in which the warehouse tables reside.

**Structure of Catalog Table SQL**

A catalog table SQL is expected to return two columns, one identifying the name space of the table and the other the name of the table. In the case where a name space is not provided, only the table name column is required. Each row of the SQL result must uniquely identify a table. Duplicates are not allowed. The column that identifies table name space uses the SQL column alias `NAME_SPACE`. The column that identifies the table name has the alias `TAB_NAME`. The following example is the default Catalog Table SQL for Oracle 8.0:

```
SELECT DISTINCT OWNER NAME_SPACE, TABLE_NAME
TAB_NAME
FROM ALL_TAB_COLUMNS
WHERE OWNER = '#LOGIN_NAME#'
```
Structure of the full catalog SQL

A full catalog SQL is expected to return between five and seven columns, depending on the RDBMS platform and the customization. The following aliases are required to identify each column returned:

- NAME_SPACE (optional): the table name space
- TAB_NAME (required): name of the table
- COL_NAME (required): name of the column
- DATA_TYPE (required): a string or a number that identifies the major data type of the column
- DATA_LEN (required): a number that describes the length or size of the column data
- DATA_PREC (optional): a number that describes the precision of the column data
- DATA_SCALE (optional): a number that describes the scale of a floating point column data

The Full Catalog SQL must return its rows ordered first by NAME_SPACE, if available, and then by TAB_NAME.

The following example is the default Full Catalog SQL for Microsoft SQL Server 7.0:

```sql
SELECT U.name NAME_SPACE, T.name TAB_NAME, C.name COL_NAME, C.type DATA_TYPE,
       C.length DATA_LEN, C.prec DATA_PREC, C.scale DATA_SCALE
FROM sysobjects T, syscolumns C, sysusers
WHERE T.id = C.id and T.type in ('U', 'V')
   AND T.uid = U.uid
ORDER BY 1, 2
```

Default Warehouse Catalog SQL

The following table shows the default warehouse catalog SQLs used by MicroStrategy 7 for each supported warehouse platform. The user is encouraged to consult this table before writing their own customized catalog SQL.
<table>
<thead>
<tr>
<th>RDBMS</th>
<th>Default Catalog Table SQL (1) and Full Catalog SQL (2)</th>
</tr>
</thead>
</table>
| IBM DB2 OS/390   | `SELECT TBCREATOR NAME_SPACE, TBNAME TAB_NAME
FROM SYSIBM.SYSCOLUMNS
WHERE TBCREATOR='#LOGIN_NAME#'
SELECT TBCREATOR NAME_SPACE, TBNAME TAB_NAME, NAME COL_NAME, COLTYPE DATA_TYPE, LENGTH DATA_LEN, SCALE DATA_SCALE
FROM SYSIBM.SYSCOLUMNS
WHERE TBCREATOR='#LOGIN_NAME#'` ORDER BY 1, 2                                                                 |
| IBM DB2 AS/400   | *Note: requires manual replacement of template string #?Schema Name?#*
SELECT DISTINCT SYSTEM_TABLE_SCHEMA NAME_SPACE, TABLE_NAME TAB_NAME
FROM QSYS2.SYSCOLUMNS
WHERE TABLE_OWNER = '#LOGIN_NAME#'
AND SYSTEM_TABLE_SCHEMA = '#?Schema_Name??#'
SELECT SYSTEM_TABLE_SCHEMA NAME_SPACE, TABLE_NAME TAB_NAME, COLUMN_NAME COL_NAME, DATA_TYPE DATA_TYPE, LENGTH DATA_LEN, NUMERIC_SCALE DATA_SCALE
FROM QSYS2.SYSCOLUMNS
WHERE TABLE_OWNER = '#LOGIN_NAME#'
AND SYSTEM_TABLE_SCHEMA = '#?Schema_Name??#'
ORDER BY 1, 2                                                                 |
| IBM DB2 UDB      | `SELECT DISTINCT TBCREATOR NAME_SPACE, TBNAME TAB_NAME
FROM SYSIBM.SYSCOLUMNS
WHERE TBCREATOR='#LOGIN_NAME#'
SELECT TBCREATOR NAME_SPACE, TBNAME TAB_NAME, NAME COL_NAME, COLTYPE DATA_TYPE, LENGTH DATA_LEN, SCALE DATA_SCALE
FROM SYSIBM.SYSCOLUMNS
WHERE TBCREATOR='#LOGIN_NAME#'` ORDER BY 1, 2                                                                 |
<table>
<thead>
<tr>
<th>RDBMS</th>
<th>Default Catalog Table SQL (1) and Full Catalog SQL (2)</th>
</tr>
</thead>
</table>
| Informix 7.x, 8.x, 9.x | SELECT DISTINCT owner NAME_SPACE, tabname TAB_NAME FROM SYSTABLES WHERE tabid >= 100 AND tabtype IN ('T', 'V')  
|                     | SELECT T.owner NAME_SPACE, T.tabname TAB_NAME, C.colname COL_NAME,                                                                                                                  |
|                     |   C.coltype DATA_TYPE, C.collength DATA_LEN FROM SYSTABLES T, SYSCOLUMNS C WHERE T.tabid = C.tabid  
|                     | AND T.tabtype IN ('T', 'V', 'S') ORDER BY 1, 2                                                                                                                                 |
| Oracle 7.3.x, 8.0.x, 8i | SELECT DISTINCT OWNER NAME_SPACE, TABLE_NAME TAB_NAME FROM ALL_TAB_COLUMNS WHERE OWNER = '#LOGIN_NAME#'  
|                     | SELECT OWNER NAME_SPACE, TABLE_NAME TAB_NAME, COLUMN_NAME COL_NAME,                                                                                                                  |
|                     |   DATA_TYPE DATA_TYPE, DATA_LENGTH DATA_LEN,                                                                                                                                       |
|                     |   DATA_PRECISION DATA_PREC, DATA_SCALE DATA_SCALE FROM ALL_TAB_COLUMNS WHERE OWNER = '#LOGIN_NAME#' ORDER BY 1, 2  
| Red Brick 5.x, 6.x  | SELECT DISTINCT CREATOR NAME_SPACE, NAME TAB_NAME FROM RBW_TABLES WHERE ID > 0 AND CREATOR='#LOGIN_NAME#'  
|                     | SELECT T.CREATOR NAME_SPACE, T.NAME TAB_NAME, C.NAME COL_NAME,                                                                                                                     |
|                     |   C.TYPE DATA_TYPE, C.LENGTH DATA_LEN,                                                                                                                                             |
|                     |   C.PRECISION DATA_PREC, C.SCALE DATA_SCALE FROM RBW_TABLES T, RBW_COLUMNS C WHERE T.ID = C.TID  
<p>|                     | AND T.ID &gt; 0 ORDER BY 1, 2                                                                                                                                                         |</p>
<table>
<thead>
<tr>
<th>RDBMS</th>
<th>Default Catalog Table SQL (1) and Full Catalog SQL (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server 7.0</td>
<td>SELECT DISTINCT U.name NAME_SPACE, T.name TAB_NAME FROM sysobjects T, sysusers U WHERE T.uid = U.uid AND T.type IN ('U', 'V')</td>
</tr>
<tr>
<td></td>
<td>SELECT U.name NAME_SPACE, T.name TAB_NAME, C.name COL_NAME, C.type DATA_TYPE, C.length DATA_LEN, C.prec DATA_PREC, C.scale DATA_SCALE FROM sysobjects T, syscolumns C, sysusers U WHERE T.id = C.id and T.type in ('U', 'V') AND T.uid = U.uid ORDER BY 1, 2</td>
</tr>
<tr>
<td>Sybase Adaptive Server 11.x, 12.x</td>
<td>SELECT DISTINCT U.name NAME_SPACE, T.name TAB_NAME FROM sysobjects T, sysusers U WHERE T.uid = U.uid AND T.type IN ('U', 'V')</td>
</tr>
<tr>
<td></td>
<td>SELECT U.name NAME_SPACE, T.name TAB_NAME, C.name COL_NAME, C.type DATA_TYPE, C.length DATA_LEN, C.prec DATA_PREC, C.scale DATA_SCALE FROM sysobjects T, syscolumns C, sysusers U WHERE T.id = C.id and T.type in ('U', 'V') AND T.uid = U.uid ORDER BY 1, 2</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Default Catalog Table SQL (1) and Full Catalog SQL (2)</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Sybase IQ 12   | ```sql
SELECT DISTINCT U.name NAME_SPACE, T.table_name TAB_NAME
FROM systable T, sysusers U
WHERE T.creator = U.uid
AND T.table_type IN ('BASE', 'VIEW')
SELECT U.name NAME_SPACE, T.table_name TAB_NAME, C.cname COL_NAME,
    C.coltype DATA_TYPE, C.length DATA_LEN,
    C.syslength DATA_SCALE
FROM systable T, syscolumns C, sysusers U
WHERE T.table_name = C.tname and T.table_type in ('BASE', 'VIEW')
AND T.creator = U.uid
ORDER BY 1, 2
``` |
| Tandem NonStop SQL | ```sql
SELECT DISTINCT U.name NAME_SPACE, T.name TAB_NAME
FROM sysobjects T, sysusers U
WHERE T.uid = U.uid
AND T.type IN ('U', 'V')
SELECT U.name NAME_SPACE, T.name TAB_NAME, C.name COL_NAME,
    C.type DATA_TYPE, C.length DATA_LEN
FROM sysobjects T, syscolumns C, sysusers U
WHERE T.id = C.id and T.type in ('U', 'V')
AND T.uid = U.uid
ORDER BY 1, 2
``` |
<table>
<thead>
<tr>
<th>RDBMS</th>
<th>Default Catalog Table SQL (1) and Full Catalog SQL (2)</th>
</tr>
</thead>
</table>
| NCR Teradata | *Note: requires manual replacement of template string #?Database_Name?#*  
SELECT DISTINCT DatabaseName NAME_SPACE, TableName TAB_NAME  
FROM DBC.TABLES  
WHERE DatabaseName = '#?DATABASE_NAME?#'  
SELECT DatabaseName NAME_SPACE, TableName TAB_NAME,  
   ColumnName COL_NAME, ColumnType DATA_TYPE,  
   ColumnLength DATA_LEN, DecimalTotalDigits DATA_PREC,  
   DecimalFractionalDigits DATA_SCALE  
FROM DBC.COLUMNS  
WHERE DatabaseName = '#?DATABASE_NAME?#'  
ORDER BY 1, 2 |
The Query Engine executes the pre-query SQL and retrieves the results. The results indicate which custom group elements from the partition-mapping table are valid for the filtering conditions for this pass. This is different from the warehouse partitioning approach where a list of tables from the pre-query SQL is returned.

Once the Query Engine knows the different custom group elements that are valid for this particular pass, it uses a lookup structure to retrieve the list of tables that map to the valid custom group elements. The engine then needs to retrieve the SQL for the different tables in the list.
MicroStrategy Drill Map Editor 7.0 allows you to create fully customized drill paths to simplify navigation through data and control the drill paths available to users. This ability is also known as investigative workflow. Drilling allows you to look at specific data at levels other than that of the originally displayed grid or graph.

For example, you create a drill map composed of several custom drill paths, then you associated the drill map with an attribute. Whenever a user drills on that attribute in a report, the Drill dialog displays only the custom drill paths that you set up.

**Note:** It is expected that you are familiar with the MicroStrategy 7 drilling functionality and COM objects. For more information on these topics, see the *MicroStrategy 7 Analyst Guide*.

### Drill map associations

Drill maps enable you to completely control and customize all the different drilling actions for different objects at many levels:

- **Project level:** A drill map can be associated to a project and is then used by default in all of the reports in that project.
- **Report/template level:** A drill map can be associated to an individual report or template, overriding the default project drill map for all drilling actions on that report or template.
- **Template unit level:** A drill map can be associated with individual objects such as attributes, consolidations, custom groups, and the metric template unit (that is, the set of metrics in a report) in a report or template. It is used only when the drilling action is generated from that object. The object’s default drill map is used in all other cases.
• Object level: A drill map can be associated with individual objects such as attributes, consolidations, and custom groups so that whenever the object is used in a report or template, the customized drill map is used instead of the default project one.

Once a drill map has been created with the Drill Map Editor, you can associate it with multiple objects at different levels.

Two types of conflict can occur when you try to associate objects to a drill map. The selected object may already be associated with a drill map, or the selected report may contain a link to a template. In the first case, you can choose whether to proceed with the association or replace the old association with the new drill map. If you replace it, the old association will be lost.

A report linked to a template cannot be associated directly to a drill map. Your options are to cancel the association or continue with the direct association. If you choose to continue, all of the reports that reference the template are modified. Before selecting this option, ensure that this does not affect your reports.

### Drill path types

Drill maps are basically collections of drill paths, which are paths that determine which objects are presented by the MicroStrategy Desktop and MicroStrategy Web products as possible destinations of a particular drilling action. A project can contain as many different drill maps as desired. They are stored and manipulated as any other object. The Drill Map Editor supports three different types of drill paths:

- **Hierarchy**, which is created based on a hierarchy that already exists in the project. Custom paths are created for each attribute pair in the hierarchy, based on the pair’s relationship.

- **Map**, which is created based on an existing drill map. All of the drill paths of the referenced drill map are included in the destination drill map.

- **Custom**, the most powerful type of drill path, is based on a drill action. This allows you to create, view, and modify custom paths that specify a particular destination object.
Custom drill paths

The Drill Map Editor allows you to define a drill action and associate it to a custom drill path. A drill action consists of three main parameters:

• drill action type
• filtering options, which allows you to apply or ignore the filter for both the original report’s filter and the drilling filter, which is the filter created on the fly by the selections made by the user before drilling
• keep original object setting, which allows you to specify whether the original object appears in the destination template

Custom drill paths allow you to define the following drilling actions:

• Drill to parent: The destination object, contrary to the name, does not have to be related to the original object. The destination is shown as part of the Drill Up menu. Supported destination objects are attributes, consolidations, and custom groups.

• Drill to child: Again, the destination object does not have to be related to the original object. The destination is shown as part of the Drill Down menu. Supported destination objects are attributes, consolidations, and custom groups.

• Drill to unit: The destination object does not have to be related to the original object. The destination is shown as part of the Other Directions menu. Supported destination objects are attributes, consolidations, and custom groups.

• Drill to template: This actually replaces the current report’s template with another base on the user’s drilling selection. Through the Drill Map Editor, users browse the project’s folders to select the template to use as a destination for the drilling action.

---

**Note:** The keep original object setting does not apply to the drill to template action.
Drill Map Browser Interface

What is it?
The Drill Map Browser allows you to view all the different drill maps available in the project. It is the main window of Drill Map Editor.

How do I access it?
You can access the Drill Map Browser by logging in to MicroStrategy Drill Map Editor and opening a project.

What can I do with it?
Through the Drill Map Browser, you can:
• Add a new drill map
• Associate a new default drill map to the project
• Create a new folder
• Cut, copy, and paste drill maps
• Delete a drill map
• Edit a drill map
• Rename a drill map
• View a drill map’s properties
For information on completing one of the above tasks, refer to the following topic:
Drill Map Browser: How do I...

What should I know before I use it?
Before you begin using the Drill Map Browser, you need to be familiar with:
• the concept of drilling in MicroStrategy 7
• drill path associations
For more information on the above topics, refer to the Microstrategy 7 Analyst Guide and the following topic:
• Drill map associations
Drill Map Browser layout

Drill Map Browser functions similar to the MicroStrategy Desktop window. It includes:

- the **Location** box across the upper part of the window, showing the folder currently selected; the default is the drill map root directory
- a **browser window** listing all the available drill maps and folders in the project and allowing navigation through them
- the **Description** box displays the description of the currently selected drill map or folder
- the **Default Project Drill Map** box indicates which drill map is currently associated with the project; it also allows you to associate a new default drill map to the project
- a **status bar**, located at the bottom of the window, that contains the following information:
  ◊ number of objects in the list
  ◊ project source name of the current connection
  ◊ project name
  ◊ current user and access rights
- a **menu bar** with the standard Windows NT menus File, Edit, View, and Window
- buttons with the following functionality:
  ◊ **New** accesses the Drill Map Editor with an empty definition
  ◊ **Delete** deletes the selected item
  ◊ **Edit** accesses the Drill Map Editor with the selected drill map’s definition
  ◊ **Properties** displays the properties dialog for the selected object
  ◊ **Go one level up** accesses the folder one level above the currently selected folder
Drill Map Editor Interface

What is it?

Drill Map Editor allows you to create and edit custom drill paths in a selected drill map.

How do I access it?

The Drill Map Editor is displayed when you create or edit a drill map.

What can I do with it?

Through the Drill Map Editor, you can:
  • Access the Drill Map Associations dialog to associate objects to the drill map
  • Add a custom drill path
  • Add a hierarchy drill path
  • Add a map drill path
  • Cut, copy, and paste drill paths
  • Delete a drill path
  • Modify a drill path
  • Modify a drill path’s importance
  • Move drill paths up and down in the drill path list

For information on completing one of the above tasks, refer to the following topic:

Drill Map Editor: How do I…

What should I know before I use it?

Before you begin using the Drill Map Editor, you need to be familiar with:
  • the difference between drill maps and drill paths
  • the different kinds of drill paths
  • drill path associations
For more information on the above topics, refer to the following topics:

- Drill map associations
- Drill path types
- Drilling actions of custom drill paths

**Drill Map Editor layout**

The Drill Map Editor functions similar to a standard Windows editor. It includes:

- the **drill path list**, which contains all the different drill paths that are part of the drill map definition
- the **drill path definition**, which displays, and allows changes to, the definition of the drill path currently selected in the drill path list
- a **menu bar** that, in addition to the standard Windows NT menus File and Edit, includes **Drill Paths** and **Tools** menus that contain items specific to the Drill Map Editor
- a **toolbar** that provides, among other options, the capability to save the drill map and move the selected drill path in the list
- a **status bar**, located at the bottom of the window, that contains information and directions
- buttons with the following functionality:
  - Clear all deletes all the drill paths from this drill map
  - Add creates a new drill path
  - Remove deletes the selected drill path

**Drill path list**

The **drill path list** contains all the different drill paths that are part of the drill map definition. The list has two columns, **Name** and **Importance**. Besides the drill path name, the Name column contains an icon representing the drill path type. High importance displays with an icon as well.

At the bottom of the drill path list there are three buttons. They allow you to clear all the drill paths from the drill map, add a new drill path, or delete the selected drill path.
Drill path definition

The **drill path definition** displays the definition of the currently selected drill path as you click through the different paths in the drill path list. Since the drill map definition requirements change depending on the type of drill path is selected, the appearance of the drill box definition changes.

For hierarchy and map drill paths, the following information is displayed:

- **Path type:** whether it is hierarchy or map
- **Hierarchy/Map:** the hierarchy or map object selected
- **Description:** the description of the hierarchy or map object selected

If a custom drill path is displayed, the interface displays different information:

- **Path type:** set to Custom.
- **Name:** the name of the drill path.
- **Description:** the description of the selected object.
- **Set Name:** the grouping name. If the Set Names of two different drill actions are the same, they are grouped together in the Drill option in MicroStrategy Desktop.
- **Action Type:** action type of the drill path; selected from a pull-down list.
- **Action Destination:** the object that is the destination of the drill action when drilling is applied to the selected path in MicroStrategy Desktop. For example, if the Action Type is set to Drill to template and the Action Destination is a specific template, every time you use this drill path in MicroStrategy Desktop, a new report with the selected template is generated.
- **Apply user filtering conditions:** if checked, uses the filter created on the fly by the selections made by the user before drilling.
- **Apply original report filtering conditions:** if checked, uses the filter from the original report.
- **Keep parent object:** if checked, the original object appears in the destination template.
Menu bar

The Drill Paths and Tools menu bar options contain items unique to the Drill Map Editor. The tables that follow describe the contents of each.

Drill Paths menu options

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Add a new drill path</td>
</tr>
<tr>
<td>Remove</td>
<td>Remove the selected drill path</td>
</tr>
<tr>
<td>Clear All</td>
<td>Clear the entire collection of drill paths</td>
</tr>
<tr>
<td>Low</td>
<td>Set the selected drill path’s importance to low</td>
</tr>
<tr>
<td>Medium</td>
<td>Set the selected drill path’s importance to medium</td>
</tr>
<tr>
<td>High</td>
<td>Set the selected drill path’s importance to high</td>
</tr>
<tr>
<td>Move Up</td>
<td>Move the current drill path up a level</td>
</tr>
<tr>
<td>Move Down</td>
<td>Move the current drill path down a level</td>
</tr>
<tr>
<td>Move to Top</td>
<td>Move the current drill path to the top level</td>
</tr>
<tr>
<td>Move to Bottom</td>
<td>Move the current drill path to the bottom level</td>
</tr>
</tbody>
</table>

Note: Only one drill path can be set to high importance.

Tools menu options

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drill Map Associations...</td>
<td>Display the Drill Map Associations dialog to associate objects to the selected drill map. Valid objects include projects, attributes, consolidations, custom groups, hierarchies, templates, and template units.</td>
</tr>
</tbody>
</table>

Drill Map Editor Interface 247
Drill Map Associations Dialog

**What is it?**

The Drill Map Associations dialog allows you to choose the objects to be associated to the drill map.

**How do I access it?**

You can access the Drill Map Associations dialog by choosing **Drill Map Associations** from the **Tools** menu of the Drill Map Editor. To associate a new default drill map to a project, you can click the **Browse** button next to the Default Drill Map box in the Drill Map Browser.

**What can I do with it?**

Through the Drill Map Associations dialog, you can:

- Associate an object to a drill map
- Resolve a conflict during drill map association

For information on completing one of the above tasks, refer to the following topic:

Drill Map Associations Dialog: How do I...

**What should I know before I use it?**

Before you begin using the Drill Map Associations dialog, you need to be familiar with:

- drill map associations

For more information on the above topics, refer to the following topic:

Drill map associations

**Drill Map Associations dialog layout**

The Drill Map Associations dialog includes:

- the **Available Objects** tree, on the left, that displays the available objects
- the **Selected Objects** tree, on the right, that displays the objects already associated with the drill map
Drill Map Browser: How do I...

Add a new drill map

Steps
1. Do one of the following:
   ◊ Right-click on a blank area of the browser window and choose New Drill Map.
   ◊ Click New.
   ◊ Select New from the File menu. From the list of options, choose Drill Map.
2. The Drill Map Editor is displayed with a blank definition.
3. If you want, you can add a drill path. See the following topic:
   Add a new drill path
4. Click Save and Close to return to the Drill Map Browser.

Associate a new default drill map to the project

Steps
1. Click the Browse button next to the Default Project Drill Map field. An Open dialog appears, displaying the drill maps in the current project.
2. Select a drill map to use as the default for the project.
3. Click OK.

Create a new folder

Steps
1. From the File menu, choose New, and then Folder.
2. Enter a name for the new folder.
Cut, copy, and paste a drill map

**Steps**

1. Do one of the following:
   ∘ Right-click the drill map and select Cut or Copy.
   ∘ Select the drill map and choose Cut or Copy from the Edit menu.

2. Do one of the following:
   ∘ Right-click the destination and select Paste.
   ∘ Select the destination and choose Paste from the Edit menu.

Delete a drill map

**Step**

Do one of the following:

- Right-click the drill map to delete and select Delete.
- Select the drill map to delete and click Delete.
- Select the drill map to delete and select Delete from the File menu.
- Select the drill map to delete and press the Delete key.

Edit a drill map

**Steps**

1. Do one of the following:
   ∘ Double-click the drill map to edit.
   ∘ Right-click on the drill map and choose Edit.
   ∘ Select the drill map and click Edit.
   ∘ Select the drill map. Choose Edit from the File menu.

2. The Drill Map Editor is displayed.

3. Modify the drill path(s). See the following topic:
   Modify a drill path

4. Click Save and Close to return to the Drill Map Browser.
Log in to the Drill Map Browser

**Steps**
1. At the Login dialog, enter your Login id and Password.
2. Select a project source from the list of Available Project Sources.
3. Select a project from the list of Available Projects.
4. Click Open. The Drill Map Browser interface is displayed.

Rename a drill map

**Steps**
1. Do one of the following:
   ◊ Right-click the drill map to rename and select Rename.
   ◊ Select the drill map. Choose Rename from the File menu.
2. Enter the new name of the drill map.

View a drill map’s properties

**Step**
Do one of the following:
• Right-click the drill map to access and select Properties.
• Select the drill map to access and click Properties.
• Select the drill map to access and select Properties from the File menu.
Drill Map Editor: How do I...

Access the Drill Map Associations dialog

1. In the Drill Map Editor, select Drill Map Associations from the Tools menu.
2. Associate an object to the drill map. See the following topic: Associate an object to the drill map

Add a custom drill path

Steps

1. Click Add.
2. Select Custom from the Path type pull-down menu.
3. Enter a Name and a Description.
4. Enter a Set Name or select one of the recently accessed set names from the pull-down menu.
5. Select an Action type from the pull-down menu.
6. Click the Browse button next to the Destination field. An Open dialog appears, displaying available objects in the current project.
7. Select an object to base the drill path on.
8. Check Apply user filtering conditions if you want to use the filter created on the fly by the selections made by the user before drilling.
9. Check Apply original report filtering conditions if you want to use the filter from the original report.
10. Check Keep parent object if you want the original object to appear in the destination template.

Note

The keep parent object setting does not apply to the drill to template action.
11. The default Importance setting is Low. To change it, do one of the following:
   ◊ Click the drill path once to change the setting to Medium and twice to change to High.
   ◊ Select the drill path, and choose the new Importance setting from the Drill Paths menu.

Add a hierarchy drill path

Steps
1. Click Add.
2. Select Hierarchy from the Path type pull-down menu.
3. Click the Browse button next to the Hierarchy field. An Open dialog appears, displaying the hierarchy objects in the current project.
4. Select a hierarchy object to base the drill path on.
5. Click OK.
6. The Description is filled in automatically.
7. The default Importance setting is Low. To change it, do one of the following:
   ◊ Click the drill path once to change the setting to Medium and twice to change to High.
   ◊ Select the drill path, and choose the new Importance setting from the Drill Paths menu option.

Add a map drill path

Steps
1. Click Add.
2. Select Map from the Path type pull-down menu.
3. Click the Browse button next to the Map field. An Open dialog appears, displaying the drill maps in the current project.
4. Select a drill map to base the drill path on.
5. Click OK.
6. The Description is filled in automatically.
7. The default **Importance** setting is Low. To change it, do one of the following:

- Click the drill path once to change the setting to Medium and twice to change to High.
- Select the drill path, and choose the new Importance setting from the **Drill Paths** menu option.

**Cut, copy, and paste a drill path**

**Steps**

1. Do one of the following:

- Right-click the drill path and select **Cut** or **Copy**.
- Select the drill path and click **Cut** or **Copy**.
- Select the drill path and choose **Cut** or **Copy** from the **Edit** menu.

2. Do one of the following:

- Right-click the destination and select **Paste**.
- Select the destination and click **Paste**.
- Select the destination and choose **Paste** from the **Edit** menu.

**Delete a drill path**

**Steps**

Do one of the following:

- Right-click the drill path to delete and select **Delete**.
- Select the drill path to delete and click **Remove**.
- Select the drill path to delete and select **Remove** from the **Drill Paths** menu.

**Modify a drill path**

**Steps**

1. Select the drill path to be changed. The drill path definition, on the right side, changes to reflect the drill path’s information.

2. Make the changes to Path type, etc.
Modify a drill path’s importance

Steps
Do one of the following:
• Double-click the drill path to cycle through the importance levels.
• Right-click the drill path and select the importance level.
• Select the drill path. Choose Drill Paths from the menu bar, then select the importance level.

Note
Only one drill path can be set to high importance.

Move a drill path up or down in the drill path list

Steps
Do one of the following:
• Right-click the drill path and select Move Up, Move Down, Move to Top, or Move to Bottom to move the drill path to its new position.
• Select the drill path and use the Move Up, Move Down, Move to Top, or Move to Bottom buttons to move the drill path to its new position.
• Select the drill path. Choose Drill Paths from the menu bar, then select Move Up, Move Down, Move to Top, or Move to Bottom to move the drill path to its new position.
Drill Map Associations Dialog: How do I...

Associate an object to a drill map

Steps

1. Browse through the folders to find the object to associate to the drill map.
2. Do one of the following:
   ◊ Double-click the object to associate.
   ◊ Select the object to associate and click the Add button.

Note

If the object is a report, you can associate the drill map to the whole report definition or only to a specific template unit. Check the box next to the object to associate it to the whole report.

3. If a conflict occurs, the Conflict Resolution dialog displays. See the following topic for more details:
   Resolve a conflict during object association

4. Click OK to return to the Drill Map Editor.

Resolve a conflict during object association

Steps

1. Double-click the conflict to resolve. The Conflict Details dialog displays.
2. If the selected object is already associated with a drill map, select whether to proceed with the association or replace the old association with the new drill map. If you replace it, the old association will be lost.
3. If the selected report contains a link to a template, it cannot be associated directly to a drill map. Select whether to directly associate the drill map to the template or cancel the association.
Note

If you choose to directly associate the drill map to the template, all of the reports that reference this template are modified. Ensure that this does not affect your reports before choosing this option.

3. Click OK to return to the Conflict Resolution dialog. The Conflict State of the selected conflict now displays as Resolved.

4. Once all conflicts have been resolved, click Continue to finish the association.
Index

A
Attribute Cardinality
definition 223
Attribute Creation Wizard
accessing 91
compound 94
create attributes 126
description column 93
ID Column 93
layout 92
lookup table 94
relationship definition 94
rules 93
uses 91
Attribute Editor
accessing 83
children tab 89
column alias tab 87
create attribute form 87
create form expression 88
display tab 89
forms tab 84
layout 84
modify attribute form 88
modify form expression 88
parents tab 89
uses 83
attribute form groups
creating 152
attribute forms
creating 149
modifying 162
modifying display 162
removing 163, 164
removing from browsing 163
removing from reporting 164
attribute relationships
joint child 37, 153
many-to-many 37
one-to-many 36
one-to-one 35
attributes
adding forms 149
compound 34, 86
compound key 34
compound keys 86
creating 153, 156
creating children 158
creating column alias 158
creating expressions 155
creating form categories 155
creating forms 155
creating parents 159
definition 25
derived 28
elements 29
expressions 33
form expressions 85
form groups 32
forms 31
implicit 28
key 85
mapping expressions 161
modifying 161
modifying form display 162
modifying forms 162
relationships 34, 35, 36, 37
rules 126
setting display properties 164
source tables 85
ungrouping 86
ungrouping compound 159
ungrouping form group 160
viewing properties 165

C
column alias
creating 150, 172
modifying 178
compound key
creating 151
copyright ii
D

Data Explorer
definition 54
data modeling
goal 75
data slice
modifying 194
Data Slice Editor
accessing 115
Default Warehouse Catalog SQLs 231
drill map associations 239
Drill Map Associations dialog
accessing 248
uses 248
Drill Map Browser
accessing 242
uses 242
Drill Map Editor 239
accessing 244
uses 244
DSS Agent 5.x
upgrading 68
duplicate project
create 132
source 131

F

Fact Creation Wizard
accessing 99
column selection 100
layout 100
rules 100
uses 99
Fact Editor
accessing 95
column alias 97
definition 96
extensions 97
layout 96, 97
uses 95
fact expressions
assigning mapping methods 168
creating formulas 170
mapping 177
unmapping 180
fact extensions
creating 168
deleting 173
modifying 179
facts
adding expressions 167
column alias 43, 46
creating 169, 171
cross product 48
definition 41
degradation and allocation 47
deleting extensions 173
derived 42
disallow 48
expressions 45
extending levels 175
fact definition 43, 44
fact entry level 42
fact relation 47
heterogeneous fact columns 49
implicit 42
Level Extension Wizard 109
level extensions 43, 46
lowering levels 176
mapping expressions 177
modifying expressions 179
modifying fact extensions 179
rules 126
simple 42
table relation 47
unmapping expressions 180
viewing properties 181
facts expressions
modifying 179
form expressions
automatic 85
manual mapping 85

H

hierarchies
adding attributes 183, 184
adding to system hierarchy 183
browsing 56
creating 185
Index

defining attributes 186
definition 51
deleting 186
drilling 58
entry point 56
filters 55
limited hierarchy 55
locked hierarchy 55
modifying data slices 194
removing attribute filters 187
removing attributes 187
setting display properties 188
setting drilling 189
structure 52
system hierarchy 53
user hierarchies 54
Hierarchy Editor
accessing 103
attributes 105
browse attributes tab 105
element display tab 106
filters tab 106
layout 105
uses 104

I
Investigative workflow 239

J
joint child
creating 153

L
Level Extension Wizard
accessing 109
extended attributes 110
join direction 111
join typ 111
uses 109
logical model
developing 14

M
MicroStrategy 7 Desktop
project 68
MicroStrategy Agent 6.x
upgrading 68

O
objects
attributes 25

P
Partition Editor
accessing 113
adding data slices 193
uses 113
partition mapping
data slice 62
definition 61
homogenous 62
logical size 114
metadata 62, 115
Partition Editor 113
Query Engine Logic 237
SQL Engine Logic 227
tables 115
types 61
warehouse 61
partitions
adding data slices 193
creating metadata partitions 194
setting mapping size 195
pass through function
Apply Simple 28, 43
Print
project schema 197
Project Creation Assistant
accessing 123
create facts 125
create project 125
creating attributes 200
creating projects 202
removing attributes 205
removing hierarchies 205

Index 261
select tables 125
selecting tables 206
uses 123
Project Creation Wizard
creating facts 201
removing facts 205
Project Duplication Wizard
accessing 129
duplicate 132
log options 133
source project 131
uses 129
viewing options 132
Project schema
print 197
projects
connection modes 67
create 125
creating 199, 202
creating attributes 200
creating facts 201
creating sources 203
creation script 67
database instance 69
displaying data explorer 204
displaying hidden objects 203
displaying system hierarchy 204
duplicating 68
facts 125
modifying sources 204
removing attributes 205
removing facts 205
removing hierarchies 205
selecting tables 206
sources 66
tables 125
Warehouse Catalog 69

Q
Query Engine Logic
definition 237

R
RDBMS platform

IBM DB2 AS/400 232
IBM DB2 OS/390 232
IBM DB2 UDB 232
Informix 233
Microsoft SQL Server 234
NCR Teradata 236
Oracle 233
Red Brick 233
Sybase Adaptive Server 234
Sybase IQ 12 235
Tandem NonStop SQL 235

S
SQL Engine Logic
definition 227
sql generation
defining columns 172

T
Table Editor
accessing 135
layout 136
logical view 136
physical view 136
uses 135
tables
logical 136
physical 136
setting logical size 207
viewing structure 207
Transformation Editor
accessing 137
layout 138
mapping type 139
member attributes 138
uses 137
transformations 71
creating 71, 210
creating member attributes 210
deleting 211
expressions 138
mapping type 139
mapping types 72
member attribute expression 72
member attributes 71, 72, 138
modifying 212
modifying member attributes 212
removing member attributes 213

V

VMall
    definition 79
    examples 33, 35, 36, 37, 38

W

warehouse
    catalog 143
    changing database instance 215
    editing database instance 216
    editing tables 217
    schema 144
    setting read mode 218
    table prefixes 143
    table size 144
    view 143
Warehouse Catalog Editor
    layout 142
    uses 141
wizards
    attributes 91
    facts 99
    level extensions 109
    projects 123