

My Collection



TechNet

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. This document does not provide you with any legal rights to any intellectual property in any Microsoft product or product name. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes. © 2013 Microsoft. All rights reserved.

Terms of Use (<http://technet.microsoft.com/cc300389.aspx>) | Trademarks (<http://www.microsoft.com/library/toolbar/3.0/trademarks/en-us.mspx>)

Table Of Contents

Chapter 1

[Master Data Services Features and Tasks](#)

Master Data Services Features and Tasks

SQL Server 2012

Learn how to use Master Data Services to manage your organization's master data. Learn how to create and manipulate model objects, which are containers for data. Learn how to organize and maintain the master data itself. Learn how to import and export master data, as well as share models between test and production environments. Finally, learn how to develop custom applications that access the Master Data Services web service.

▲ Create Structures to Contain Data

- [Models \(Master Data Services\)](#)
- [Entities \(Master Data Services\)](#)
- [Attributes \(Master Data Services\)](#)
- [Domain-Based Attributes \(Master Data Services\)](#)
- [Attribute Groups \(Master Data Services\)](#)

▲ Maintain Master Data

- [Master Data Services Add-in for Microsoft Excel](#)
- [Members \(Master Data Services\)](#)
- [Transactions \(Master Data Services\)](#)
- [Annotations \(Master Data Services\)](#)
- [Hierarchies \(Master Data Services\)](#)
- [Collections \(Master Data Services\)](#)

▲ Improve Data Quality

- [Business Rules \(Master Data Services\)](#)
- [Validation \(Master Data Services\)](#)
- [Versions \(Master Data Services\)](#)
- [Notifications \(Master Data Services\)](#)
- [Security \(Master Data Services\)](#)

▲ Move Data

- [Importing Data \(Master Data Services\)](#)
- [Exporting Data \(Master Data Services\)](#)
- [Deploying Models \(Master Data Services\)](#)

▲ Develop a Custom Application

- [Developer's Guide \(Master Data Services\)](#)
- [Microsoft.MasterDataServices](#)


Master Data Services Overview


SQL Server 2012

The following workflow provides an overview of how to use Master Data Services.

You might want to complete the following tasks in a test environment and use the sample data provided when you install Master Data Services. For more information, see [Deploying Models \(Master Data Services\)](#).

Unless otherwise noted, all of the following procedures require you to be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

Action	Details	Related Topics
Create a model	A model is the highest level of organization in Master Data Services. Each model you create should address a specific business need. For example, you might need to track a master list of products or customers. When you create a model, it is considered VERSION_1.	Models (Master Data Services) Create a Model (Master Data Services)
Create entities	Entities are model objects that contain members. Members are the physical master data that you manage in Master Data Services. Create as many entities as you need to contain your members. For a Product model, you might create a Product entity to contain the master list of product members.	Entities (Master Data Services) Create an Entity (Master Data Services)
Create entities to use as domain-based attributes	A domain-based attribute is an attribute with values that are populated by members from an entity. To create a domain-based attribute, first create the entity to populate the attribute value list. For example, you might create a Color entity to contain a master list of colors. These colors can then be used as attribute values for the Product entity.	Domain-Based Attributes (Master Data Services) Create a Domain-Based Attribute (Master Data Services)
Create attributes for your entities	Create attributes to describe members. A Name and Code attribute are automatically included in each entity and cannot be removed. You might want to create other free-form attributes to contain text, dates, numbers, or files. For example, a Product entity might have free-form attributes for Cost, Weight, or SellStartDate.	Attributes (Master Data Services) Create a Text Attribute (Master Data Services) Create a Numeric Attribute (Master Data Services) Create a Date Attribute (Master Data Services) Create a Link Attribute (Master Data Services) Create a File Attribute (Master Data Services)
Create attribute groups	If you have more than four or five attributes for an entity, you might want to create attribute groups. These groups are the tabs that are displayed above the grid in Explorer and they help ease navigation by grouping attributes together on individual tabs.	Attribute Groups (Master Data Services) Create an Attribute Group (Master Data Services)
Import members for your supporting entities	Begin populating your master data by using the staging process. You may want to start by importing the members for your supporting entities. For the Product model, this might mean importing colors or sizes. You can also create members manually. <div> Note Users can create members in Master Data Manager if they have a minimum of Update permission to an entity's leaf model object and access to the Explorer functional area.</div>	Importing Data (Master Data Services) Load or Update Members in Master Data Services by Using the Staging Process Create a Leaf Member (Master Data Services)
Create business rules to ensure data quality	Create and publish business rules to ensure the accuracy of your data. You can use business rules to: <ul style="list-style-type: none">Set default attribute values.Change attribute values.Send email notifications when data doesn't pass business rule validation.	Business Rules (Master Data Services) Create and Publish a Business Rule (Master Data Services) Notifications (Master Data Services) Configure Email Notifications (Master Data Services) Configure Business Rules to Send Notifications (Master Data Services)
Import members for your primary entities and apply business rules	Import the members for your primary entities by using the staging process. When done, validate the version, which applies business rules to all members in the model version. You can then work to correct any business rule validation issues.	Validation (Master Data Services) Validate a Version against Business Rules (Master Data Services) Validation Stored Procedure (Master Data Services)
Create derived hierarchies	Create derived hierarchies from the domain-based attribute relationships that exist between the entities in the model. These level-based hierarchies can be updated as your business needs change and ensure that	Derived Hierarchies (Master Data Services)

	all members are accounted for at the appropriate level.	Create a Derived Hierarchy (Master Data Services)
If needed, create explicit hierarchies	If you want to create hierarchies that are not level-based and that include members from a single entity, you can create explicit hierarchies.	Explicit Hierarchies (Master Data Services) Create an Explicit Hierarchy (Master Data Services)
If needed, create collections	<p>If you want to view different groupings of members for reporting or analysis and do not need a complete hierarchy, create a collection.</p> <div>  Note </div> <p>Users can create collections in Master Data Manager if they have a minimum of Update permission to the collection model object and access to the Explorer functional area.</p>	Collections (Master Data Services) Create a Collection (Master Data Services)
Create user-defined metadata	To describe your model objects, add user-defined metadata to your model. The metadata might include the owner of an object or the source the data comes from.	Metadata (Master Data Services) Add Metadata (Master Data Services)
Lock a version of your model and assign a version flag	Lock a version of your model to prevent changes to the members, except by administrators. When the version's data has validated successfully against business rules, you can commit the version, which prevents changes to members by all users. Create and assign a version flag to the model. Flags help users and subscribing systems identify which version of a model to use.	Versions (Master Data Services) Lock a Version (Master Data Services) Create a Version Flag (Master Data Services)
Create subscription views	For your subscribing systems to consume your master data, create subscription views, which create standard views in the Master Data Services database.	Exporting Data (Master Data Services) Create a Subscription View (Master Data Services)
Configure user and group permissions	You cannot copy user and group permissions from a test to a production environment. However, you can use your test environment to determine the security you want to use eventually in production.	Security (Master Data Services) Add a Group (Master Data Services) Add a User (Master Data Services)

When ready, you can deploy your model, with or without its data, to your production environment. For more information, see [Deploying Models \(Master Data Services\)](#).

Master Data Services Add-in for Microsoft Excel

SQL Server 2012

With the SQL Server Master Data Services Add-in for Excel, master lists of reference data can be distributed to everyone at your organization who uses Excel. Security determines which data users can view and update.

You can load filtered lists of data from MDS into Excel, where you can work with it just as you would any other data. When you are done, you can publish the data back to MDS, where it is centrally stored.

If you are an administrator, use the Add-in for Excel to create entities and attributes and to load them with data. This eliminates the need to use any other tools to load data into your models.

In the Add-in for Excel, you can use Data Quality Services (DQS) to match data before loading it into MDS. This helps to prevent duplicate data in MDS.

Terms

When working with the Add-in, you may encounter the following terms.

- The *MDS repository* is where all master data is stored. It is a SQL Server database that is configured to store MDS data. To work with data from the repository, you load data it into Excel; when you're done working with it, you publish changes back to the repository. Administrators can add new entities and attributes to the repository.
- *MDS-managed data* is data that is stored in the MDS repository and that you load into Excel, where the data is displayed as highlighted rows. You can add data that is not MDS-managed to your worksheet, and it is not affected when you refresh the MDS-managed data.
- A *model* is a container of data. Versions of these containers can be created, and usually the latest version is the most recent. For more information, see [Models \(Master Data Services\)](#).
- An *entity* is a list of data. You might think of an entity as a table in a database. For example, the **Color** entity might contain a list of colors. For more information, see [Entities \(Master Data Services\)](#).
- A *member* is a row of data. Each entity contains members. An example of a member is **Blue**. For more information, see [Members \(Master Data Services\)](#).
- An *attribute* is a column of data. Each member has attributes. For example, the **Code** attribute for the **Blue** member is **B**. For more information about attributes, see [Attributes \(Master Data Services\)](#).

Related Tasks

Task Description	Topic
Create a connection to a Master Data Services repository.	Connect to an MDS Repository (MDS Add-in for Excel)
Load MDS-managed data into Excel.	Load Data from MDS into Excel
Save a shortcut query that you can use open the currently displayed MDS-managed data in the future.	Save a Shortcut Query File (MDS Add-in for Excel)
Share shortcuts with others.	Email a Shortcut Query File (MDS Add-in for Excel)
View all changes that have been made to a member.	View All Annotations or Transactions for a Member (MDS Add-in for Excel)
Before publishing new data, find out whether duplication exists.	Match Similar Data (MDS Add-in for Excel)
Publish data from a worksheet into the MDS repository.	Publish Data from Excel to MDS (MDS Add-in for Excel)
Create a new entity with data in the worksheet. (Administrators only)	Create an Entity (MDS Add-in for Excel)
Create a domain-based attribute, also known as a constrained list. (Administrators only)	Create a Domain-based Attribute (MDS Add-in for Excel)
Set properties for loading and publishing data in the Master Data Services Add-in for Excel. (Administrators only)	Setting Properties for Master Data Services Add-in for Excel

Related Content

- [Connections \(MDS Add-in for Excel\)](#)
- [Loading Data \(MDS Add-in for Excel\)](#)
- [Shortcut Query Files \(MDS Add-in for Excel\)](#)
- [Data Quality Matching in the MDS Add-in for Excel](#)

- [Publishing Data \(MDS Add-in for Excel\)](#)
- [Building a Model \(MDS Add-in for Excel\)](#)
- [Security \(Master Data Services\)](#)

Connections (MDS Add-in for Excel)

SQL Server 2012

To download data in to the Master Data Services Add-in for Excel, you must first create a connection. A connection is how the Master Data Services web service knows which MDS database to connect to.

The connection string is usually the URL of the Master Data Manager web application, for example <http://contoso/mds>.

Each time you start Excel, you must connect to an MDS repository. The only exception is when the active spreadsheet already contains MDS-managed data. In this case, a connection is automatically made each time you refresh or publish data in the sheet.

You can create multiple connections. The most recently-accessed connection is considered the default.

Multiple users can be connected at the same time. However, conflicts can arise when multiple users attempt to publish the same data. For more information, see [Publishing Data \(MDS Add-in for Excel\)](#).

Connect Automatically and Load Frequently-Used Data

If you want to always connect to the same server and load the same set of data, you can create shortcut query files, which contain connection and filter information. For more information about query files, see [Shortcut Query Files \(MDS Add-in for Excel\)](#).

Data Quality Services

The Add-in for Excel has Data Quality Services functionality to help you match data before publishing it to the MDS repository. When you make a connection, if a DQS database is installed on the same instance of SQL Server as the MDS database, you will be able to view DQS buttons on the ribbon. If the DQS_Main database does not exist on the instance, these buttons are not displayed and data quality functionality is not available.

Troubleshooting Connections

When you connect to MDS, if you encounter any issues see <http://social.technet.microsoft.com/wiki/contents/articles/4520.aspx> for troubleshooting tips.

Related Tasks

Task Description	Topic
Create a connection to a Master Data Services database.	Connect to an MDS Repository (MDS Add-in for Excel)
Load MDS data into Excel.	Load Data from MDS into Excel
Filter MDS data before you load it into Excel.	Filter Data before Loading (MDS Add-in for Excel)

Related Content

- [Loading Data \(MDS Add-in for Excel\)](#)
- [Shortcut Query Files \(MDS Add-in for Excel\)](#)
- [Master Data Services Add-in for Microsoft Excel](#)

Connect to an MDS Repository (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, you must connect to an MDS repository before you can load or publish data.

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.

To connect to an MDS repository

- In the MDS Add-in for Excel, on the **Master Data** tab, in the **Connect and Load** group, click the arrow under the **Connect** button and click **Manage Connections**.
- On the **Manage Connections** dialog box, in the **New connection** section, click **Create a new connection**.
- Click **New**.
- On the **Add New Connection** dialog box, in the **Description** field, type a description for your connection. This connection will be displayed when you click the arrow under the **Connect** button on the toolbar.
- In the **MDS server address** box, type the URL of the Master Data Manager web application, for example <http://contoso/mds>.

 Note
Ensure that you use the computer name; do not use "localhost".

- Click **OK**. The name is displayed in the **Existing Connections** section.
- Optionally, click **Test** to test the connection. A confirmation or error dialog is displayed. Click **OK** to close it.
- Click **Connect**. The **Master Data Services** pane is displayed.

Next Steps

- [Load Data from MDS into Excel](#)
- [Filter Data before Loading \(MDS Add-in for Excel\)](#)

See Also

Concepts
[Connections \(MDS Add-in for Excel\)](#)

Loading Data (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, you must load data from the MDS repository into an active Excel worksheet before you can work with it. When you are done working with the data, publish it to the MDS repository so other users can share it.

The data you can load is limited to the data you have permission to access. Permission to access data is set in the Master Data Manager web application or set programmatically.

When you load large amounts of data, you can set warnings that are displayed when the data that might take a long time to load. To do this, in the **Options** group, click **Settings**. On the **Data** tab, select the **Display filter warning for large data sets**.

⚠ Caution

An MDS-enabled workbook must be opened and updated only in Excel with the MDS Add-in for Excel. Opening an MDS-enabled workbook in Excel on a computer in which the MDS Excel Add-in is not installed is not supported, and could cause corruption of the workbook file. If you want to share data with someone else, email a shortcut query file to them, rather than saving the worksheet and emailing it. For more information on the query, see [Email a Shortcut Query File \(MDS Add-in for Excel\)](#).

▀ Filtering Data

You can filter data before loading to limit the amount of data that you're going to download. This includes choosing which attributes (columns) you want to load, the order you want to display the attributes, and the members (rows of data) you want to work with. For more info see [Filter Data before Loading \(MDS Add-in for Excel\)](#).

▀ Connect Automatically and Load Frequently-Used Data

If you want to always connect to the same server and load the same set of data, you can create shortcut query files, which contain connection and filter information. For more information about query files, see [Shortcut Query Files \(MDS Add-in for Excel\)](#).

▀ Refreshing Data

Data in the MDS repository may be updated by other users after you load it. You can retrieve this data without losing changes you've made to non-MDS data. For more information, see [Refreshing Data \(MDS Add-in for Excel\)](#).

▀ Related Tasks

Task Description	Topic
Filter MDS data before you load it into Excel.	Filter Data before Loading (MDS Add-in for Excel)
Load MDS data into Excel.	Load Data from MDS into Excel
Change the order of columns before you download data.	Reorder Columns (MDS Add-in for Excel)

▀ Related Content

- [Connections \(MDS Add-in for Excel\)](#)
- [Shortcut Query Files \(MDS Add-in for Excel\)](#)
- [Master Data Services Add-in for Microsoft Excel](#)
- [Security \(Master Data Services\)](#)

Filter Data before Loading (MDS Add-in for Excel)

SQL Server 2012

In Master Data Services Add-in for Excel, filter data when you want to limit the size or scope of data that you are loading into Excel.


Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.

To filter data before loading

- Open Excel and on the **Master Data** tab, connect to an MDS repository. For more information, see [Connect to an MDS Repository \(MDS Add-in for Excel\)](#).
- In the **Master Data Explorer** pane, select a model and version. The list of entities is populated.
 - If the **Master Data Explorer** pane is not visible, in the **Connect and Load** group, click **Show Explorer**.
 - If the **Master Data Explorer** pane is disabled, it is because the existing sheet already contains MDS-managed data. To enable the pane, open a new worksheet.
- In the **Master Data Explorer** pane, in the list of entities, click the entity you want to filter.
- On the ribbon, in the **Connect and Load** group, click **Filter**.
- Complete the **Filter** dialog box by selecting the attributes (columns) to display, setting the order of the columns, and if needed, filtering the data so fewer rows are returned. View the **Summary** pane for how much data will be returned. For more information, see [Filter Dialog Box \(MDS Add-in for Excel\)](#).
- Click **Load Data**. The sheet is populated with MDS-managed data.

 Note
<ul style="list-style-type: none">Only the first one million members are loaded into Excel.In columns that are constrained lists (domain-based attributes), only the first 1000 values are loaded.

Next Steps

[Publish Data from Excel to MDS \(MDS Add-in for Excel\)](#)

See Also

Tasks

[Reorder Columns \(MDS Add-in for Excel\)](#)

Reference

[Filter Dialog Box \(MDS Add-in for Excel\)](#)

Concepts

[Loading Data \(MDS Add-in for Excel\)](#)

Load Data from MDS into Excel

SQL Server 2012

In the Master Data Services Add-in for Excel, you must load data from the MDS repository in order to work with it.

If you want to filter the dataset before loading, see [Filter Data before Loading \(MDS Add-in for Excel\)](#) instead.


Prerequisites

To perform this procedure:


- You must have permission to access the **Explorer** functional area.

To load data from MDS into Excel

- Open Excel and on the **Master Data** tab, connect to an MDS repository. For more information, see [Connect to an MDS Repository \(MDS Add-in for Excel\)](#).
- In the **Master Data Explorer** pane, select a model and version. The list of entities is populated.
 - If the **Master Data Explorer** pane is not visible, in the **Connect and Load** group, click **Show Explorer**.
 - If the **Master Data Explorer** pane is disabled, it is because the existing sheet already contains MDS-managed data. To enable the pane, open a new worksheet.
- In the **Master Data Explorer** pane, in the list of entities, double-click the entity you want to load.

 **Note**

- Only the first one million members are loaded into Excel. To filter the list before loading, on the ribbon in the **Connect and Load** group, click **Filter**.
- In columns that are constrained lists (domain-based attributes), only the first 25,000 values are loaded. You can change this number in the MaximumDbEntitySize property in the excelusersettings.config file located on the computer that Excel is installed on. This file is located in C:\Users\<user>\AppData\Local\Microsoft\Microsoft SQL Server\110\MasterDataServices\.

 **Note**

When you load text-delimited data using the Add-in for Microsoft Excel with 32-bit Excel, and the settings for the **Cell Count to Load** and **Cell Count to Publish** properties are both set to the maximum of 1000, an out-of-memory error will occur. You have to use 64-bit Excel to use the maximum settings for **Cell Count to Load** and **Cell Count to Publish**.

Next Steps

[Publish Data from Excel to MDS \(MDS Add-in for Excel\)](#)

See Also

Reference

[Filter Dialog Box \(MDS Add-in for Excel\)](#)

Concepts

[Loading Data \(MDS Add-in for Excel\)](#)

[Publishing Data \(MDS Add-in for Excel\)](#)

Filter Dialog Box (MDS Add-in for Excel)


SQL Server 2012

In the Master Data Services Add-in for Excel, use the **Filter** dialog box to narrow the list of MDS-managed data before loading it into Excel.

This dialog box contains three sections: **Columns**, **Rows**, and **Summary**.


Columns

Use the **Columns** section to determine which attributes (columns) you want to display in Excel.

Control Name	Description
Attribute type	An attribute type describes the type of members you want to work with. In most cases, this is Leaf . For more information about member types, see Members (Master Data Services) .
Explicit hierarchy	If you chose the Consolidated attribute type, choose the hierarchy the consolidated members belong to. For more information, see Explicit Hierarchies (Master Data Services) .
Attribute Groups	Attribute groups are a way of grouping subsets of attributes. Choose an attribute group if you want to show a subset of available attributes. For more information about attribute groups, see Attribute Groups (Master Data Services) .
Select All	Click to select all attributes displayed in the list.
Clear All	Click to clear the selected attributes displayed in the list. <div> Note You cannot clear Name and Code.</div>
Up Arrow	Click to move the selected attribute up in the list. The top-to-bottom order corresponds to the left-to-right order the columns are displayed in the worksheet.
Down Arrow	Click to move the selected attribute down in the list. The top-to-bottom order corresponds to the left-to-right order the columns are displayed in the worksheet.

Rows

Use the **Rows** section to determine which members (rows) you want to display in Excel. You do this by defining criteria to filter the rows that will be displayed.

Control Name	Description
Attribute	Displays an attribute you want to filter by. If no attributes are listed, it's because they have not been added. <div> Note You can filter by attributes that you don't plan to show in the worksheet.</div>
Operator	Displays operators that correspond to the type of attribute that was selected. For more information, see Filter Operators (Master Data Services) .
Criteria	The criteria you want to filter by.
Update Summary	When working with large datasets, click to update the Summary section with details of the amount of data that will be loaded.
Add	When you click an attribute in the Columns section and then click Add , an attribute is added to the list of filters.
Remove All	Removes all filters from the list.
Remove	Removes selected filter from the list.

Summary

Use the **Summary** section to view details about the amount of data that will be loaded, before loading it.

Control Name	Description
Model	The name of the model.
Version	The name of the version.

Entity	The name of the entity.
Rows	The number of rows that will be loaded into Excel, based on the filters applied in the Rows section.
Columns	The number of columns that will be loaded into Excel, based on the attributes selected in the Columns section.

▀ See Also

- Tasks
- [Filter Data before Loading \(MDS Add-in for Excel\)](#)
- Concepts
- [Loading Data \(MDS Add-in for Excel\)](#)

Reorder Columns (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, you can reorder columns by filtering the list before loading.

When you reorder attributes in the **Filter** dialog box, the data is loaded into Excel with the new order. However, the next time that you filter the attribute data, the order will revert to the order in the original design. To change the order permanently, an administrator should change the order in the **System Administration** area of Master Data Manager. For more information, see [Change the Order of Attributes](#).

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.

To reorder MDS-managed columns

1. Open Excel and on the **Master Data** tab, connect to an MDS repository. For more information, see [Connect to an MDS Repository \(MDS Add-in for Excel\)](#).
2. In the **Master Data Explorer** pane, select a model and version. The list of entities is populated.
 - If the **Master Data Explorer** pane is not visible, in the **Connect and Load** group, click **Show Explorer**.
 - If the **Master Data Explorer** pane is disabled, it is because the existing sheet already contains MDS-managed data. To enable the pane, open a new worksheet.
3. In the **Master Data Explorer** pane, click an entity.
4. In the **Connect and Load** group, click **Filter**.
5. In the **Filter** dialog box, in the **Columns** section, in the list of attributes, click the attribute you want to move.
6. To the right of the list, click the **Up** or **Down** arrow to move the attribute left and right in the worksheet.
7. Repeat step 7 for each attribute until the top-to-bottom order represents the left-to-right order you want in the worksheet.
8. Click **Load Data**. The sheet is populated with MDS-managed data and the columns are displayed in the order you specified.

See Also

Concepts

[Loading Data \(MDS Add-in for Excel\)](#)

Shortcut Query Files (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, use shortcut query files to quickly connect and load frequently-used data. You can also use them when you want to share MDS data with others. Instead of saving the worksheet and emailing it, you should save a shortcut query file and email that instead. This ensures that you are both connecting to the MDS repository to get the latest data.

Shortcut query files are XML files that contain information about:

- The MDS repository connection.
- The model, version, and entity.
- Any filters that were applied when the shortcut was created.
- The left-to-right order of the columns when the shortcut was created.

You can save these files in a list and choose from them when you open the Add-in. You can export them to your computer or to a shared location, or you can send them to others.

QueryOpener Application

All users who install the Add-in for Excel have an application called QueryOpener installed. This application is used to open shortcut query files in the Add-in for Excel. If you double-click a shortcut query file, this application is automatically used to open the file in the Add-in.

When you open a shortcut query file with this application, you are prompted to make the connection a “safe” connection, which means you trust content from this location. Each time you mark a connection as safe, it is added to a list. If you want to clear this list, open the **Settings** dialog box and in the **Servers Added to Safe List** section, click **Clear All**.

The default location for the application is *drive:\Program Files\Microsoft SQL Server\110\Master Data Services\Excel Add-In\Microsoft.MasterDataServices.QueryOpener.exe*.

There are two ways to open shortcut query files: you can import them or you can double-click them and they are opened automatically.

Related Tasks

Task Description	Topic
Save the contents of the active worksheet as a shortcut query file.	Save a Shortcut Query File (MDS Add-in for Excel)
Email a shortcut query file that represents the contents of the active worksheet.	Email a Shortcut Query File (MDS Add-in for Excel)

Related Content

- [Connections \(MDS Add-in for Excel\)](#)
- [Master Data Services Add-in for Microsoft Excel](#)
- [Security \(Master Data Services\)](#)

Save a Shortcut Query File (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, save a shortcut query file when you frequently need to access the same set of data.

Prerequisites

To perform this procedure:

- The active worksheet must contain MDS-managed data in the format you want. For example, you should be viewing the columns in the order you want them, with any filters applied. For more information, see [Filter Data before Loading \(MDS Add-in for Excel\)](#) and [Reorder Columns \(MDS Add-in for Excel\)](#).

To save a shortcut query file

1. In the **Save and Send** group, click the arrow under **Save Query** and click **Save as Query**.
2. Optional. To rename the query, click the arrow under **Save Query** and click **Manage Queries**. Select the query and click **Rename**.

Next Steps

- Click the arrow under **Save Query** to view the shortcut query and open it.
- [Email a Shortcut Query File \(MDS Add-in for Excel\)](#)

See Also

Concepts

[Shortcut Query Files \(MDS Add-in for Excel\)](#)

Email a Shortcut Query File (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, email a shortcut query file to someone when you want to ensure they're working with the same data that you are. You should share queries rather than saving the worksheet and emailing it.

Prerequisites

To perform this procedure:

- You must have Outlook 2010 or later installed.
- You must have MDS-managed data in an active worksheet in Excel.

To send a shortcut query file

1. Ensure that the data in the worksheet is in the format you want to share. For more information about filtering data and reordering columns, see [Filter Data before Loading \(MDS Add-in for Excel\)](#) and [Reorder Columns \(MDS Add-in for Excel\)](#).
2. In the **Save and Send** group, click **Send Query**. An email message opens and the shortcut query file is attached.

Next Steps

- To open the shortcut query file, the recipient of the email must have the MDS Add-in for Excel installed. The recipient can double-click the file to open it.

See Also

Concepts

[Shortcut Query Files \(MDS Add-in for Excel\)](#)

Refreshing Data (MDS Add-in for Excel)

SQL Server 2012

In Master Data Services Add-in for Excel, refresh data when you want to get the latest information from the MDS repository without opening a new worksheet. You can refresh either all cells or a selection of cells. This can be useful when you have inserted columns with custom formulas or other data that is not managed in MDS and you want to preserve it.

When You Can Refresh MDS-Managed Data

You can refresh MDS-managed data in an active worksheet if the active worksheet already contains MDS-managed data. If you have changed attribute values or added members to the worksheet, you must publish your changes before you can refresh.

Refreshing a Selection

You have the choice of refreshing all cells or refreshing only selected cells. The selected cells must be contiguous. If a set of contiguous cells is selected, all MDS managed cells in that set will be updated to display the values currently stored on the server. Unchanged rows and columns that are not managed by MDS are not affected in any way.

What Happens When You Refresh MDS-Managed Data

When you refresh data in the Add-in for Excel, what happens to the MDS-managed data in the sheet depends on what has changed in the MDS repository since you last loaded the data.

- If new members have been added to repository, they are added to the end of the worksheet and are highlighted in green.
- If members were deleted from the repository, they are deleted from the worksheet.
- If an attribute value has changed in the MDS repository, the value in the worksheet is updated with value from the MDS repository. The cell color does not change.

Caution

- In the active worksheet, if non-managed data exists in rows beneath the MDS-managed data, the non-managed data may be overwritten. This occurs when you refresh the sheet and new rows of MDS-managed data, which overlap with the non-managed data, are added.
- When you refresh, comments on MDS-managed cells are deleted.

How to Refresh MDS-Managed Data

In the **Connect and Load** group on the ribbon, the **Refresh** button has two options, **Refresh All** and **Refresh Selection**. The default action of the ribbon button is **Refresh All**. To refresh the entire sheet with values from the server, click the **Refresh** button or choose the **Refresh All** option. To refresh only some of the cells in a sheet, select the cells (must be one contiguous selection) and choose the **Refresh Selection** option.

Related Tasks

Task Description	Topic
Create a connection to a Master Data Services database.	Connect to an MDS Repository (MDS Add-in for Excel)
Load MDS data into Excel.	Load Data from MDS into Excel

Related Content

- [Connections \(MDS Add-in for Excel\)](#)
- [Loading Data \(MDS Add-in for Excel\)](#)
- [Master Data Services Add-in for Microsoft Excel](#)

Publishing Data (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, publish data to the MDS repository when you want to share it with other users. As soon as data is published, it is available for other users of the Add-in to download.

When you publish data, any data you've added or updated is published to the MDS repository. Data you've deleted is not published—you must delete data separately. For more information, see [Delete a Row \(MDS Add-in for Excel\)](#).

Note

Publishing cannot be used to create a new entity. For more information about creating entities, see [Create an Entity \(MDS Add-in for Excel\)](#).

When Multiple Users Publish at the Same Time

Multiple users can publish updates to the same data. As each user publishes, the update is saved to the database. This means that a user who was not working with the most recently-updated data can still change the value when he or she publishes.

Only the updates you make are published to the database. Any out-of-date data in the worksheet is not published.

Transactions and Annotations

Each published change is saved as a transaction. If you choose, you can add annotations (comments) to a transaction, to explain why you made the change.

- You cannot annotate deletions, although deletions are saved as transactions that can be reversed by an administrator.
- If you change the **Code** value for a member, it is not recorded as a transaction, and all previous transactions for the member are unavailable.
- You can view transactions made to a member by other users. You can also view all transactions you've made to a member, even if you no longer have permission to specific attributes.

You can view all transactions made to a member. For more information, see [View All Annotations or Transactions for a Member \(MDS Add-in for Excel\)](#).

Important

If you enter an annotation of more than 500 characters, the annotation is automatically truncated.

Business Rule and Other Validation

When you publish data, validation is performed to ensure data is accurate before it's added to the MDS repository. If the data does not meet specified criteria, it will not be published to the repository. For more information, see [Validating Data \(MDS Add-in for Excel\)](#).

Related Tasks

Task Description	Topic
Publish data from the active worksheet back to the MDS repository.	Publish Data from Excel to MDS (MDS Add-in for Excel)
Delete a row from the MDS repository and from the worksheet at the same time.	Delete a Row (MDS Add-in for Excel)

Related Content

- [Refreshing Data \(MDS Add-in for Excel\)](#)
- [Master Data Services Add-in for Microsoft Excel](#)

Publish Data from Excel to MDS (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, publish data to the MDS repository when you are finished working in Excel and want to save your changes so other users have access to them.

Note

- When you publish changes, comments on MDS-managed cells are deleted.
- A formula is not supported in an MDS-managed cell. A formula in an MDS-managed cell is handled as a text value.

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.
- The active worksheet must contain MDS-managed data and you must have made changes or additions to the MDS-managed data.
- If you are adding members, you do not have to specify a **Code** value if codes for the entity are being automatically generated. For more information, see [Automatic Code Creation \(Master Data Services\)](#).

To publish data to the MDS repository

1. In the **Publish and Validate** group, click **Publish**.
2. Optional. If the **Publish and Annotate** dialog box is displayed, choose to share the same annotation (comment) for all updates, or to annotate each change individually.
3. Optional. Select the **Do not show this dialog box again** check box. You can always show the dialog box in the future by choosing **Settings** and selecting the **Show Publish and Annotate dialog box when publishing** check box.
4. Click **Publish**.

Note

If you are adding new members (rows) to your worksheet and you cannot successfully publish them to the MDS repository, you may not have **Update** permission to all of the attributes in the worksheet. On the **Review** tab, in the **Changes** group, click **Unprotect Sheet** and try to publish again.

Next Steps

[Apply Business Rules \(MDS Add-in for Excel\)](#)

See Also

Concepts

[Publishing Data \(MDS Add-in for Excel\)](#)

[Validating Data \(MDS Add-in for Excel\)](#)

Delete a Row (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, delete a row of data (a member) from the MDS repository when you no longer want it to be available to users.

Note

Administrators can reactivate deleted members. For more information, see [Reactivate a Member or Collection \(Master Data Services\)](#).

Prerequisites

To perform this procedure:

- You must have permission to the **Explorer** functional area.
- You must have an active worksheet that contains MDS-managed data.

To delete a MDS-managed row of data

1. Select the entire row you want to delete by clicking the row heading.
2. In the **Publish and Validate** group, click **Delete**.
3. On the confirmation dialog box, click **Yes**.

See Also

Concepts

[Publishing Data \(MDS Add-in for Excel\)](#)

View All Annotations or Transactions for a Member (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, view annotations (comments) and transactions for rows of data (members) when you want to view updates to the data over time.

Prerequisites

To perform this procedure:

- You must have an active worksheet that contains MDS-managed data.

To view annotations or transactions

1. Click a cell in the row that contains transactions you want to view.
2. Right-click and from the menu that's displayed, click **View Transactions**.
3. In the **View Transactions** dialog box, the list of transactions is displayed. To view all annotations associated with a transaction, click a row in the grid.

See Also

Concepts

[Publishing Data \(MDS Add-in for Excel\)](#)

Combine Data (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, combine data from two worksheets when you want to compare data before publishing. In this procedure, you combine data from a two worksheets into one. Then you can perform further comparisons and determine which data, if any, to publish to the MDS repository.

Prerequisites

- You must have a worksheet that contains MDS-managed data. For more information, see [Load Data from MDS into Excel](#).
- You must have a worksheet that contains data you want to combine with MDS-managed data. This sheet must have a header row.

To combine non-managed data into an MDS-managed sheet

1. On the sheet that contains MDS-managed data, in the **Publish and Validate** group, click **Combine Data**.
2. In the **Combine Data** dialog box, next to the **Range to combine with MDS data** text box, click the icon. The dialog box contracts.
3. Click the sheet that contains the data you want to combine.
4. Highlight all cells on the sheet that you want to combine, including the header row.
5. In the **Combine Data** dialog box, click the icon. The dialog box expands.
6. For a column listed for the MDS entity, select a column under **Corresponding Column**. All MDS columns do not need corresponding columns.
7. Click **Combine**. A **SOURCE** column is displayed, indicating whether the data is from MDS or an external source.

Next Steps

- To find similarities between the MDS-managed and external data, see [Match Similar Data \(MDS Add-in for Excel\)](#).

See Also

Concepts

[Loading Data \(MDS Add-in for Excel\)](#)

[Data Quality Matching in the MDS Add-in for Excel](#)

Validating Data (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, when you publish data, two types of validation take place:

- Any defined business rules are applied to the data.
- Data is checked against allowed attribute values (for example, number of characters or type of data).

In each case, valid data is published to the MDS repository. Data that is not valid is highlighted, and details of the error can be shown in status columns.

When Validation Occurs

In the Master Data Services Add-in for Excel, validation occurs when you publish new or changed data, or when you manually apply business rules.

When business rules fail, the data is still published to the MDS repository. When input validation fails, the data is not published to the repository.

Validation Statuses

In the Master Data Services Add-in for Excel, the following validation statuses are possible.

Status	Description
Error	One or more values in the row failed validation against business rules defined by an MDS administrator.
Not Validated	Values in the row have not yet been validated against business rules.
Success	All values in the row have passed validation against business rules.

Input Statuses

In the Master Data Services Add-in for Excel, the following input statuses are possible

Status	Description
Error	One or more values in the row don't meet system requirements like length or data type. The value is not updated in the MDS repository.
New Row	The values in the row have not yet been published to the MDS repository.
Read Only	The logged in user has Read-only permissions to one or more values in the row and the value(s) cannot be updated.
Unchanged	No values in the row have been changed in the worksheet. This does not mean the values in the repository have not changed; to get the latest data in the sheet, in the Connect and Load group, click Load or Refresh . This is the default setting for each row.

Related Tasks

Task Description	Topic
Determine which values do not pass the defined business rules.	Apply Business Rules (MDS Add-in for Excel)
To help correct validation errors, view all transactions that have taken place for a member.	View All Annotations or Transactions for a Member (MDS Add-in for Excel)

Related Content

- [Publishing Data \(MDS Add-in for Excel\)](#)

Apply Business Rules (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel apply business rules when you want to validate data and confirm that it is valid. You can correct validations and re-publish the data.

Note

Data validation occurs automatically when you publish data. For more information, see [Validation \(Master Data Services\)](#).

Prerequisites

To perform this procedure:

- You must have access to the **Explorer** functional area.
- You must have an active worksheet that contains MDS-managed data.

To apply business rules

1. In the **Publish and Validate** group, click **Apply Rules**.

Note

The number of members (rows) that are validated at one time depends on a setting in Master Data Services Configuration Manager. For more information, see [Business Rule Settings](#).

2. The data is validated against business rules and two status columns are displayed. If these columns are not displayed automatically, in the **Publish and Validate** group, click **Show Status** to view them.

See Also

Concepts

[Publishing Data \(MDS Add-in for Excel\)](#)

Data Quality Matching in the MDS Add-in for Excel

SQL Server 2012

Over time, you will want to add more data to the MDS repository. Before adding data, it can be useful to compare the new data to the data that's already managed in MDS, to ensure you are not adding duplicate or inaccurate data.

The MDS Add-in for Excel uses the Data Quality Services (DQS) feature of SQL Server to match data that's similar. When you use the matching functionality in the Add-in, similar records are grouped together and a score that represents the accuracy of the result is displayed. For more information about the matching functionality provided by DQS, see [Data Matching](#).

Workflow for Data Quality Matching

When using DQS with the MDS Add-in for Excel, use the following workflow:

1. Retrieve a list of MDS-managed data and combine it with a list that is not managed in MDS. For more information, see [Combine Data \(MDS Add-in for Excel\)](#).
2. Use DQS knowledge to compare the data in the combined list. For more information, see [Match Similar Data \(MDS Add-in for Excel\)](#).
3. To view more details about the similarities found by DQS, show the detail columns.
4. Go through the results and determine which data should be added to the MDS repository and which data is duplicated.
5. Publish the new and/or updated data to the MDS repository.

Knowledge Bases

The matching results provided in the Add-in are based on a DQS knowledge base.

- The default knowledge base (DQS Data) is created when DQS is installed. If you choose to use the default knowledge base (without adding a matching policy to the default knowledge base in the Data Quality Client), you must map columns in the worksheet to domains in the knowledge base, and then assign a weight value to the domains you choose.
- You can use the Data Quality Client to create a new knowledge base with a matching policy, or to add a matching policy to the default knowledge base. In this case, the weight values are determined by the matching policy you already created and you need only to map the columns to the domains. For more information, see [Create a Matching Policy](#).

For more information about knowledge bases, see [DQS Knowledge Bases and Domains](#).

Related Tasks

Task Description	Topic
Combine external data with MDS-managed data in preparation to compare it.	Combine Data (MDS Add-in for Excel)
Use DQS knowledge to find similarities in your data.	Match Similar Data (MDS Add-in for Excel)

Related Content

- [Publishing Data \(MDS Add-in for Excel\)](#)
- [Data Matching](#)

Match Similar Data (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, use Data Quality Services (DQS) functionality to find similarities in your data.

To perform this procedure, you can:

- Use the default Data Quality Services knowledge base, or
- Create your own custom DQS knowledge base and matching policy. For more information, see [Create a Matching Policy](#).

Prerequisites

- You must have a worksheet that contains MDS-managed data. For more information, see [Load Data from MDS into Excel](#).
- Optional. You can combine other data with the MDS-managed data before checking for similarities. For more information, see [Combine Data \(MDS Add-in for Excel\)](#).

To find similarities by using the default knowledge base

1. From the worksheet that contains MDS-managed data, in the **Data Quality** group, click **Match Data**.
2. In the **Match Data** dialog box, from the **DQS Knowledge Base** list, select **DQS Data (default)**.
3. For each column that contains data you want to match, add a row in the dialog box. For information about the fields in this dialog box, see [How to Set Matching Rule Parameters](#).
4. When the total of all weight values equals 100 percent, click **OK**.

To find similarities by using a custom knowledge base

1. From the worksheet that contains MDS-managed data, in the **Data Quality** group, click **Match Data**.
2. From the **DQS Knowledge Base** list, select the name of your custom knowledge base.
3. For each column in the worksheet, select a DQS domain.
4. When all DQS domains are mapped to columns in the worksheet, click **OK**.

Next Steps

- View additional information to determine which data is similar. For more information, see [Data Quality Matching Columns \(MDS Add-in for Excel\)](#).

See Also

Concepts

[Data Quality Matching in the MDS Add-in for Excel](#)
[Data Matching](#)

Data Quality Matching Columns (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, after you match data, in the **Data Quality** group on the ribbon, you can click **Show Details** to display columns that provide matching details.

The following table shows the columns that are displayed when matching data.

Name	Description
CLUSTER_ID	A unique identifier used to group similar records. All rows that are similar have the same CLUSTER_ID . If no CLUSTER_ID is displayed for a row, then no similar records were found.
RECORD_ID	A unique identifier used to identify records. Similar to the Code value stored in the MDS repository, it is a value used to identify a record. It is generated automatically each time matching takes place.
PIVOT_MARK	An arbitrary record that other records are compared to; it does not have a score value.
SCORE	Represents how similar the records in the group are to the pivot record. This score is determined by DQS. If no score is displayed, either the record is the pivot for other records, or no matches were found.

See Also

Tasks

[Match Similar Data \(MDS Add-in for Excel\)](#)

Concepts

[Data Quality Matching in the MDS Add-in for Excel](#)

[Data Matching](#)

Building a Model (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, administrators can perform a subset of the administrative functions available in the Master Data Manager web application.

The model building tasks administrators can do in the Master Data Services Add-in for Excel are:

- Create entities. For more information about entities, see [Entities \(Master Data Services\)](#).
- Create attributes of all types, including domain-based attributes. For more information about attributes, see [Attributes \(Master Data Services\)](#) and [Domain-Based Attributes \(Master Data Services\)](#).

As an administrator, you must create the model by using the Master Data Manager web application or the web service. Then you can use the Add-in for Excel to create entities and attributes within the model. For more information about model objects, see [Models \(Master Data Services\)](#).

Related Tasks

Most administrative tasks must still be done in the Master Data Manager web application or by using the web service. The following table shows which tools administrators can use to complete tasks in MDS.

Task Description	Tool	Topic
Create models.	Master Data Manager web application or web service	Create a Model (Master Data Services)
Create an entity.	Master Data Manager web application, web service, or the Add-in for Excel	Create an Entity (MDS Add-in for Excel)
Create a domain-based attribute.	Master Data Manager web application, web service, or the Add-in for Excel	Create a Domain-based Attribute (MDS Add-in for Excel)
Create attribute groups.	Master Data Manager web application or web service	Create an Attribute Group (Master Data Services)
Create business rules.	Master Data Manager web application or web service	Create and Publish a Business Rule (Master Data Services)
Create subscription views.	Master Data Manager web application or web service	Create a Subscription View (Master Data Services)
Create hierarchies.	Master Data Manager web application or web service	Create a Derived Hierarchy (Master Data Services) Create an Explicit Hierarchy (Master Data Services)
Create collections.	Master Data Manager web application or web service	Create a Collection (Master Data Services)
Create versions of data.	Master Data Manager web application or web service	Lock a Version (Master Data Services)
Deploy models.	Master Data Manager web application, web service, or MDSModelDeploy tool.	Create a Model Deployment Package by Using MDSModelDeploy

Related Content

- [Models \(Master Data Services\)](#)
- [Entities \(Master Data Services\)](#)
- [Attributes \(Master Data Services\)](#)
- [Domain-Based Attributes \(Master Data Services\)](#)
- [Attribute Groups \(Master Data Services\)](#)
- [Business Rules \(Master Data Services\)](#)
- [Exporting Data \(Master Data Services\)](#)
- [Hierarchies \(Master Data Services\)](#)
- [Collections \(Master Data Services\)](#)
- [Versions \(Master Data Services\)](#)
- [Security \(Master Data Services\)](#)
- [Deploying Models \(Master Data Services\)](#)

Create an Entity (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, administrators can create new entities to store data. When you create an entity you should load at least a sampling of the data you want to store.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** and **Explorer** functional areas.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- You must have an existing model to create the entity in. For more information, see [Create a Model \(Master Data Services\)](#).
- Ensure that your data meets the following requirements:
 - The data should have a header row.
 - It is helpful to have **Name** and **Code** columns. **Code** is a unique identifier for each row.
 - You should have at least one row of data other than the header. All columns do not need values, but the data should be representative of the data that will be in the entity.
 - If you have a column that contains a unique identifier (known in MDS as **Code**), ensure that the values are unique. If no column contains identifiers, you can have them generated automatically when you create the entity.
 - Ensure that no cells contain formulas.
 - Ensure that no cells contain time values. Date values can be saved in MDS but time values cannot.

To create an entity and load data

1. Open or create an Excel worksheet that contains data you want to load.
2. Select the cells you want to load into the new entity.
3. On the **Master Data** tab, in the **Build Model** group, click **Create Entity**.
4. If you are prompted to connect to an MDS repository, connect.
5. In the **Create Entity** dialog box, leave the default range or change it to apply to the data you want to load.
6. Do not clear the **My data has headers** check box.
7. From the **Model** list, select a model.
8. From the **Version** list, select a version.
9. In the **New entity name** box, type a name for the entity.
10. From the **Code** list, select the column that contains unique identifiers or have codes generated automatically.
11. Optional. From the **Name** list, select a column that contains names for each member.
12. Click **OK**. When the entity has been created successfully, a new header row is displayed, the cells are highlighted, and the sheet name is updated to match the entity name.

Next Steps

- To view errors that occurred, in the **Publish and Validate** group, click **Show Status**. ValidationStatus and InputStatus columns are displayed. For more information, see [Validating Data \(MDS Add-in for Excel\)](#).
- Confirm that the attributes were created as the data type you expected.

See Also

Tasks

[Create a Domain-based Attribute \(MDS Add-in for Excel\)](#)

Create a Domain-based Attribute (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, administrators can create a domain-based attribute when they want to constrain the values in a column to a specific set of values.

The values can already be in the worksheet or they can come from an existing entity.

Note

If users type a value in the constrained column, rather than selecting from the list, errors are displayed in the **\$InputStatus\$** column when they publish.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** and **Explorer** functional areas.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- The model and entity must already exist.

To perform this procedure:

1. In Excel, load the entity that contains the column (attribute) you want to constrain. For more information, see [Load Data from MDS into Excel](#).
2. Click any cell in the column you want to constrain.
3. In the **Build Model** group, click **Attribute Properties**.
4. In the **Attribute Properties** dialog box, in the **Attribute type** list, choose **Constrained list (domain-based)**.
5. In the **Populate the attribute with values from** list:
 - To use values from the worksheet, choose **the selected column**. A new entity and new staging table will be created with the values from the selected column.
 - To use values from an existing entity, choose the name of the entity.
6. If you chose **the selected column** in the previous step, in the **New entity name** box, type a name for the new entity. This can be the same as the column (attribute) name.
7. Click **OK**. Each cell in the column now has a list of values for users to choose from.

Next Steps

- To add and delete values in the constrained list, load the entity that the attribute is based on. For more information on loading entities, see [Load Data from MDS into Excel](#).

See Also

Tasks

[Create an Entity \(MDS Add-in for Excel\)](#)

Concepts

[Domain-Based Attributes \(Master Data Services\)](#)

[Building a Model \(MDS Add-in for Excel\)](#)

Change the Attribute Type (MDS Add-in for Excel)

SQL Server 2012

In the Master Data Services Add-in for Excel, administrators can change the attribute type when the data type or number of allowed characters is incorrect.

If you want to change the attribute type to create a constrained list (domain-based attribute), see [Create a Domain-based Attribute \(MDS Add-in for Excel\)](#).

Note

You cannot update the type or length of the **Name** or **Code** columns.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** and **Explorer** functional areas.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- There must be an existing model, entity, and attribute.

To change the attribute type

1. In Excel, load the entity that contains the column (attribute) you want to change. For more information, see [Load Data from MDS into Excel](#).
2. Click any cell in the column you want to change.
3. In the **Build Model** group, click **Attribute Properties**.
4. In the **Attribute Properties** dialog box, update settings as needed.
5. Click **OK**.

What happens when you change the attribute type?

If there is any dependency on the attribute, such as the attribute is referenced by any MDS business rule or the attribute is included in a subscription view, and you change the data type of an attribute, MDS will:

- Change the data type of the attribute.
- Generate a copy of the attribute with the suffix “_old” that does not contain any value. This is called a **deprecated** attribute.

However, all the existing dependencies on the original attribute will point to the deprecated attribute, and not to the changed one.

This implies that:

- You must refresh the business rules to point to the changed attribute because the logic may not be the same given the new data type of the attribute. You will have to edit each affected rule, and then rework the expressions to point to remove references from the deprecated attribute (_old) to point to the updated attribute.
- You must open any subscription views under the Integration Management selection, select the view row, open it for editing by clicking the pencil icon, and then click the **Save disk** icon to refresh the view definition. No other change is needed to regenerate the view syntax.
- Staging tables that include the attribute will have a deprecated attribute column added to them, which means that your staging code will be affected. To get rid of the deprecated attribute, you can delete it after you have updated the business rules and subscription views.

Deleting the deprecated attribute

Before you delete any deprecated attribute, you must remove any references to the attribute such as fixing the business rules and regenerating subscription views as described earlier. Otherwise, you will get an error in the System Administration web page when you attempt to delete the deprecated attribute stating that the attribute cannot be deleted because it is referenced by an object.

To delete an attribute, see [Delete an Attribute \(Master Data Services\)](#)

Tip

It is cumbersome to change data types for MDS attributes that have existing data and related entities, especially if there is a business rule or subscription view declared which depends on the entity. The best practice is to start with a data type that is flexible enough to hold the necessary values. For example, strings may start small, but may need to be lengthened over time, so consider the worst case scenarios. Extra text string length can be burdensome (for example, wide GUI text boxes are hard to fit on the screen), so avoid too long string length.

See Also

Concepts

[Attributes \(Master Data Services\)](#)

[Building a Model \(MDS Add-in for Excel\)](#)

Setting Properties for Master Data Services Add-in for Excel

SQL Server 2012

Master Data Services Add-in for Excel settings determine how data is loaded from MDS into the Excel Add-in and how data is published from the Excel Add-in into MDS.

To make settings for the Excel Add-in, open **Excel**, click the **Master Data** menu, and then click **Settings**. Anyone with access to Excel can change these settings. The settings apply to the computer that Excel is open on.

Excel Add-in Settings

Tab and Section	Setting	Description
Settings: Publishing	Show Publish and Annotate dialog box when publishing	Select to display the Publish and Annotate dialog box after you click Publish , enabling you to enter a single annotation for all changes or to enter an annotation for each change. Deselect to specify that the Publish process is initiated without the Publish and Annotate dialog box being displayed. You will not have the opportunity to enter an annotation.
Settings: Version	Version selection	Select the version of the master data that will be loaded into the Excel Add-in. Can be: None to have the version not default to any version Oldest to default to the oldest version Newest to default to the most recent version.
Settings: Logging	Turn on detailed logging	Enable logging for the process of loading master data from MDS into the Excel Add-in, such that the the result of every command in the service is logged.
Settings: Batching size	Number of cells for load	Select a number to indicate how many thousands of cells will be loaded in a batch that is loaded from the MDS server to Excel. The default is 50,000 cells.
Settings: Batching size	Number of cells for publish	Select a number to indicate how many thousands of cells will be published in a batch that is returned from Excel to the server. The default is 50,000 cells.
Settings: Servers Added to Safe List	Clear All	Click to clear the list of connections that were designated as safe when the associated shortcut query file was opened.
Data: Filters	Display filter warning for large data sets	Click to display a warning if the data set being loaded from MDS to Excel exceeds the maximum number of rows or columns.
Data: Filters	Maximum rows	Select the threshold for the number of rows being loaded, beyond which a filter warning will be posted.
Data: Filters	Maximum columns	Select the threshold for the number of columns being loaded, beyond which a filter warning will be posted.
Data: Cell Format	Change the color when: Attribute values change	Click to specify that the color of a cell will be changed if the attribute value in that cell changes when you refresh the Excel Add-in table with new data from the MDS repository.
Data: Cell Format	Change the color when: Members are added	Click to specify that the color of a row's cells will be changed if a new member is added to the row when you refresh the Excel Add-in table with new data from the MDS repository.

Models (Master Data Services)

SQL Server 2012

Models are the highest level of data organization in Master Data Services. A model defines the structure of data in your master data management solution. A model contains the following objects:

- Entities
- Attributes and attribute groups
- Explicit and derived hierarchies
- Collections

Models organize the structure of your master data. Your Master Data Services implementation can have one or many models that each group similar kinds of data. In general, master data can be categorized in one of four ways: people, places, things, or concepts. For example, you can create a Product model to contain product-related data or a Customer model to contain customer-related data.

You can assign users and groups permission to view and update objects within the model. If you do not give permission to the model, it is not displayed.

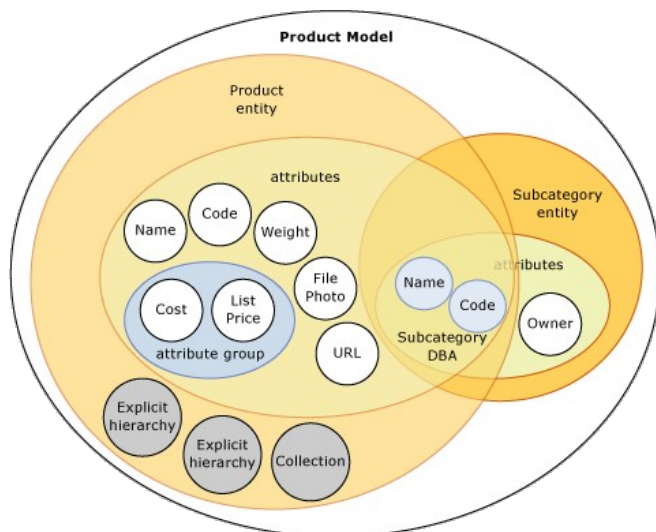
At any given time, you can create copies of the master data within a model. These copies are called versions.

When you have defined a model in a test environment, you can deploy it, with or without the corresponding data, from the test environment to a production environment. This eliminates the need to recreate your models in your production environment.

How Models Relate to Other Objects

A model contains entities. Entities contain attributes, explicit hierarchies, and collections. Attributes can be contained in attribute groups. Domain-based attributes exist when an entity is used as an attribute for another entity.

This image shows the relationships among the objects in a model.



Note

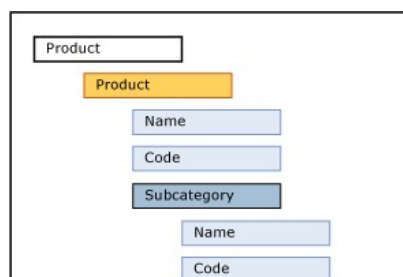
Derived hierarchies are also model objects, but they are not shown in the image. Derived hierarchies are derived from the domain-based attribute relationships that exist between entities. See [Derived Hierarchies \(Master Data Services\)](#) for more information.

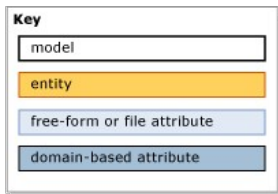
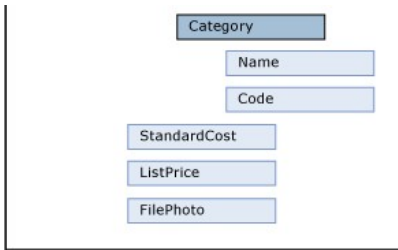
Master data is the data that is contained in the model objects. In Master Data Services, master data is stored as members in an entity.

Model objects are maintained in the **System Administration** functional area of the Master Data Manager user interface.

Model Example

In the following example, the objects in the Product model logically group product-related data.





Other common models are:

- Accounts, which could include entities such as balance sheet accounts, income statement accounts, statistics, and account type.
- Customer, which could include entities such as gender, education, occupation, and marital status.
- Geography, which could include entities such as postal codes, cities, counties, states, provinces, regions, territories, countries, and continents.

Related Tasks

Task Description	Topic
Create a model to organize your master data.	Create a Model (Master Data Services)
Change the name of an existing model.	Change a Model Name (Master Data Services)
Delete an existing model.	Delete a Model (Master Data Services)

Related Content

- [Master Data Services Overview](#)
- [Entities \(Master Data Services\)](#)
- [Attributes \(Master Data Services\)](#)
- [Deploying Models \(Master Data Services\)](#)
- [Model Object Permissions \(Master Data Services\)](#)

Create a Model (Master Data Services)

SQL Server 2012

In Master Data Services, create a model to contain model objects.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.

To create a model

- In Master Data Manager, click **System Administration**.
- On the **Model View** page, from the menu bar, point to **Manage** and click **Models**.
- On the **Model Maintenance** page, click **Add model**.
- In the **Model name** box, type the name of the model.
- Optionally, select **Create entity with same name as model** to create an entity with the same name as the model.
- Optionally, select **Create explicit hierarchy with same name as model** to create an explicit hierarchy with the same name as the model. This option also enables the entity for collections.
- Optionally, select **Mandatory hierarchy (all leaf members are included)** to create the explicit hierarchy as a mandatory hierarchy.
- Click **Save model**.

Next Steps

- [Create an Entity \(Master Data Services\)](#)

See Also

Tasks

[Delete a Model \(Master Data Services\)](#)

[Change a Model Name \(Master Data Services\)](#)

Concepts

[Models \(Master Data Services\)](#)

[Entities \(Master Data Services\)](#)

Change a Model Name (Master Data Services)

SQL Server 2012

In Master Data Services, you can change the name of a model.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To change a model name

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Models**.
3. On the **Model Maintenance** page, in the grid, click the row for the model with the name you want to change.

 Note
You cannot change the name of the Metadata model.

4. Click **Edit selected model**.
5. In the **Model name** box, type the updated name of the model.
6. Click **Save model**.

See Also

Tasks

[Create a Model \(Master Data Services\)](#)

[Delete a Model \(Master Data Services\)](#)

Concepts

[Models \(Master Data Services\)](#)

Delete a Model (Master Data Services)

SQL Server 2012

Delete a model to remove the model and all of its data from Master Data Services.

Note

When you complete this procedure, all objects and all data from all versions of the model will be permanently deleted.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete a model

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Models**.
3. On the **Model Maintenance** page, select the row for the model that you want to delete.
4. Click **Delete selected model**.
5. In the confirmation dialog box, click **OK**.
6. In the additional confirmation dialog box, click **OK**.

See Also

Tasks

[Create a Model \(Master Data Services\)](#)

Concepts

[Models \(Master Data Services\)](#)

Entities (Master Data Services)

SQL Server 2012

Entities are objects that are contained in Master Data Services models. Each entity contains members, which are the rows of master data that you manage.

How Many Entities are Appropriate?

Models can contain as many entities as you want to manage. Each entity should group a similar kind of data. For example, you might want an entity for all of your corporate accounts, or an entity for your master list of employees.

Typically, there are one or more central entities that are important to your business, and to which other objects in the model relate. For example, in a Product model, you could have a central entity called Product and entities called Subcategory and Category that relate to the Product entity. However, you do not need to have a central entity. Depending on your needs, you might have several entities that you consider to be of equal importance.

How Entities Relate to Other Model Objects

You can think of an entity as a table that contains master data, where the rows represent members and the columns represent attributes.

		Attributes (columns)			
Members (rows)		attribute value	attribute value	attribute value	attribute value
		attribute value	attribute value	attribute value	attribute value
		attribute value	attribute value	attribute value	attribute value
		attribute value	attribute value	attribute value	attribute value
		attribute value	attribute value	attribute value	attribute value

You populate the entity with a list of master data that you want to manage.

Entities can be used to build derived hierarchies, which are level-based hierarchies based on multiple entities. For more information, see [Derived Hierarchies \(Master Data Services\)](#).

Entities can also be enabled to contain explicit hierarchies (ragged structures based on a single entity) and collections (one-off combinations of subsets of members). For more information, see [Explicit Hierarchies \(Master Data Services\)](#) and [Collections \(Master Data Services\)](#).

Using Entities as Constrained Lists

When users are assigning attributes to the members in an entity, you can have them choose from a constrained list of values. To do this, you use an entity to populate the list of values for the attribute. This is called a domain-based attribute. For more information, see [Domain-Based Attributes \(Master Data Services\)](#).

Base Entities

A base entity is a starting point for users when navigating objects in the model. A base entity determines the layout of the screen when a user opens the **Explorer** functional area and clicks **Explorer** on the menu bar. To specify an entity as a base entity, navigate to the **System Administration** functional area. On the **Model View** page, drag the entity from the tree control on the right to the name of the model in the tree control on the left.

Entity Security

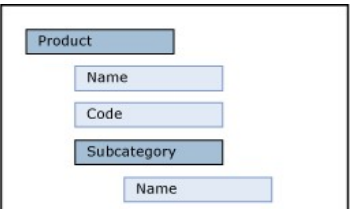
You can give users permission to an entity, which includes related model objects. For more information, see [Entity Permissions \(Master Data Services\)](#).

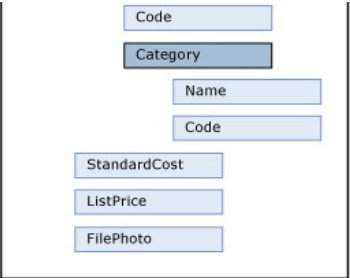
Entity Examples

The following example shows an entity that has these attributes: Name, Code, Subcategory, StandardCost, ListPrice, and FilePhoto. These attributes describe the members. Each member is represented by a single row of attribute values.

Name	Code	Subcategory	StandardCost	ListPrice	FilePhoto
Mountain-100	BK-M101	Mountain Bikes {5}	1912.15	3399.99	
Mountain-200	BK-M201	Mountain Bikes {5}	1265.62	2319.99	
Road-150	BK-R150	Road Bikes {6}	2171.29	3578.27	
Road-450	BK-450	Road Bikes {6}	884.71	1457.99	

In the following example, the Product entity is the central entity. The Subcategory entity is a domain-based attribute of the Product entity. The Category entity is a domain-based attribute of the Subcategory entity. StandardCost and ListPrice are free-form attributes of the Product entity, and FilePhoto is a file attribute of the Product entity.





Note

This is an example based on the Master Data Manager user interface (UI). The hierarchical tree structure shows relationships between entities and domain-based attributes. It is intended to show relationships rather than represent levels of importance.

Related Tasks

Task Description	Topic
Create a new entity.	Create an Entity (Master Data Services)
Specify that an entity can contain explicit hierarchies and collections.	Enable an Entity for Explicit Hierarchies and Collections (Master Data Services)
Change the name of an existing entity.	Change an Entity Name (Master Data Services)
Delete an existing entity.	Delete an Entity (Master Data Services)
Assign permission to entities.	Assign Model Object Permissions (Master Data Services)

Related Content

- [Models \(Master Data Services\)](#)
- [Members \(Master Data Services\)](#)
- [Attributes \(Master Data Services\)](#)

Create an Entity (Master Data Services)

SQL Server 2012

In Master Data Services, create an entity to contain members and their attributes.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- A model must exist. For more information, see [Create a Model \(Master Data Services\)](#).

To create an entity

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Click **Add entity**.
5. In the **Entity name** box, type the name of the entity.
6. In the **Name for staging tables** box, type a name for the staging table.

Tip

Use the model name as part of the staging table name, for example *Modelname_Entityname*. This makes the tables easier to find in the database. For more information about the staging tables, see [Importing Data \(Master Data Services\)](#).

7. Optional. Select the **Create Code values automatically** check box. For more information, see [Automatic Code Creation \(Master Data Services\)](#).
8. From the **Enable explicit hierarchies and collections** list, select one of the following options:
 - **No**. Select this option if you do not need to enable the entity for explicit hierarchies and collections. You can change this later if needed.
 - **Yes**. Select this option when you want to enable the entity for explicit hierarchies and collections. In the **Explicit hierarchy name** box, type a name. Optionally, select **Mandatory hierarchy (all leaf members are included)** to make the explicit hierarchy a mandatory hierarchy.
9. Click **Save entity**.

Next Steps

- [Create a Text Attribute \(Master Data Services\)](#)
- [Create a Domain-Based Attribute \(Master Data Services\)](#)
- [Create a File Attribute \(Master Data Services\)](#)

See Also

Tasks

[Change an Entity Name \(Master Data Services\)](#)

[Delete an Entity \(Master Data Services\)](#)

Concepts

[Entities \(Master Data Services\)](#)

[Explicit Hierarchies \(Master Data Services\)](#)

Enable an Entity for Explicit Hierarchies and Collections (Master Data Services)

SQL Server 2012

In Master Data Services, enable an entity for explicit hierarchies and collections so that you can create explicit hierarchies and collections for the entity.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- An entity must exist. For more information, see [Create an Entity \(Master Data Services\)](#).

To enable an entity for explicit hierarchies and collections

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to update.
5. Click **Edit selected entity**.
6. From the **Enable explicit hierarchies and collections** list, select **Yes**.
7. In the **Explicit hierarchy name** box, type a name for an explicit hierarchy.
8. Optionally, clear the **Mandatory hierarchy** check box to create the hierarchy as a non-mandatory hierarchy.
9. Click **Save entity**.

Next Steps

- [Create an Explicit Hierarchy \(Master Data Services\)](#)
- [Create a Collection \(Master Data Services\)](#)

See Also

Concepts

[Entities \(Master Data Services\)](#)

[Explicit Hierarchies \(Master Data Services\)](#)

[Collections \(Master Data Services\)](#)

Change an Entity Name (Master Data Services)

SQL Server 2012

In Master Data Services, you can change the name of an entity.

Note

The names of the associated staging tables will not be updated.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To change an entity name

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Click the row for the entity with the name you want to change.
5. Click **Edit selected entity**.
6. In the **Entity name** box, type the updated name of the entity.
7. Click **Save entity**.

See Also

Tasks

[Create an Entity \(Master Data Services\)](#)

[Delete an Entity \(Master Data Services\)](#)

Concepts

[Entities \(Master Data Services\)](#)

Delete an Entity (Master Data Services)

SQL Server 2012

In Master Data Services, delete an entity to delete all members and attributes for the entity.

Note

The entity's members from all versions will be permanently deleted.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete an entity

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to delete.
5. Click **Delete selected entity**.
6. In the confirmation dialog box, click **OK**.
7. In the additional confirmation dialog box, click **OK**.

See Also

Tasks

[Create an Entity \(Master Data Services\)](#)

Concepts

[Entities \(Master Data Services\)](#)

Attributes (Master Data Services)

SQL Server 2012

Attributes are objects that are contained in Master Data Services entities. Attribute values describe the members of the entity. An attribute can be used to describe a leaf member, a consolidated member, or a collection.

How Attributes Relate to Other Model Objects

You can think of an attribute as a column in an entity table. An attribute value is the value used to describe a specific member.

		Attributes (columns)			
Members (rows)		attribute value	attribute value	attribute value	attribute value
		attribute value	attribute value	attribute value	attribute value
		attribute value	attribute value	attribute value	attribute value
		attribute value	attribute value	attribute value	attribute value

When you create an entity that contains many attributes, you can organize the attributes into attribute groups. For more information, see [Attribute Groups \(Master Data Services\)](#).

Required Attributes

When you create an entity, the Name and Code attributes are automatically created. Code requires a value and must be unique within the entity. You cannot remove the Name and Code attributes.

Attribute Types

There are three types of attributes:

- Free-form attributes, which allow free-form input for text, numbers, dates, or links.
- Domain-based attributes, which are populated by entities. For more information, see [Domain-Based Attributes \(Master Data Services\)](#).
- File attributes, which are used to store files, documents, or images. File attributes are intended to help with the consistency of your data by requiring files to have a specific extension. File attributes cannot be guaranteed to prevent a malicious user from uploading a file of a different type.

Numeric Free-Form Attributes




Numeric free-form attributes require special handling, because numeric free-form attribute values are limited to the **SqlDouble** value type.

By default, a **SqlDouble** value contains 15 decimal digits of precision, although a maximum of 17 digits is maintained internally. The precision of a floating-point number has several consequences:

- Two floating-point numbers that appear equal for a particular precision might not compare equal because their least significant digits are different.
- A mathematical or comparison operation that uses a floating-point number might not yield the same result if a decimal number is used because the floating-point number might not exactly approximate the decimal number.
- A value might not *roundtrip* if a floating-point number is involved. A value is said to roundtrip if an operation converts an original floating-point number to another form, an inverse operation transforms the converted form back to a floating-point number, and the final floating-point number is equal to the original floating-point number. The roundtrip might fail because one or more least significant digits are lost or changed in a conversion.

Attribute Examples

In the following example, the entity has the attributes: Name, Code, Subcategory, StandardCost, ListPrice, and FilePhoto. These attributes describe the members. Each member is represented by a single row of attribute values.

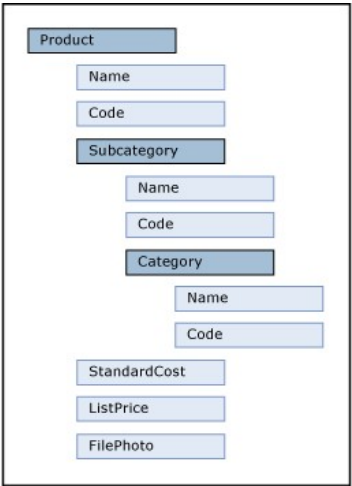
Name	Code	Subcategory	StandardCost	ListPrice	FilePhoto
Mountain-100	BK-M101	Mountain Bikes {5}	1912.15	3399.99	
Mountain-200	BK-M201	Mountain Bikes {5}	1265.62	2319.99	
Road-150	BK-R150	Road Bikes {6}	2171.29	3578.27	
Road-450	BK-450	Road Bikes {6}	884.71	1457.99	

In the following example, the Product entity contains:

- The free-form attributes of Name, Code, StandardCost and ListPrice.
- The domain-based attribute of Subcategory.
- The file attribute of FilePhoto.

Subcategory is an entity that is used as a domain-based attribute of Product. Category is an entity that is used as a domain-based attribute of Subcategory. Like the

Product entity, the Category and Subcategory entities each contain the default Name and Code attributes.



Related Tasks

Task Description	Topic
Create a new free-form text attribute.	Create a Text Attribute (Master Data Services)
Create a new free-form numeric attribute.	Create a Numeric Attribute (Master Data Services)
Create a new free-form link attribute.	Create a Link Attribute (Master Data Services)
Create a new file attribute.	Create a File Attribute (Master Data Services)
Create a new domain-based attribute.	Create a Domain-Based Attribute (Master Data Services)
Change the name of an existing attribute.	Change an Attribute Name (Master Data Services)
Add existing attributes to a change tracking group.	Add Attributes to a Change Tracking Group (Master Data Services)
Delete an existing attribute.	Delete an Attribute (Master Data Services)
Change the order of attributes.	Change the Order of Attributes

Related Content

- [Domain-Based Attributes \(Master Data Services\)](#)
- [Attribute Groups \(Master Data Services\)](#)
- [Members \(Master Data Services\)](#)
- [Leaf Permissions \(Master Data Services\)](#)
- [Consolidated Permissions \(Master Data Services\)](#)

Create a Text Attribute (Master Data Services)

SQL Server 2012

In Master Data Services, create a text attribute when you want users to enter a text string as an attribute value.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- An entity must exist to create the attribute for. For more information, see [Create an Entity \(Master Data Services\)](#).

To create a text attribute

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to create an attribute for.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page:
 - If the attribute is for leaf members, in the **Leaf member attributes** pane, click **Add leaf attribute**.
 - If the attribute is for consolidated members, in the **Consolidated member attributes** pane, click **Add consolidated attribute**.
 - If the attribute is for collections, in the **Collection attributes** pane, click **Add collection attribute**.
7. On the **Add Attribute** page, select the **Free-form** option.
8. In the **Name** box, type a name for the attribute. For a list of words that should not be used as attribute names, see [Reserved Words \(Master Data Services\)](#).
9. In the **Display pixel width** box, type the width of the attribute column to be displayed in the **Explorer** grid.
10. From the **Data type** list, select **Text**.
11. In the **Length** box, type the maximum number of characters allowed.
12. Optionally, select **Enable change tracking** to track changes to groups of attributes. For more information, see [Add Attributes to a Change Tracking Group \(Master Data Services\)](#).
13. Click **Save attribute**.
14. On the **Entity Maintenance** page, click **Save entity**.

See Also

Tasks

[Change an Attribute Name \(Master Data Services\)](#)

[Create a Domain-Based Attribute \(Master Data Services\)](#)

[Create a File Attribute \(Master Data Services\)](#)

Concepts

[Attributes \(Master Data Services\)](#)

Create a Numeric Attribute (Master Data Services)

SQL Server 2012

In Master Data Services, create a numeric attribute when you want users to enter a number as an attribute value.

Note

Numeric attributes have limitations. For more information, see [Attributes \(Master Data Services\)](#).

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- An entity must exist to create the attribute for. For more information, see [Create an Entity \(Master Data Services\)](#).

To create a numeric attribute

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to create an attribute for.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page:
 - If the attribute is for leaf members, in the **Leaf member attributes** pane, click **Add leaf attribute**.
 - If the attribute is for consolidated members, in the **Consolidated member attributes** pane, click **Add consolidated attribute**.
 - If the attribute is for collections, in the **Collection attributes** pane, click **Add collection attribute**.
7. On the **Add Attribute** page, select the **Free-form** option.
8. In the **Name** box, type a name for the attribute. For a list of words that should not be used as attribute names, see [Reserved Words \(Master Data Services\)](#).
9. In the **Display pixel width** box, type the width of the attribute column to be displayed in the **Explorer** grid.
10. From the **Data type** list, select **Number**.
11. In the **Decimals** box, type the number of numbers that can be entered after a decimal.
12. From the **Input mask** list, select a format for negative numbers.
13. Optionally, select **Enable change tracking** to track changes to groups of attributes. See [Add Attributes to a Change Tracking Group \(Master Data Services\)](#) for more information.
14. Click **Save attribute**.
15. On the **Entity Maintenance** page, click **Save entity**.

See Also

Tasks

[Change an Attribute Name \(Master Data Services\)](#)
[Create a Domain-Based Attribute \(Master Data Services\)](#)
[Create a File Attribute \(Master Data Services\)](#)

Concepts

[Attributes \(Master Data Services\)](#)

Create a Date Attribute (Master Data Services)

SQL Server 2012

In Master Data Services, create a date attribute when you want users to enter a date as an attribute value.

Note

The attribute is called DateTime, but time values are not supported.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- You must have an entity to create the attribute for. For more information, see [Create an Entity \(Master Data Services\)](#).

To create a date attribute

- In Master Data Manager, click **System Administration**.
- On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
- On the **Entity Maintenance** page, from the **Model** list, select a model.
- Select the row for the entity that you want to create an attribute for.
- Click **Edit selected entity**.
- On the **Edit Entity** page:
 - If the attribute is for leaf members, in the **Leaf member attributes** pane, click **Add leaf attribute**.
 - If the attribute is for consolidated members, in the **Consolidated member attributes** pane, click **Add consolidated attribute**.
 - If the attribute is for collections, in the **Collection attributes** pane, click **Add collection attribute**.
- On the **Add Attribute** page, select the **Free-form** option.
- In the **Name** box, type a name for the attribute. For a list of words that should not be used as attribute names, see [Reserved Words \(Master Data Services\)](#).
- In the **Display pixel width** box, type the width of the attribute column to be displayed in the **Explorer** grid.
- From the **Data type** list, select **DateTime**.
- From the **Input mask** list, select a format for dates.
- Optionally, select **Enable change tracking** to track changes to groups of attributes. For more information, see [Add Attributes to a Change Tracking Group \(Master Data Services\)](#).
- Click **Save attribute**.
- On the **Entity Maintenance** page, click **Save entity**.

To display the time portion of a datetime value

To have the user interface display the time portion of a datetime value, you must select an appropriate input mask for the attribute. None of the built-in masks for Datetime attributes do this, but you can add a new mask that will allow you to display time. To do so, add a row in the mdm.tblList table of the MDS database, where the built-in masks are stored. The row should have the following values:

ListCode	lstInputMask
ListName	Input Mask
Seq	19
List Option	dd/MM/yyyy hh:mm:ss
Option ID	19
IsVisible	1
Group_ID	3

After you enter a row with the above values in the mdm.tblList table, the "dd/MM/yyyy hh:mm:ss" mask will be available in the Input mask list box. You can then select

that mask to display the date and time in a datetime attribute column of an entity in the MDS Explorer.

▲ See Also

Tasks

[Change an Attribute Name \(Master Data Services\)](#)

[Create a Domain-Based Attribute \(Master Data Services\)](#)

[Create a File Attribute \(Master Data Services\)](#)

Concepts

[Attributes \(Master Data Services\)](#)

Create a Link Attribute (Master Data Services)

SQL Server 2012

In Master Data Services, create a link attribute when you want users to enter a hyperlink as an attribute value, such as <http://www.contoso.com>.

Note

When users enter a value for a link attribute, the string must begin with **http://** or an error will be displayed.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- An entity must exist to create the attribute for. For more information, see [Create an Entity \(Master Data Services\)](#).

To create a link attribute

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to create an attribute for.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page:
 - If the attribute is for leaf members, in the **Leaf member attributes** pane, click **Add leaf attribute**.
 - If the attribute is for consolidated members, in the **Consolidated member attributes** pane, click **Add consolidated attribute**.
 - If the attribute is for collections, in the **Collection attributes** pane, click **Add collection attribute**.
7. On the **Add Attribute** page, select the **Free-form** option.
8. In the **Name** box, type a name for the attribute. For a list of words that should not be used as attribute names, see [Reserved Words \(Master Data Services\)](#).
9. In the **Display pixel width** box, type the width of the attribute column to be displayed in the **Explorer** grid.
10. From the **Data type** list, select **Link**.
11. In the **Length** box, type the maximum number of characters allowed.
12. Optionally, select **Enable change tracking** to track changes to groups of attributes. For more information, see [Add Attributes to a Change Tracking Group \(Master Data Services\)](#).
13. Click **Save attribute**.
14. On the **Entity Maintenance** page, click **Save entity**.

See Also

Tasks

[Change an Attribute Name \(Master Data Services\)](#)

[Create a Domain-Based Attribute \(Master Data Services\)](#)

[Create a File Attribute \(Master Data Services\)](#)

Concepts

[Attributes \(Master Data Services\)](#)

Create a File Attribute (Master Data Services)

SQL Server 2012

In Master Data Services, create a file attribute to populate attribute values with files.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- An entity must exist to create the attribute for. For more information, see [Create an Entity \(Master Data Services\)](#).

To create a file attribute

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to create an attribute for.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page:
 - If the attribute is for leaf members, in the **Leaf member attributes** pane, click **Add leaf attribute**.
 - If the attribute is for consolidated members, in the **Consolidated member attributes** pane, click **Add consolidated attribute**.
 - If the attribute is for collections, in the **Collection attributes** pane, click **Add collection attribute**.
7. On the **Add Attribute** page, select the **File** option.
8. In the **Name** box, type a name for the attribute. For a list of words that should not be used as attribute names, see [Reserved Words \(Master Data Services\)](#).
9. In the **Display pixel width** box, type the width of the attribute column to be displayed in the **Explorer** grid.
10. From the **File extension** list, select one or more file types that a user can upload, or leave the default (*.*) to allow all file types.
11. Optionally, select **Enable change tracking** to track changes to groups of attributes. For more information, see [Add Attributes to a Change Tracking Group \(Master Data Services\)](#).
12. Click **Save attribute**.
13. On the **Entity Maintenance** page, click **Save entity**.

See Also

Tasks

[Change an Attribute Name \(Master Data Services\)](#)

[Create a Domain-Based Attribute \(Master Data Services\)](#)

[Create a Text Attribute \(Master Data Services\)](#)

Concepts

[Attributes \(Master Data Services\)](#)

Change an Attribute Name (Master Data Services)

SQL Server 2012

In Master Data Services, you can change the name of an attribute.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To change an attribute name

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Click the row for the entity that contains the attribute with the name you want to change.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page, click the attribute with the name you want to change.
7. Click **Edit selected attribute**.
8. In the **Name** box, type the updated name of the attribute. For a list of words that should not be used as attribute names, see [Reserved Words \(Master Data Services\)](#).
9. Click **Save attribute**.

See Also

Tasks

[Create a Text Attribute \(Master Data Services\)](#)

[Delete an Attribute \(Master Data Services\)](#)

Concepts

[Attributes \(Master Data Services\)](#)

Delete an Attribute (Master Data Services)

SQL Server 2012

In Master Data Services, delete an attribute when you want to permanently delete the attribute and all associated attribute values.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete an attribute

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that contains the attribute you want to delete.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page, click the attribute you want to delete.

 Note
You cannot delete the Name or Code attributes.

7. Click **Delete selected attribute**.
8. In the confirmation dialog box, click **OK**.
9. In the additional confirmation dialog box, click **OK**.

See Also

Tasks

[Create a Text Attribute \(Master Data Services\)](#)

[Create a Domain-Based Attribute \(Master Data Services\)](#)

Concepts

[Attributes \(Master Data Services\)](#)

[Domain-Based Attributes \(Master Data Services\)](#)

Change the Order of Attributes

SQL Server 2012

In Master Data Services, you can change the order of attributes.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To change the order of an attribute

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Click the row for the entity that contains the attributes that you want to change the order of.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page, in **Leaf member attributes**, or **Consolidated member attributes**, or **Collection attributes**, select the attribute that you want to change the order of, and then click the up arrow or the down arrow to change its order.
7. Click **Save entity**.

See Also

Concepts

[Attributes \(Master Data Services\)](#)

Domain-Based Attributes (Master Data Services)

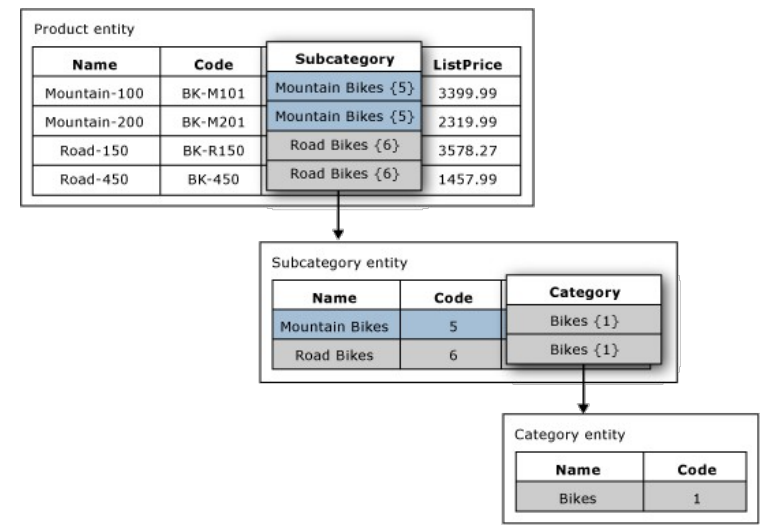
SQL Server 2012

In Master Data Services, a domain-based attribute is an attribute with values that are populated by members from another entity. You can think of a domain-based attribute as a constrained list; domain-based attributes prevent users from entering attribute values that are not valid. To select an attribute value, the user must pick from a list.

Domain-Based Attribute Example

In the following image, the Product entity has a domain-based attribute called Subcategory. The Subcategory attribute is populated by values from the Subcategory entity.

The Subcategory entity has a domain-based attribute called Category. The Category attribute is populated by values from the Category entity.



Use Same Entity for Multiple Domain-Based Attributes

You can use the same entity as a domain-based attribute of multiple entities. For example, you can create an entity called YesNoIndicator with the members: Yes, No, and Maybe. You can create a domain-based attribute named InStock and use the YesNoIndicator entity as the source. You can also create another domain-based attribute named Approved and use the YesNoIndicator entity as a source. Any time you want users to choose from a list of the YesNoIndicator entity's members, you can use the entity as a domain-based attribute.

Domain-Based Attributes Form Derived Hierarchies

Domain-based attribute relationships are the basis for derived hierarchies. For more information, see [Derived Hierarchies \(Master Data Services\)](#).

Related Tasks

Task Description	Topic
Create a new domain-based attribute that is sourced from an existing entity.	Create a Domain-Based Attribute (Master Data Services)
Create a new entity.	Create an Entity (Master Data Services)

Related Content

- [Derived Hierarchies \(Master Data Services\)](#)
- [Attributes \(Master Data Services\)](#)
- [Entities \(Master Data Services\)](#)

Create a Domain-Based Attribute (Master Data Services)

SQL Server 2012

In Master Data Services, create a domain-based attribute to populate an attribute's values with members from an entity.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- An entity must exist to use as the source of the attribute values. For example, to create a domain-based attribute based on the Color entity, you must first create the Color entity. For more information, see [Create an Entity \(Master Data Services\)](#).
- An entity must exist to create the attribute for. For more information, see [Create an Entity \(Master Data Services\)](#).

To create a domain-based attribute

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to create an attribute for.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page:
 - If the attribute is for leaf members, in the **Leaf member attributes** pane, click **Add leaf attribute**.
 - If the attribute is for consolidated members, in the **Consolidated member attributes** pane, click **Add consolidated attribute**.
 - If the attribute is for collections, in the **Collection attributes** pane, click **Add collection attribute**.
7. On the **Add Attribute** page, select the **Domain-based** option.
8. In the **Name** box, type a name for the attribute. It does not have to be the same name as the entity that you use for the source of the attribute values.
9. In the **Display pixel width** box, type the width of the attribute column to be displayed in the **Explorer** grid.
10. From the **Entity** list, choose the entity to be used to populate the attribute values.
11. Optional. Select **Enable change tracking** to track changes to groups of attributes. For more information, see [Add Attributes to a Change Tracking Group \(Master Data Services\)](#).
12. Click **Save attribute**.
13. On the **Entity Maintenance** page, click **Save entity**.

See Also

Tasks

[Create a Derived Hierarchy \(Master Data Services\)](#)

[Change an Attribute Name \(Master Data Services\)](#)

[Delete an Attribute \(Master Data Services\)](#)

Concepts

[Domain-Based Attributes \(Master Data Services\)](#)

Attribute Groups (Master Data Services)

SQL Server 2012

In Master Data Services, attribute groups help organize attributes in an entity. When an entity has lots of attributes, attribute groups improve the way an entity is displayed in the Master Data Manager web application.

How Attribute Groups Change the Display

Attribute groups are displayed as tabs above the grid in the **Explorer** functional area of Master Data Manager.

When an entity has a large number of attributes and you view that entity in a grid in **Explorer**, you must scroll to the right to view all of the attributes. To prevent this scrolling, you can create attribute groups.

- Attribute groups always include the Name and Code attributes.
- Each attribute for an entity can belong to one or more attribute groups.
- All attributes are automatically included on the **All Attributes** tab in **Explorer**.
- There is no way to hide the **All Attributes** tab.

Attribute groups are administered in the **System Administration** functional area of Master Data Manager.

Show or Hide Attribute Groups

When you create an attribute group, it is automatically hidden from all users except the one who created it. For more information about making the group visible, see [Make an Attribute Group Visible to Users \(Master Data Services\)](#).

If you want to hide a specific attribute within a group, you can assign **Deny** permission to the attribute. For more information, see [Leaf Permissions \(Master Data Services\)](#) or [Consolidated Permissions \(Master Data Services\)](#).

Related Tasks

Task Description	Topic
Create a new attribute group and add attributes to it.	Create an Attribute Group (Master Data Services)
Make an attribute group visible to users.	Make an Attribute Group Visible to Users (Master Data Services)
Change the name of an existing attribute group.	Change an Attribute Group Name (Master Data Services)
Delete an existing attribute group.	Delete an Attribute Group (Master Data Services)

Related Content

- [Attributes \(Master Data Services\)](#)

Create an Attribute Group (Master Data Services)

SQL Server 2012

In Master Data Services, create attribute groups when you want to display attributes on individual tabs in the **Explorer** grid.

Note

When you create an attribute group, it is automatically hidden from all users except the one who created it. For more information about making the group visible, see [Make an Attribute Group Visible to Users \(Master Data Services\)](#).

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- At least one attribute must exist. For more information, see [Create a Text Attribute \(Master Data Services\)](#).

To create an attribute group

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Attribute Groups**.
3. From the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. Click **Leaf Groups**, **Consolidated Groups**, or **Collection Groups** to create an attribute group of leaf members, consolidated members, or collections respectively.
6. Click **Add attribute group**.
7. In the **Leaf group name** box, type a name for the group. This is the name displayed on the tab in **Explorer**.

Note

If you selected **Consolidated Groups** or **Collection Groups** in step 5, this box is **Consolidated group name** or **Collection group name**, respectively.

8. Click **Save group**.
9. Expand the folder for the group.
10. Click **Attributes**.
11. Click **Edit selected item**.
12. Click attributes in the **Available** box and click the **Add** arrow. To add all, click the **Add All** arrow.
13. Optionally, click the **Up** and **Down** arrows to change the left-to-right order of the attributes.
14. Click **Save**.

Next Steps

- [Make an Attribute Group Visible to Users \(Master Data Services\)](#)

See Also

Tasks

[Change an Attribute Group Name \(Master Data Services\)](#)

[Delete an Attribute Group \(Master Data Services\)](#)

Concepts

[Attribute Groups \(Master Data Services\)](#)

[Attributes \(Master Data Services\)](#)

[Leaf Permissions \(Master Data Services\)](#)

[Consolidated Permissions \(Master Data Services\)](#)

Make an Attribute Group Visible to Users (Master Data Services)

SQL Server 2012

In Master Data Services, make an attribute group visible to users or groups when you want users to have tabs above the grid in the **Explorer** functional area.

When you create an attribute group, it is automatically hidden from all users except the one who created it.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- At least one attribute group must exist. For more information, see [Create an Attribute Group \(Master Data Services\)](#).

To make an attribute group visible to users

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Attribute Groups**.
3. From the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. Click the plus sign to expand **Leaf Groups**, **Consolidated Groups**, or **Collection Groups**, depending on the type of group you want to make visible.
6. Click the plus sign to expand the group you want to make visible.
7. Click either **Users** or **Groups**, depending on whether you are making the group visible to a user or a group.
8. Click **Edit selected item**.
9. Click users or groups in the **Available** box and click the **Add** arrow. To add all, click the **Add All** arrow.
10. Click **Save**.

See Also

Tasks

[Create an Attribute Group \(Master Data Services\)](#)

Concepts

[Attribute Groups \(Master Data Services\)](#)

Change an Attribute Group Name (Master Data Services)

SQL Server 2012

In Master Data Services, you can change the name of an attribute group.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To change an attribute group name

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Attribute Groups**.
3. From the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. Click the plus sign to expand **Leaf Groups**, **Consolidated Groups**, or **Collection Groups**, depending on the type of group you want to update.
6. Click the name of the attribute group with the name you want to change.
7. Click **Edit selected item**.
8. In the **Leaf group name** box, **Consolidated group name** box, or **Collection group name** box, type the updated name of the attribute group.
9. Click **Save group**.

See Also

Tasks

[Create an Attribute Group \(Master Data Services\)](#)

[Delete an Attribute Group \(Master Data Services\)](#)

Concepts

[Attribute Groups \(Master Data Services\)](#)

Delete an Attribute Group (Master Data Services)

SQL Server 2012

In Master Data Services, delete an attribute group when you no longer need the tab to display in the **Explorer** functional area of Master Data Manager.

- **Note** When attribute groups exist, attributes that do not belong to an attribute group are not displayed in **Explorer**. When no attribute groups exist, all attributes are displayed.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete an attribute group

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Attribute Groups**.
3. From the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. Click the plus sign to expand **Leaf Groups**, **Consolidated Groups**, or **Collection Groups**, depending on the type of group you want to delete.
6. Click the attribute group you want to delete.
7. Click **Delete selected group**.
8. In the confirmation dialog box, click **OK**.
9. In the additional confirmation dialog box, click **OK**.

See Also

Tasks

[Create an Attribute Group \(Master Data Services\)](#)

Concepts

[Attribute Groups \(Master Data Services\)](#)

Importing Data (Master Data Services)

SQL Server 2012

You can populate the Master Data Services database with data from your organization's disparate data sources. For data to be in the appropriate Master Data Services database tables, you must import data into staging tables and then process the staged data as a batch.

You can use the staging process to:

- Create, update, deactivate, and delete leaf and consolidated members.
- Update attribute values.
- Designate relationships in explicit hierarchies.

Note

The staging process has changed since SQL Server 2008 R2. You can continue to use the SQL Server 2008 R2 process; however, the staging process is now more flexible and has better performance. For more information on the changes, see [Deprecated Master Data Services Features in SQL Server 2012](#) and [Discontinued Master Data Services Features in SQL Server 2012](#).

Staging Tables

The Master Data Services database includes three staging tables that you can populate with your organization's data. After you populate these tables, initiate the staging process to load the data from the staging tables into the appropriate MDS tables.

The name of each table is specified when an entity is created and cannot be changed. If the staging table name contains a _1 or other number, another table of that name already existed when the entity was created.

Table Name	Description
stg.name_Leaf	Create, update, and delete leaf members and their attributes.
stg.name_Consolidated	Create, update, and delete consolidated members and their attributes.
stg.name_Relationship	Move members in an explicit hierarchy.

Initiating the Staging Process

You can initiate the staging process by using stored procedures or by using the Master Data Manager web application.

Using Stored Procedures to Initiate the Staging Process

The Master Data Services database includes stored procedures associated with the staging tables. Use these stored procedures to load data from the staging tables into the appropriate MDS tables. The stored procedures are named similarly to the associated tables:

- [stg.udp_name_Leaf](#)
- [stg.udp_name_Consolidated](#)
- [stg.udp_name_Relationship](#)

For more information on these stored procedures, see [Staging Stored Procedure \(Master Data Services\)](#).

Using the Web Application to Initiate the Staging Process

You can use the web application's **Integration Management** functional area to start the staging process. The staging process in the UI is started at intervals determined by the **Staging batch interval** setting in Master Data Services Configuration Manager.

Logging Transactions

All transactions that occur when data or relationships are imported or updated can be logged. An option in the stored procedure allows this logging. If you initiate the staging process from the Master Data Manager web application, no logging occurs.

In Master Data Services Configuration Manager, the **Log staging transactions** setting does not apply to this method of staging data.

Validating Data

When you load data by using the staging process, the data is not automatically validated against business rules. You must validate the data manually or by using a stored procedure.

- For information on validating manually, see [Validate a Version against Business Rules \(Master Data Services\)](#).

- For information on using a stored procedure, see [Validation Stored Procedure \(Master Data Services\)](#).

For more information on what validation is and when it occurs, see [Validation \(Master Data Services\)](#).

Related Tasks

Task Description	Topic
Create, update, and delete members and their attributes.	Load or Update Members in Master Data Services by Using the Staging Process
Move members in an explicit hierarchy.	Move Explicit Hierarchy Members by Using the Staging Process (Master Data Services)
Troubleshoot the staging process.	Troubleshooting the Staging Process (Master Data Services)

Related Content

- [Validation \(Master Data Services\)](#)
- [Business Rules \(Master Data Services\)](#)

Load or Update Members in Master Data Services by Using the Staging Process

SQL Server 2012

In Master Data Services, you can use the staging process to create, update, and delete members in bulk. You can also use the staging process to set and update attribute values.

Prerequisites

To perform this procedure:

- You must have permission to insert data into the `stg.name_Leaf` (for leaf members) or the `stg.name_Consolidated` (for consolidated members) table in the Master Data Services database.
- You must have permissions to execute either the `stg.udp_name_Leaf` (for leaf members) or `stg.udp_name_Consolidated` (for consolidated members) stored procedure in the Master Data Services database.
- The model must not have a status of **Committed**.

To load members into the MDS database

1. Prepare the members for import into the appropriate staging table in the Master Data Services database.
 - For leaf members, use `stg.name_Leaf`. For more information about the required fields, see [Leaf Member Staging Table \(Master Data Services\)](#).
 - For consolidated members, use `stg.name_Consolidated`. For more information about the required fields, see [Consolidated Member Staging Table \(Master Data Services\)](#).
2. Open SQL Server Management Studio and connect to the Database Engine instance for your Master Data Services database.
3. Import the data into the Master Data Services database. For information about importing data into a database by using SQL Server Integration Services, see [Run the SQL Server Import and Export Wizard](#).
4. After you successfully import the data, run the stored procedure to load the staged data into the appropriate tables in the MDS database. For more information on the stored procedure, see [Staging Stored Procedure \(Master Data Services\)](#).

Next Steps

- View errors that occurred during staging. For more information, see [View Errors that Occur During the Staging Process \(Master Data Services\)](#).

See Also

Concepts

[Importing Data \(Master Data Services\)](#)

[Staging Process Errors \(Master Data Services\)](#)

Other Resources

[Troubleshooting the Staging Process \(Master Data Services\)](#)

Deactivate or Delete Members by Using the Staging Process (Master Data Services)

SQL Server 2012

In Master Data Services, you can deactivate members or delete them completely.

Deactivating means the member can be reactivated. If you reactivate a member, its attributes and its membership in hierarchies and collections are restored. All previous transactions are intact. Deactivation transactions are visible to administrators in the **Version Management** functional area of the Master Data Manager web application.

Deleting means purging the member from the system permanently. All transactions for the member, all relationships, and all attributes are permanently deleted.

Notes:

- You cannot use staging to reactivate members. You must do it manually in the Master Data Manager web application. For more information, see [Reactivate a Member or Collection \(Master Data Services\)](#).
- You cannot use staging to delete or deactivate collections. For more information on manually deactivating collections, see [Delete a Member or Collection \(Master Data Services\)](#).

Prerequisites

To perform this procedure:

- You must have permission to insert data into the stg.name_Leaf (for leaf members) or stg.name_Consolidated (for consolidated members) table in the Master Data Services database.
- You must have permission to execute either the stg.udp_name_Leaf (for leaf members) or stg.udp_name_Consolidated (for consolidated members) stored procedure in the Master Data Services database.
- The model must not have a status of **Committed**.

To deactivate or delete members

- Follow the procedure to load or update members. For more information, see [Load or Update Members in Master Data Services by Using the Staging Process](#). For the **ImportType** field, use **3** to deactivate members, and **4** to permanently delete.

See Also

Tasks

[Reactivate a Member or Collection \(Master Data Services\)](#)

[Create a Leaf Member \(Master Data Services\)](#)

[Delete a Member or Collection \(Master Data Services\)](#)

Move Explicit Hierarchy Members by Using the Staging Process (Master Data Services)

SQL Server 2012

In Master Data Services, you can use the staging process to move the location of members in explicit hierarchies in bulk. You can designate:

- A consolidated member as a parent of a consolidated member.
- A consolidated member as a parent of a leaf member.
- A leaf member as a sibling of a leaf or consolidated member.
- A consolidated member as a sibling of a leaf or consolidated member.

Prerequisites

To perform this procedure:

- You must have permission to insert data into the stg.name_Relationship table in the Master Data Services database.
- You must have permission to execute the stg.udp_name_Relationship stored procedure in the Master Data Services database.
- The model must not have a status of **Committed**.

To move explicit hierarchy members

1. Prepare the members for import into the stg.name_Relationship table in the Master Data Services database. For more information about the required fields, see [Relationship Staging Table \(Master Data Services\)](#).
2. Open SQL Server Management Studio and connect to the Database Engine instance for your Master Data Services database.
3. Import the data into stg.name_Relationship. For information about importing data into a database by using SQL Server Integration Services, see [Run the SQL Server Import and Export Wizard](#).
4. After you successfully import the data, run the stg.udp_name_Relationship stored procedure to load the staged data into the appropriate MDS tables. For more information about the stored procedure, see [Staging Stored Procedure \(Master Data Services\)](#)

Next Steps

- View errors that occurred during staging. For more information, see [View Errors that Occur During the Staging Process \(Master Data Services\)](#).

See Also

Tasks

[Move Members within a Hierarchy \(Master Data Services\)](#)

Concepts

[Importing Data \(Master Data Services\)](#)

[Staging Process Errors \(Master Data Services\)](#)

Other Resources

[Troubleshooting the Staging Process \(Master Data Services\)](#)

View Errors that Occur During the Staging Process (Master Data Services)

SQL Server 2012

In Master Data Services, you can view errors that occur during the staging process. In the Master Data Services database, there are two views that show errors:

- `stg.viw_name_MemberErrorDetails` for leaf or consolidated member updates.
- `stg.viw_name_RelationshipErrorDetails` for hierarchy relationship updates.

Prerequisites

To perform this procedure:

- In the Master Data Services database, you must have SELECT permissions to either the `stg.viw_name_MemberErrorDetails` or `stg.viw_name_RelationshipErrorDetails` view.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To view staging errors

1. Open SQL Server Management Studio and connect to the Database Engine instance for your Master Data Services database.
2. Open a new query.
3. Type the following text, replacing name with the name of your staging table, for example, `viw_Product_MemberErrorDetails`.

```
SELECT * FROM stg.vi w_name_MemberErrorDetail s
```

4. Execute the query. Error details are displayed in the **ErrorDescription** field.

Next Steps

For more details on error messages, see [Staging Process Errors \(Master Data Services\)](#).

See Also

Concepts

[Importing Data \(Master Data Services\)](#)

Other Resources

[Troubleshooting the Staging Process \(Master Data Services\)](#)


Leaf Member Staging Table (Master Data Services)

SQL Server 2012

Use the leaf members staging table (stg.name_Leaf) in the Master Data Services database to create, update, deactivate, and delete leaf members. You can also use it to update attribute values for leaf members.

Table Columns

The following table explains what each of the fields in the Leaf staging table are used for.

Column Name	Description
ID	An automatically assigned identifier. Do not enter a value in this field. If the batch has not been processed, this field is blank.
ImportType Required	Determines what to do when staged data matches data that already exists in the Master Data Services database. <ul style="list-style-type: none">● 0: Create new members. Replace existing MDS data with staged data, but only if the staged data is not NULL. NULL values are ignored. To change a string attribute value to NULL, set it to ~NULL~. To change a number attribute value to NULL, set it to -98765432101234567890. To change a datetime attribute value to NULL, set it to 5555-11-22T12:34:56.● 1: Create new members only. Any updates to existing MDS data fail.● 2: Create new members. Replace existing MDS data with staged data. If you import NULL values, they will overwrite existing MDS values.● 3: Deactivate the member, based on the Code value. All attributes, hierarchy and collection memberships, and transactions are maintained but no longer available in the UI. If the member is used as a domain-based attribute value of another member, the deactivation will fail. See ImportType 5 for an alternative.● 4: Permanently delete the member, based on the Code value. All attributes, hierarchy and collection memberships, and transactions are permanently deleted. If the member is used as a domain-based attribute value of another member, the deletion will fail. See ImportType 6 for an alternative.● 5: Deactivate the member, based on the Code value. All attributes, hierarchy and collection memberships, and transactions are maintained but no longer available in the UI. If the member is used as a domain-based attribute value of other members, the related values will be set to NULL. ImportType 5 is for leaf members only.● 6: Permanently delete the member, based on the Code value. All attributes, hierarchy and collection memberships, and transactions are permanently deleted. If the member is used as a domain-based attribute value of other members, the related values will be set to NULL. ImportType 6 is for leaf members only.
ImportStatus_ID Required	The status of the import process. Possible values are: <ul style="list-style-type: none">● 0, which you specify to indicate that the record is ready for staging.● 1, which is automatically assigned and indicates that the staging process for the record has succeeded.● 2, which is automatically assigned and indicates that the staging process for the record has failed.
Batch_ID Required by web service only	An automatically assigned identifier that groups records for staging. All members in the batch are assigned this identifier, which is displayed in the Master Data Manager user interface in the ID column. If the batch has not been processed, this field is blank.
BatchTag Required, except by web service	A unique name for the batch, up to 50 characters.
ErrorCode	Displays an error code. For all records with a ImportStatus_ID of 2 , see Staging Process Errors (Master Data Services) .
Code Required, except when codes are generated automatically for ImportType 1 or 2 : see Automatic Code Creation (Master Data Services) for more information	A unique code for the member.
Name Optional	A name for the member.
NewCode	Use only if you are changing the member code.
<Attribute name>	<p>A column exists for each attribute in the entity. Use this with an ImportType of 0 or 2. For free-form attributes, specify the new text or string value for the attribute. For domain-based attributes, specify the code for the member that will be the attribute. For link attributes, the URL must start with http://.</p> <div> Note You cannot stage file attributes.</div>

▲ See Also

Tasks

[Load or Update Members in Master Data Services by Using the Staging Process](#)
[View Errors that Occur During the Staging Process \(Master Data Services\)](#)

Concepts

[Importing Data \(Master Data Services\)](#)
[Staging Process Errors \(Master Data Services\)](#)


Consolidated Member Staging Table (Master Data Services)

SQL Server 2012

Use the consolidated members staging table (stg.name_Consolidated) in the Master Data Services database to create, update, deactivate, and delete consolidated members. You can also use it to update attribute values for consolidated members.

Table Columns

The following table explains what each of the fields in the Consolidated staging table are used for.

Column Name	Description
ID	An automatically assigned identifier. Do not enter a value in this field. If the batch has not been processed, this field is blank.
ImportType Required	Determines what to do when staged data matches data that already exists in the Master Data Services database. <ul style="list-style-type: none">• 0: Create new members. Replace existing MDS data with staged data, but only if the staged data isn't NULL. NULL values are ignored. To change an attribute value to NULL, use ~NULL~.• 1: Create new members only. Any updates to existing MDS data fail.• 2: Create new members. Replace existing MDS data with staged data. If you import NULL values, they will overwrite existing MDS values.• 3: Deactivate the member, based on the Code value. All attributes, hierarchy and collection memberships, and transactions are maintained but no longer available in the UI. If the member is used as a domain-based attribute value of another member, the deactivation will fail.• 4: Permanently delete the member, based on the Code value. All attributes, hierarchy and collection memberships, and transactions are permanently deleted. If the member is used as a domain-based attribute value of another member, the deletion will fail.
ImportStatus_ID Required	The status of the import process. Possible values are: <ul style="list-style-type: none">• 0, which you specify to indicate that the record is ready for staging.• 1, which is automatically assigned and indicates that the staging process for the record has succeeded.• 2, which is automatically assigned and indicates that the staging process for the record has failed.
Batch_ID Required by web service only	An automatically assigned identifier that groups records for staging. All members in the batch are assigned this identifier, which is displayed in the Master Data Manager user interface in the ID column. If the batch has not been processed, this field is blank.
BatchTag Required, except by web service	A unique name for the batch, up to 50 characters.
HierarchyName Required	The explicit hierarchy name. Each consolidated member can belong to one hierarchy only.
ErrorCode	Displays an error code. For all records with a ImportStatus_ID of 2 , see Staging Process Errors (Master Data Services) .
Code Required, except when codes are generated automatically for ImportType 1 or 2 ; see Automatic Code Creation (Master Data Services) for more information	A unique code for the member.
Name Optional	A name for the member.
NewCode	Use only if you are changing the member code.
<Attribute name>	A column exists for each attribute in the entity. Use this with an ImportType of 0 or 2 . For free-form attributes, specify the new text or string value for the attribute. For domain-based attributes, specify the code for the member that will be the attribute. For link attributes, the URL must start with http:// . <div> Note You cannot stage file attributes.</div>

See Also

Tasks

[Load or Update Members in Master Data Services by Using the Staging Process](#)

[Move Explicit Hierarchy Members by Using the Staging Process \(Master Data Services\)](#)

[View Errors that Occur During the Staging Process \(Master Data Services\)](#)

Concepts

[Importing Data \(Master Data Services\)](#)

[Staging Process Errors \(Master Data Services\)](#)

Relationship Staging Table (Master Data Services)

SQL Server 2012

Use the relationship staging table (stg.name_Relationship) in the Master Data Services database to change the location of members in an explicit hierarchy, based on the relationship the members have to one another.

Table Columns

The following table explains what each of the fields in the Relationship staging table are used for.

Column Name	Description
ID	An automatically assigned identifier. Do not enter a value in this field. If the batch has not been processed, this field is blank.
RelationshipType Required	The type of relationship that's being set. Possible values are: <ul style="list-style-type: none">• 1:Parent• 2: Sibling (at the same level)
ImportStatus_ID Required	The status of the import process. Possible values are: <ul style="list-style-type: none">• 0, which you specify to indicate that the record is ready for staging.• 1, which is automatically assigned and indicates that the staging process for the record has succeeded.• 2, which is automatically assigned and indicates that the staging process for the record has failed.
Batch_ID Required by web service only	An automatically assigned identifier that groups records for staging. All members in the batch are assigned this identifier, which is displayed in the Master Data Manager user interface in the ID column. If the batch has not been processed, this field is blank.
BatchTag Required, except by web service	A unique name for the batch, up to 50 characters.
HierarchyName Required	The explicit hierarchy name. Each consolidated member can belong to one hierarchy only.
ParentCode Required	For parent-child relationships, the code of the consolidated member that will be the parent of the child leaf or consolidated member. For sibling relationships, the code of one of the siblings.
ChildCode Required	For parent-child relationships, the code of the consolidated or leaf member that will be the child. For sibling relationships, the code of one of the siblings.
Sort Order Optional	An integer that indicates the order of the member in relation to the other members under the parent. Each child member should have a unique identifier.
ErrorCode	Displays an error code. For all records with a ImportStatus_ID of 2 , see Staging Process Errors (Master Data Services) .

See Also

Tasks

[Move Explicit Hierarchy Members by Using the Staging Process \(Master Data Services\)](#)

[View Errors that Occur During the Staging Process \(Master Data Services\)](#)

Concepts

[Importing Data \(Master Data Services\)](#)

[Staging Process Errors \(Master Data Services\)](#)

Staging Stored Procedure (Master Data Services)

SQL Server 2012

When initiating the staging process from SQL Server Management Studio, you use one of three stored procedures.

- stg.udp_name_Leaf
- stg.udp_name_Consolidated
- stg.udp_name_Relationship

Where name is the name of the staging table that was specified when the entity was created.

▀ Staging Process Stored Procedure Parameters

The following table lists the parameters of these stored procedures.

Parameter	Description
VersionName Required	The name of the version. This may or may not be case-sensitive, depending on your SQL Server collation setting.
LogFlag Required	Determines whether transactions are logged during the staging process. Possible values are: <ul style="list-style-type: none">• 0: Do not log transactions.• 1: Log transactions. For more information about transactions, see Transactions (Master Data Services) .
BatchTag Required, except by web service	The BatchTag value as specified in the staging table.
Batch_ID Required by web service only	The Batch_ID value as specified in the staging table.

Staging Process Stored Procedure Example

The following example shows how to start the staging process by using the staging stored procedure.

```
USE [DATABASE_NAME]
GO

EXEC[stg].[udp_name_Leaf]
    @VersionName = N' VERSION_1' ,
    @LogFlag = 1,
    @BatchTag = N' batch1'

GO
```

▀ See Also

- Tasks
- [Load or Update Members in Master Data Services by Using the Staging Process](#)
 - [View Errors that Occur During the Staging Process \(Master Data Services\)](#)
- Concepts
- [Validation Stored Procedure \(Master Data Services\)](#)

Staging Process Errors (Master Data Services)

SQL Server 2012

When the staging process is complete, all processed records in the staging tables have a value in the ErrorCode column. These values are listed in the following table.

Code	Error	Occurs When/Details	Applies to Table
210001	The same member code exists multiple times in the staging table.	Your staging batch includes the same member code multiple times. Neither member is created or updated.	Leaf Consolidated Relationship
210003	The attribute values references a member that does not exist or is inactive.	When you stage domain-based attributes, you must use the code, rather than the name. Applies to ImportType 0, 1, and 2 .	Leaf Consolidated
210006	The member code is inactive.	ImportType = 1 and you specified a member code that doesn't exist.	Leaf Consolidated Relationship
210032	The hierarchy name is missing or is not valid.	The explicit hierarchy was not found or the HierarchyName value was blank.	Consolidated Relationship
210035	Because a code generation business rule does not exist, the MemberCode is required.	When creating or updating members, a MemberCode is always required, unless you are using automatic code generation. For more information, see Automatic Code Creation (Master Data Services) .	Leaf Consolidated
210036	Because a code generation business rule exists, the MemberCode is not required.	When creating or updating members, a MemberCode is not required when you are using automatic code generation. However, you can specify a code if you choose. For more information, see Automatic Code Creation (Master Data Services) .	Leaf Consolidated
210041	"ROOT" is not a valid member code.	The MemberCode value contains the word "ROOT."	Leaf Consolidated Relationship
210042	"MDMUNUSED" is not a valid member code.	The MemberCode value contains the word "MDMUNUSED."	Leaf Consolidated Relationship
210052	The MemberCode cannot be deactivated because it is used as a domain-based attribute value.	When ImportType = 3 or 4 , staging fails if the member is used as an attribute value for other members. Either use ImportType 5 or 6 to set the value to NULL, or change the values before running the staging process.	Leaf Consolidated
300002	The member code is not valid.	Relationships: Either the parent or child member code does not exist. Leaf or Consolidated: ImportType = 3 or 4 and the member code does not exist.	Leaf Consolidated Relationship
300003	The member code already exists.	ImportType = 1 and you used a member code that already exists in the entity.	Leaf Consolidated
210011	When RelationshipType is 1, the ParentCode cannot be a leaf member.	Ensure that the ParentCode value is a consolidated member code.	Relationship
210015	The member code exists multiple times in the staging table for a hierarchy and a batch.	For an explicit hierarchy, you specified the location of the same member multiple times in the same batch.	Relationship
210016	The relationship could not be created because it would cause a circular reference.	This occurs when you try to assign a child as a parent.	Relationship
210046	The member cannot be a sibling of Root.	This occurs when RelationshipType = 2 (sibling) and either the ParentCode or ChildCode is Root . Members cannot be at the same level as the Root node; they can only be children.	Relationship
210047	The member cannot be a sibling of Unused.	This occurs when RelationshipType = 2 (sibling) and either the ParentCode or ChildCode is Unused . Members can only be children of the Unused node.	Relationship
210048	ParentCode and ChildCode cannot be the same.	The ParentCode value is the same as the ChildCode value. These values must be different.	Relationship
210043	The TargetType_ID must be 1 (parent) or 2 (sibling).	The TargetType_ID is neither 1 nor 2.	Relationship
210058	The import type is not valid.	The import type is not valid.	Leaf Consolidated

▲ See Also

Tasks

[View Errors that Occur During the Staging Process \(Master Data Services\)](#)

Concepts

[Importing Data \(Master Data Services\)](#)

Import Statuses (Master Data Services)

SQL Server 2012

In the **Integration Management** functional area, on the **Staging Batches** page, the following statuses are possible.

Status	Description	Status_ID
Queued to run	The batch has not started processing.	1
Running	The batch is processing.	2
Completed	The batch has finished processing.	3
Queued to clear	The batch has finished processing and will be cleared.	4
Cleared	The batch has been cleared.	5

See Also

- Tasks
 - [Load or Update Members in Master Data Services by Using the Staging Process](#)
- Concepts
 - [Importing Data \(Master Data Services\)](#)

Members (Master Data Services)

SQL Server 2012

In Master Data Services, members are the physical master data. For example, a member can be a Road-150 bike in a Product entity or a specific customer in a Customer entity.

How Members Relate to Other Model Objects

You can think of members as rows in a table. Related members are contained in an entity, and each member is defined by attribute values.

In this example, the table represents an entity, the rows in the table represent members, and the columns in the table represent attributes. Each cell represents an attribute value for a specific member.

Members (rows)	Attributes (columns)			
	attribute value	attribute value	attribute value	attribute value
	attribute value	attribute value	attribute value	attribute value
	attribute value	attribute value	attribute value	attribute value
	attribute value	attribute value	attribute value	attribute value

Member Types

There are three types of members: leaf members, consolidated members, and collection members.

Leaf members are the default members in an entity.

- In derived hierarchies, leaf members are the only type of member. Leaf members from one entity are used as parents of leaf members from another entity.
- In explicit hierarchies, leaf members are the lowest level and cannot have children.

Consolidated members exist only when explicit hierarchies and collections are enabled for the entity.

- Derived hierarchies do not contain consolidated members.
- In explicit hierarchies, consolidated members can be parents of other members within the hierarchy, or they can be children.

Use Hierarchies and Collections to Organize Members

Hierarchies and collections can be used to group members for reporting or analysis. For more information, see [Hierarchies \(Master Data Services\)](#) and [Collections \(Master Data Services\)](#).

Member Example

In the following example, each member is made up of a Name, Code, Subcategory, StandardCost, ListPrice, and FilePhoto attribute value.

Name	Code	Subcategory	StandardCost	ListPrice	FilePhoto
Mountain-100	BK-M101	Mountain Bikes {5}	1912.15	3399.99	
Mountain-200	BK-M201	Mountain Bikes {5}	1265.62	2319.99	
Road-150	BK-R150	Road Bikes {6}	2171.29	3578.27	
Road-450	BK-450	Road Bikes {6}	884.71	1457.99	

Related Tasks

Task Description	Topic
Create a new leaf member.	Create a Leaf Member (Master Data Services)
Create a new consolidated member.	Create a Consolidated Member (Master Data Services)
Delete an existing member or collection.	Delete a Member or Collection (Master Data Services)
Reactivate a deleted member or collection.	Reactivate a Member or Collection (Master Data Services)
Update the attribute values of a member.	Change the Attribute Type (MDS Add-in for Excel)
Move members within a hierarchy.	Move Members within a Hierarchy (Master Data Services)

▲ Related Content

- [Master Data Services Overview](#)
- [Entities \(Master Data Services\)](#)
- [Attributes \(Master Data Services\)](#)
- [Hierarchies \(Master Data Services\)](#)
- [Collections \(Master Data Services\)](#)
- [Leaf Permissions \(Master Data Services\)](#)
- [Consolidated Permissions \(Master Data Services\)](#)
- [Filter Operators \(Master Data Services\)](#)

Create a Leaf Member (Master Data Services)

SQL Server 2012

In Master Data Manager, create a leaf member when you want to add master data to your system and are not using staging tables or the Master Data Services Add-in for Excel to import data.

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.
- You must have a minimum of **Update** permission to the leaf model object for the entity you are adding a member to.

To create a leaf member

1. On the Master Data Manager home page, from the **Model** list, select a model.
2. If you are a user, select an open version from the **Version** list. If you are an administrator, select a version with open or locked status from the **Version** list.
3. Click **Explorer**.
4. From the menu bar, point to **Entities** and click the name of the entity you want to add a member to.
5. Click **Add member**.
6. In the **Details** pane, complete the fields.
7. Optional. In the **Annotations** box, type a comment about why the member was added. All users who have access to the member can view the annotation.
8. Click **OK**.

See Also

Tasks

[Load or Update Members in Master Data Services by Using the Staging Process](#)

[Create a Consolidated Member \(Master Data Services\)](#)

Concepts

[Members \(Master Data Services\)](#)

Create a Consolidated Member (Master Data Services)

SQL Server 2012

In Master Data Manager, create a consolidated member when you want a parent node for an explicit hierarchy.

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.
- You must have a minimum of **Update** permission to the consolidated model object for the entity you are adding a member to.

To create a consolidated member

1. On the Master Data Manager home page, from the **Model** list, select a model.
2. From the **Version** list, select a version.
3. Click **Explorer**.
4. From the menu bar, point to **Hierarchies** and click the name of the hierarchy you want to add a consolidated member to.
5. Above the grid, select either the **Consolidated members** or the **All consolidated members in hierarchy** option.
6. Click **Add**.
7. In the pane on the right, complete the fields.
8. Optional. In the **Annotations** box, type a comment about why the member was added. All users who have access to the member can view the annotation.
9. Click **OK**.

Next Steps

- [Move Members within a Hierarchy \(Master Data Services\)](#)

See Also

Tasks

[Create an Explicit Hierarchy \(Master Data Services\)](#)

[Create a Leaf Member \(Master Data Services\)](#)

[Load or Update Members in Master Data Services by Using the Staging Process](#)

Concepts

[Members \(Master Data Services\)](#)

[Explicit Hierarchies \(Master Data Services\)](#)

Delete a Member or Collection (Master Data Services)

SQL Server 2012

In Master Data Manager, delete a member or collection when you no longer need it. If you want to delete members in bulk, use the staging process instead. For more information, see [Deactivate or Delete Members by Using the Staging Process \(Master Data Services\)](#)

Note

You cannot delete a member if it is used as a domain-based attribute value for another member.

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.
- For members, you must have a minimum of **Update** permission to the leaf model object you are deleting a member from.
- For collections, you must have a minimum of **Update** permission to the leaf collection object you are deleting.

To delete a member or collection

1. On the Master Data Manager home page, from the **Model** list, select a model.
2. From the **Version** list, select a version.
3. Click **Explorer**.
4. To delete:
 - A leaf member, from the menu bar, point to **Entities** and click the name of the entity that contains the member.
 - A consolidated member, from the menu bar, point to **Hierarchies** and click the name of the hierarchy that contains the member. Then click the node in the hierarchy that contains the member.
 - A collection, from the menu bar, point to **Collections** and click the name of the entity that contains the collection.
5. In the grid, click the row of the member or collection you want to delete.
6. Click **Delete Member**, **Delete**, or **Delete Collection**.
7. In the confirmation dialog box, click **OK**.

See Also

Tasks

[Reactivate a Member or Collection \(Master Data Services\)](#)

Concepts

[Members \(Master Data Services\)](#)

[Collections \(Master Data Services\)](#)

Reactivate a Member or Collection (Master Data Services)

SQL Server 2012

In Master Data Services, you can reactivate a member that was either:

- Deactivated by the staging process.
- Deleted in the MDS Add-in for Excel.
- Deleted in the Master Data Manager web application.

When you reactivate a member, its attributes and its membership in hierarchies and collections are restored.

You can also reactivate collections. When you do, the collection's attributes and the members that belong to the collection are restored.

When either a collection or member is reactivated, all previous transactions are restored.

Prerequisites

To perform this procedure:

- In Master Data Manager, you must have permission to the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To reactivate a member or collection

1. On the Master Data Manager home page, click **Version Management**.
2. On the menu bar, click **Transactions**.
3. On the **Transactions** page, from the **Model** list, select a model.
4. From the **Version** list, select a version.
5. In the **Transactions** pane, click the row for the member or collection you want to reactivate. This row should have **Active** displayed in the **Prior Value** column and **De-Activated** in the **New Value** column.
6. Click **Reverse Transaction**.
7. On the confirmation dialog box, click **OK**. A new transaction is added, showing **Active** in the **New Value** column.

See Also

Tasks

[Deactivate or Delete Members by Using the Staging Process \(Master Data Services\)](#)

[Delete a Member or Collection \(Master Data Services\)](#)

Concepts

[Members \(Master Data Services\)](#)

[Collections \(Master Data Services\)](#)

Move Members within a Hierarchy (Master Data Services)

SQL Server 2012

In Master Data Manager, move members within a hierarchy to change their location or parent assignment.

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.
- For explicit hierarchies, you must have a minimum of **Update** permission to the entity.
- For derived hierarchies, you must have a minimum of **Update** to the model and to any domain-based attributes used in the hierarchy.

To move members within a hierarchy

1. On the Master Data Manager home page, from the **Model** list, select a model.
2. From the **Version** list, select a version.
3. Click **Explorer**.
4. From the menu bar, point to **Hierarchies** and click *hierarchy_name*.
5. In the **Hierarchy** pane, where the hierarchy is displayed in a tree structure, click the check box for each member you want to move.
6. At the top of the **Hierarchy** pane, click **Cut**.
7. In the **Hierarchy** pane, click the check box for the member you want to move the members to.
8. Click **Paste**.

Note

In derived hierarchies, you can move members to the same level only. Also, you cannot change the sort order of members.

See Also

Tasks

[Move Explicit Hierarchy Members by Using the Staging Process \(Master Data Services\)](#)

Concepts

[Derived Hierarchies \(Master Data Services\)](#)

[Explicit Hierarchies \(Master Data Services\)](#)

Filter Operators (Master Data Services)

SQL Server 2012

When filtering a list of members, the following operators are available.

Note

When you filter by multiple criteria, all criteria must be true to return results. For example, SquareFeet = 2000 **AND** Division <> 123.

Filter Operators

Control Name	Description
Is equal to	Returns attribute values that are exactly the same as the specified criteria. For example, to filter on Mountain-100 , you must type Mountain-100 .
Is not equal to	Returns attribute values that are not exactly the same as the specified criteria. The filter criteria must be exactly the same as the attribute value you want to omit from the results. For example, to omit results that match Mountain-100 , you must type Mountain-100 . <div><div>Note</div><p>When you apply a filter condition with an "Is not equal" clause on an attribute, a member for which the attribute is NULL will pass the filter condition and be returned if SET ANSI_NULLS is set to ON in your database settings. To stop this behavior, turn SET ANSI_NULLS to OFF in your database settings. When SET ANSI_NULLS is set to OFF, comparisons of all data against a null value evaluate to TRUE if the data value is NULL, with the result that the member would not pass the "Is not equal" clause. For more information, see SET ANSI_NULLS (Transact-SQL).</p></div>
Is like	Uses the LIKE operator from Transact-SQL to filter results. For more information, see LIKE (Transact-SQL) in SQL Server Books Online.
Is not like	Uses the NOT operator from Transact-SQL to filter results. For more information, see NOT (Transact-SQL) in SQL Server Books Online.
Is greater than	Returns attribute values that are greater than the specified criteria. For example, to return attribute values that start with a letter greater than F , type F .
Is less than	Returns attribute values that are less than the specified criteria. For example, to return attribute values that start with a letter less than F , type F .
Is greater than or equal to	Returns attribute values that are greater than or equal to the specified criteria. For example, to return attribute values that start with the number 3 or greater, type 3 .
Is less than or equal to	Returns attribute values that are less than or equal to the specified criteria. For example, to return attribute values that start with the number 3 or less, type 3 .
Matches	Uses a fuzzy lookup index to filter results. Use the Similarity Level field to specify how closely the attribute values must match the specified filter criteria (with a default of 30%). Select one of the following in the Algorithm list box: <ul style="list-style-type: none">• Levenshtein: A distance that is based upon the number of edits (for example, adds or deletions) that it takes for one string to match another. This is the default. Does not require any additional parameters.• Jaccard: An index that works best when trying to match multiple strings. This search supports an additional parameter of containment bias (see below).• Jaro-Winkler: A distance that is best used for finding duplicate person names. This method returns more results than any other method. Does not support containment bias.• Longest Common Subsequence: Works based upon a subsequence in which the letters in a pattern appear in order, although they can be separated (for example, "MSR" is a subsequence of "MaSteR"). This search supports an additional parameter of containment bias (see below). Add a Containment Bias for the Jaccard or Longest Common Subsequence algorithm. This is a length threshold that is provided in a decimal percentage between 0 and 1, with a default of .62. A lower threshold will increase the number of possible matches returned.
Does not match	Uses a fuzzy lookup index to filter results. Use the Similarity Level field to specify how closely the attribute values must not match the specified filter criteria.
Contains pattern	Uses .NET Framework regular expressions to filter results on a specified pattern. For more information about regular expressions, see Regular Expression Language Elements in the MSDN Library.
Does not contain pattern	Uses the .NET Framework regular expressions to filter results that do not match a specified pattern. For more information about regular expressions, see Regular Expression Language Elements in the MSDN Library.
Is NULL	Returns attribute values that are null. The Criteria field disables when you select the Is NULL operator.

Is not NULL	Returns attribute values that are not null. The Criteria field disables when you select the Is not NULL operator.
--------------------	---

Reserved Words (Master Data Services)

SQL Server 2012

In Master Data Services, when you create model objects or members, some words cannot be used. Using these words may cause errors.

Note

You should also limit your use of special characters (symbols, hyphenation, etc.).

- [Models](#)
- [Entities](#)
- [Explicit Hierarchies](#)
- [Attributes](#)
- [Members](#)

Models

If you create a model with the name set to **Name**, do not select **Create entity with same name as model** because **Name** cannot be used for the name of an entity.

Entities

For entity names, you cannot use **Name** or **Code**.

Explicit Hierarchies

For explicit hierarchy names, you cannot use **Name** or **Code**.

Attributes

- ID
- Code
- Name
- EnterDTM
- EnterUserID
- EnterUserName
- LastChgDTM
- LastChgUserID
- Status_ID
- ValidationStatus_ID
- Version_ID

Members

For members, you cannot use **MDMMemberStatus** or **ROOT** for the **Code** attribute value.

See Also

Concepts
[Master Data Services Overview](#)

Automatic Code Creation (Master Data Services)

SQL Server 2012

In Master Data Services, numeric values can be automatically generated for the Code attribute, or for any other numeric attribute. When codes are generated automatically, you are not prevented from entering other values for codes; rather an initial value is automatically set.

Generating Code Values

Administrators can configure automatically-generated values for the Code attribute by editing the associated entity's properties. They can specify an initial value, and each subsequent value is increased by one.

When you enter Code values into MDS, either in one of the tools or by using the staging process, you can leave the Code value blank and a Code value will be automatically generated. Or you can specify a Code value of your choice.

Generating Other Attribute Values

Administrators can automatically generate values for attributes other than Code by creating business rules. They can specify an initial value, and specify the number each subsequent value is incremented by.

When you enter attribute values into MDS, either in one of the tools or by using the staging process, you can leave the attribute value blank. When business rules are applied, the values will be incremented based on the highest existing value. For example, if your rule is "Default attribute to a generated value that starts at 1 and increments by 4" and the highest current value for the attribute is 700, the value for the next member that's added will be 704.

Deleting Automatically Generated Values

After an administrator enables automatically generated values for the Code attribute, users may accidentally delete a member that had a Code value they want to reuse. The error message "The member code is already used by a member that was deleted" will be displayed. There are two possible solutions:

- In the **Version Management** functional area, an administrator can reverse the transaction that occurred when the member was deleted. However, this means that all of the former member's attributes and membership in hierarchies and collections is restored. For more information, see [Reverse a Transaction \(Master Data Services\)](#).
- An administrator can use the staging process to permanently delete the member. For more information, see [Deactivate or Delete Members by Using the Staging Process \(Master Data Services\)](#).

Related Tasks

Task Description	Topic
Automatically generate values for the Code attribute.	Automatically Generate Code Attribute Values (Master Data Services)
Automatically generate values for other attributes.	Automatically Generate Attribute Values Other Than Code (Master Data Services)

Related Content

- [Master Data Services Overview](#)
- [Business Rules \(Master Data Services\)](#)
- [Entities \(Master Data Services\)](#)

Automatically Generate Code Attribute Values (Master Data Services)

SQL Server 2012

In Master Data Services, automatically generate values for an entity's Code attribute when you want an integer to be automatically assigned to the Code value each time a new member is created.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- The entity must exist. For more information, see [Create an Entity \(Master Data Services\)](#).

To automatically generate Code values

1. In Master Data Manager, click **System Administration**.
2. On the **Model Explorer** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to generate codes for.
5. Click **Edit selected entity**.
6. Select the **Create Code values automatically** check box.
7. In the **Start with** box, type a number to begin incrementing. If members already exist, the Code will be set based on the highest existing value. For example, if the highest existing Code value is 299, the next member's Code value will be set to 300.
8. Click **Save entity**.

See Also

Tasks

[Automatically Generate Attribute Values Other Than Code \(Master Data Services\)](#)

Concepts

[Automatic Code Creation \(Master Data Services\)](#)

Automatically Generate Attribute Values Other Than Code (Master Data Services)

SQL Server 2012

In Master Data Services, automatically generate values for an entity's attribute values when you want an integer to be automatically assigned as the value each time business rules are applied.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- A numeric attribute must exist. For more information, see [Create a Numeric Attribute \(Master Data Services\)](#).

To automatically generate an attribute value

1. In Master Data Manager, click **System Administration**.
2. From the menu bar, point to **Manage** and click **Business Rules**.
3. On the **Business Rule Maintenance** page, from the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. From the **Member Type** list, select a type of member for the business rule to apply to.
6. From the **Attribute** list, leave the default of **All**.
7. Click **Add business rule**.
8. Click **Edit selected business rule**.
9. In the **Components** pane, expand the **Actions** node.
10. In the Default Value node, click **defaults to a generated value** and drag it to the **THEN** pane's **Actions** label.
11. In the **Attributes** pane, click the attribute with the value you want to generated and drag it to the **Edit Action** pane's **Select attribute** label.
12. Type a value in the **Start with** and **Increment by** boxes. If members already exist, the value will be set based on the highest existing value. For example, if the highest existing value is 299 and you set **Increment by** to 1, the next member's value will be set to 300.
13. In the **Edit Action** pane, click **Save item**.
14. Click **Back**.
15. Optionally, on the **Business Rules Maintenance** page, for the row that contains your business rule, double-click a cell in the **Name**, **Description**, or **Notification** column to update the value.
16. Click **Publish business rules**.
17. On the confirmation dialog box, click **OK**. The rule's status changes to **Active**.

Next Steps

- [Validate Specific Members against Business Rules \(Master Data Services\)](#)
- [Validate a Version against Business Rules \(Master Data Services\)](#)

See Also

Concepts

[Automatic Code Creation \(Master Data Services\)](#)

[Business Rules \(Master Data Services\)](#)

[Validation \(Master Data Services\)](#)

Business Rules (Master Data Services)

SQL Server 2012

In Master Data Services, a business rule is a rule that you use to ensure the quality and accuracy of your master data. You can use a business rule to automatically update data, to send email, or to start a business process or workflow.

■ Create and Publish Business Rules

Business rules are **If/Then** statements that you create in Master Data Manager. If an attribute value meets a specified condition, then an action is taken. Possible actions include setting a default value or changing a value. These actions can be combined with sending an email notification.

Business rules can be based on specific attribute values (for example, take action if Color=Blue), or when attribute values change (for example, take action if the value of the Color attribute changes). For more information about tracking non-specific changes, see [Change Tracking \(Master Data Services\)](#).

To use business rules you must first create and publish your rules, then apply the published rules to data. You can apply rules to subsets of data or to all data for a version by validating the version. A version cannot be committed until all attributes pass business rule validation.

If a user attempts to add an attribute value that doesn't pass business rule validation, the value can still be saved. You can review and correct validation issues, which are displayed in Master Data Manager.

When you create a model deployment package, if you want to include business rules you must include data from the version in the package.

If you create a business rule that uses the **OR** operator, you should create a separate rule for each conditional statement that can be evaluated independently. You can then exclude rules as needed, providing more flexibility and easier troubleshooting.

■ How Business Rules Are Applied

You can set priority order for rules to run in. However, before priority is taken into account, business rules are applied based on the type of action the rule takes. The order is as follows:

1. **Default Value**
2. **Change Value**
3. **Validation**
4. **External Action**

Within these groups, actions are applied in priority order, from lowest to highest. So for example, four separate rules might have **Default Value** actions. The **Default Value** action that occurs first depends on the priority order specified in the web UI.

Other important notes about applying rules:

- If a business rule is excluded or is not published with a status of **Active**, the rule is still available but is not included when business rules are applied.
- Business rules apply to the attribute values for all leaf or all consolidated members, not both.
- Business rules can be applied to any version of a model that is **Open** or **Locked**.
- Changes made to data when business rules are applied are not logged as transactions.
- A business rule cannot contain more than one **start workflow** action.

■ System Settings

There are two settings in Master Data Services Configuration Manager that affect business rules. You can adjust these settings in Master Data Services Configuration Manager or directly in the System Settings table. For more information, see [System Settings \(Master Data Services\)](#).

■ Related Tasks

Task Description	Topic
Create and publish a new business rule.	Create and Publish a Business Rule (Master Data Services)
Add multiple conditions to a business rule.	Add Multiple Conditions to a Business Rule (Master Data Services)
Create a business rule to require that attributes have values.	Require Attribute Values (Master Data Services)
Create a business rule to take an action based on changes to attribute values.	Initiate Actions Based on Attribute Value Changes (Master Data Services)
Change the name of an existing business rule.	Change a Business Rule Name (Master Data Services)

Configure Master Data Manager to send notifications when business rules are applied.	Configure Business Rules to Send Notifications (Master Data Services)
Apply business rules to specific members.	Validate Specific Members against Business Rules (Master Data Services)
Exclude a business rule so it is not used.	Exclude a Business Rule (Master Data Services)
Delete an existing business rule.	Delete a Business Rule (Master Data Services)

Related Content

- [Master Data Services Overview](#)
- [Versions \(Master Data Services\)](#)
- [Validation \(Master Data Services\)](#)
- [Change Tracking \(Master Data Services\)](#)

Create and Publish a Business Rule (Master Data Services)

SQL Server 2012

In Master Data Services, create a business rule to ensure the accuracy of your master data. After you create a rule, you must publish it before you can apply it to data.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To create and publish a business rule

1. In Master Data Manager, click **System Administration**.
2. From the menu bar, point to **Manage** and click **Business Rules**.
3. On the **Business Rule Maintenance** page, from the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. From the **Member Type** list, select a type of member for the business rule to apply to.
6. From the **Attribute** list, select an attribute or leave the default of **All**.
7. Click **Add business rule**.
8. Click **Edit selected business rule**.
9. In the **Components** pane, expand the **Conditions** node.
10. Click a condition and drag it to the **IF** pane's **Conditions** label.

Tip

You can delete items from your business rule by right-clicking and choosing **Delete**.

11. In the **Attributes** pane, click an attribute and drag it to the **Edit Condition** pane's **Select attribute** label.
12. In the **Edit Condition** pane, complete any required fields.
13. In the **Edit Condition** pane, click **Save item**.
14. In the **Components** pane, expand the **Actions** node.
15. Click an action and drag it to the **THEN** pane's **Action** label.
16. In the **Attributes** pane, click an attribute and drag it to the **Edit Action** pane's **Select attribute** label.
17. In the **Edit Action** pane, complete any required fields.
18. In the **Edit Action** pane, click **Save item**.
19. Optionally, add multiple conditions to the rule. For more information, see [Add Multiple Conditions to a Business Rule \(Master Data Services\)](#).
20. Click **Back**.
21. Optionally, on the **Business Rules Maintenance** page, for the row that contains your business rule, double-click a cell in the **Name**, **Description**, or **Notification** column to update the value.

Note

Notifications are sent only for rules that include a validation action.

22. Click **Publish business rules**.
23. On the confirmation dialog box, click **OK**. The rule's status changes to **Active**.

Next Steps

- Apply business rules to data by following one of these procedures:
 - [Validate Specific Members against Business Rules \(Master Data Services\)](#)
 - [Validate a Version against Business Rules \(Master Data Services\)](#)

▲ See Also

Tasks

[Configure Business Rules to Send Notifications \(Master Data Services\)](#)

[Change a Business Rule Name \(Master Data Services\)](#)

[Add Multiple Conditions to a Business Rule \(Master Data Services\)](#)

Add Multiple Conditions to a Business Rule (Master Data Services)

SQL Server 2012

In Master Data Services, add multiple **AND** or **OR** conditions to a business rule when you want a more complex rule.

Note

If you create a business rule that uses the **OR** operator, consider creating a separate rule for each conditional statement that can be evaluated independently. You can then exclude rules as needed, providing more flexibility and easier troubleshooting.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- A business rule must exist. For more information, see [Create and Publish a Business Rule \(Master Data Services\)](#).

To add multiple conditions to a business rule

1. In Master Data Manager, click **System Administration**.
2. From the menu bar, point to **Manage** and click **Business Rules**.
3. On the **Business Rule Maintenance** page, from the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. From the **Member Type** list, select a type of member.
6. From the **Attribute** list, select an attribute or leave the default of **All**.
7. Click the row for the business rule you want to edit.
8. Click **Edit selected business rule**.
9. In the **Components** pane, expand the **Logical Operators** node.
10. Click **AND** or **OR** and drag it to the **IF** pane's **AND** label.
11. In the **Components** pane, expand the **Conditions** node.
12. Click a condition and drag it to **IF** pane, to the **AND** or **OR** label from step 10.
13. In the **Attributes** pane, click an attribute and drag it to the **Edit Condition** pane's **Select attribute** label.
14. In the **Edit Condition** pane, complete any required fields.
15. In the **Edit Condition** pane, click **Save item**.
16. Optionally, to add more conditions, from the **Components** pane, drag **AND** or **OR** to any **AND** or **OR** in the **IF** pane. Then follow steps 13-15.

Tip

To delete a condition, click the name of the condition and in the **Edit Condition** pane, click **Delete item**.

See Also

Tasks

[Change a Business Rule Name \(Master Data Services\)](#)

[Configure Business Rules to Send Notifications \(Master Data Services\)](#)

Concepts

[Business Rules \(Master Data Services\)](#)

Require Attribute Values (Master Data Services)

SQL Server 2012

In Master Data Services, require attribute values when you want to ensure your master data is complete.

Note

Members that are missing domain-based attribute values are not displayed in derived hierarchies that are based on those relationships.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To require attribute values

1. In Master Data Manager, click **System Administration**.
2. From the menu bar, point to **Manage** and click **Business Rules**.
3. On the **Business Rule Maintenance** page, from the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. From the **Member Type** list, select a type of member for the business rule to apply to.
6. From the **Attribute** list, select an attribute or leave the default of **All**.
7. Click **Add business rule**.
8. Click **Edit selected business rule**.
9. In the **Components** pane, expand the **Actions** node.
10. Click **is required** and drag it to the **THEN** pane's **Action** label.
11. In the **Attributes** pane, click an attribute and drag it to the **Edit Action** pane's **Select attribute** label.
12. In the **Edit Action** pane, click **Save item**.
13. Click **Back**.
14. Optionally, on the **Business Rules Maintenance** page, for the row that contains your business rule, double-click a cell in the **Name**, **Description**, or **Notification** column to update the value.
15. Click **Publish business rules**.
16. On the confirmation dialog box, click **OK**. The rule's status changes to **Active**.

Next Steps

- Apply business rules to data by following one of these procedures:
 - [Validate Specific Members against Business Rules \(Master Data Services\)](#)
 - [Validate a Version against Business Rules \(Master Data Services\)](#)

See Also

Concepts

[Business Rules \(Master Data Services\)](#)

[Derived Hierarchies \(Master Data Services\)](#)

Change a Business Rule Name (Master Data Services)

SQL Server 2012

In Master Data Services, change a business rule name when the name assigned does not meet your business needs.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- A business rule must exist. For more information, see [Create and Publish a Business Rule \(Master Data Services\)](#).

To change the name of a business rule

1. In Master Data Manager, click **System Administration**.
2. From the menu bar, point to **Manage** and click **Business Rules**.
3. On the **Business Rule Maintenance** page, from the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. From the **Member Type** list, select a type of member.
6. From the **Attribute** list, select an attribute or leave the default of **All**.
7. In the grid, in the row for the business rule, double-click the **Name** field.
8. Type a name for the business rule.
9. Press ENTER.
10. Click **Publish business rules**.
11. On the confirmation dialog box, click **OK**. The rule's status changes to **Active**.

See Also

Concepts

[Business Rules \(Master Data Services\)](#)

Configure Business Rules to Send Notifications (Master Data Services)

SQL Server 2012

In Master Data Services, configure business rules to send notifications when you want to notify users about attribute value changes.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** and **User and Group Permissions** functional areas. If you do not have permission to the **User and Group Permissions** functional area, you cannot view the list of users and groups to send notifications to.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- A business rule that uses a validation action must already exist. For more information, see [Create and Publish a Business Rule \(Master Data Services\)](#).
- The user or group that receives the notification must have at least **Read-only** permission to the attribute that fails validation. Users or groups that are explicitly or implicitly denied permission to the attribute will receive the email but will not be able to access the attribute in Master Data Manager.
- If mail is sent to a group, only members of the group that have accessed Master Data Manager will get the email.

To configure business rules to send notifications

1. In Master Data Manager, click **System Administration**.
2. From the menu bar, point to **Manage** and click **Business Rules**.
3. On the **Business Rule Maintenance** page, from the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. From the **Member Type** list, select a type of member.
6. From the **Attribute** list, select an attribute or leave the default of **All**.
7. In the grid, in the row for the business rule, double-click the **Notification** field.
8. From the sub-menu, click a user or group to send the email notification to.

Next Steps

- Apply business rules to data by following one of these procedures:
 - [Validate Specific Members against Business Rules \(Master Data Services\)](#)
 - [Validate a Version against Business Rules \(Master Data Services\)](#)
- Configure the email protocol as follows:
 - [Configure Email Notifications \(Master Data Services\)](#)

See Also

Tasks

[Configure Email Notifications \(Master Data Services\)](#)

Concepts

[Notifications \(Master Data Services\)](#)

Validate Specific Members against Business Rules (Master Data Services)

SQL Server 2012

In Master Data Services, apply business rules selectively when you want to update or validate subsets of members against business rules.

Note

If you want to apply business rules to all members in a version of a model, see [Validate a Version against Business Rules \(Master Data Services\)](#).

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.
- You must have a minimum of **Update** permission to the model object you are applying business rules to.

To apply business rules selectively

1. On the Master Data Manager home page, from the **Model** list, select a model.
2. From the **Version** list, select a version.
3. Click **Explorer**.
4. From the menu bar, point to **Entities** and click the name of the entity that contains members you want to apply rules to.
5. Click **Apply business rules**. Business rules are applied only to the members displayed in the grid.

See Also

Tasks

[Validate a Version against Business Rules \(Master Data Services\)](#)

Concepts

[Business Rules \(Master Data Services\)](#)

Exclude a Business Rule (Master Data Services)

SQL Server 2012

In Master Data Services, exclude a business rule when you do not want to delete the rule permanently, but you do not want to validate data against it.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To exclude a business rule

1. In Master Data Manager, click **System Administration**.
2. From the menu bar, point to **Manage** and click **Business Rules**.
3. On the **Business Rule Maintenance** page, from the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. From the **Member Type** list, select a type of member.
6. From the **Attribute** list, select an attribute or leave the default of **All**.
7. In the grid, in the row for the business rule, select the check box in the **Exclude** column. The value in the **Status** column is **Exclusion pending**.
8. Click **Publish business rules**.
9. In the confirmation dialog box, click **OK**. The value in the **Status** column is **Excluded**.

See Also

Tasks

[Delete a Business Rule \(Master Data Services\)](#)

[Create and Publish a Business Rule \(Master Data Services\)](#)

Concepts

[Business Rules \(Master Data Services\)](#)

Delete a Business Rule (Master Data Services)

SQL Server 2012

In Master Data Services, delete a business rule when it is no longer needed.

Note

You can prevent data from being validated against a business rule by excluding it, rather than deleting it. For more information, see [Exclude a Business Rule \(Master Data Services\)](#).

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete a business rule

1. In Master Data Manager, click **System Administration**.
2. From the menu bar, point to **Manage** and click **Business Rules**.
3. On the **Business Rule Maintenance** page, from the **Model** list, select a model.
4. From the **Entity** list, select an entity.
5. From the **Member Type** list, select a type of member for the business rule to apply to.
6. From the **Attribute** list, select an attribute or leave the default of **All**.
7. In the grid, click the row for the business rule you want to delete.
8. Click **Delete selected business rule**.
9. In the confirmation dialog box, click **OK**. The value in the **Status** column is **Deletion pending**.
10. Click **Publish business rules**.
11. In the confirmation dialog box, click **OK**.

See Also

Tasks

[Exclude a Business Rule \(Master Data Services\)](#)

[Create and Publish a Business Rule \(Master Data Services\)](#)

Concepts

[Business Rules \(Master Data Services\)](#)

Business Rule Conditions (Master Data Services)

SQL Server 2012

In Master Data Services, business rule conditions determine the conditions that must be true for one or more actions to be taken.

Note

Conditions are optional. If you do not specify a condition, then actions are taken any time data is validated against business rules.

Business Rule Conditions

Condition Name	Description
is equal to	The selected attribute is equal to a specific attribute, a specific attribute value, or is blank. This condition is valid for text, number, date, and link values.
is not equal to	The selected attribute is not equal to a specific attribute, a specific attribute value, or is blank. This condition is valid for text, number, date, and link values.
is greater than	The selected attribute is greater than a specific attribute, a specific attribute value, or is blank. This condition is valid for text, number, and date values.
is greater than or equal to	The selected attribute is greater than or equal to a specific attribute, a specific attribute value, or is blank. This condition is valid for text, number, and date values.
is less than	The selected attribute is less than a specific attribute, a specific attribute value, or is blank. This condition is valid for text, number, and date values.
is less than or equal to	The selected attribute is less than or equal to a specific attribute, a specific attribute value, or is blank. This condition is valid for text, number, and date values.
starts with	The selected attribute starts with a specific attribute, a specific attribute value, or is blank. This condition is valid for text and link values.
ends with	The selected attribute ends with a specific attribute, a specific attribute value, or is blank. This condition is valid for text and link values.
contains	The selected attribute contains a specific attribute, a specific attribute value, or is blank. This condition is valid for text and link values.
contains the pattern	The selected attribute contains the pattern of a specific attribute, a specific attribute value, or is blank. Use .NET Framework regular expressions to specify the pattern. For more information about regular expressions, see Regular Expression Language Elements in the MSDN Library. This condition is valid for text and link values.
contains the subset	The selected attribute contains the subset of a specific attribute or a specific attribute value. You must specify the starting position for the search (for example, 1 means start searching at the first character). This condition is valid for text and link values.
has changed	The selected attribute has changed since the last time business rules were applied to the member. You must specify the change group the attribute is a member of. For more information on change tracking groups, see Add Attributes to a Change Tracking Group (Master Data Services) . This condition is valid for text, number, date, and link values.
is between	The selected attribute is between two specific attribute values. This condition is valid for text, number, and date values.

Note

When a business rule contains a condition that compares two values, and the rule is applied to a member for which both values are NULL, that member will fail validation.

See Also

Tasks

[Create and Publish a Business Rule \(Master Data Services\)](#)

Concepts

[Business Rule Actions \(Master Data Services\)](#)

[Business Rules \(Master Data Services\)](#)

Business Rule Actions (Master Data Services)

SQL Server 2012

In Master Data Services, business rule actions are the consequence of business rule condition evaluations. If a condition is true, the action is initiated.

Note

For default value and change value actions, if the generated value exceeds the maximum length of the attribute, the generated value is truncated.

Default Value Actions

Default value actions set the default value of a specified attribute. Users with permission can change these default values.

Value Name	Description
defaults to	The selected attribute defaults to a specific attribute, a specific attribute value, or is blank. This action is valid for text, number, date, and link values.
defaults to a generated value	The selected attribute defaults to a generated value that is determined by entering a starting and incremental value. This action is valid for text and number values.
defaults to a concatenated value	The selected attribute defaults to a concatenated value that is determined by specifying multiple attributes. This action is valid for text and link values.

Change Value Actions

Change value actions update the value of a specified attribute or attribute value. Users can change these values only if the new value causes the action to be true.

Value Name	Description
Equals	The selected attribute is changed to a defined attribute value, another attribute, or blank. This action is valid for text, number, date, and link values.
equals a concatenated value	The selected attribute is changed to a concatenated value, which is determined by specifying multiple attributes. This action is valid for text and link values.

Validation Actions

Validation actions, when they do not evaluate to true, send email to a specified user or group. To commit a version, all validation actions must evaluate to true.

The only exceptions are the **is mandatory** and **is not valid** actions. These actions must be combined with a change value action, so that the data can successfully be validated and the version committed.

Validation Name	Description
is required	The selected attribute is required , which means it cannot be null or blank. This action is valid for text, number, date, and link values.
is not valid	The selected attribute is not valid . This action is valid for text, number, date, and link values.
must contain the pattern	The selected attribute must contain the pattern that is specified. Use .NET Framework regular expressions to specify the pattern. For more information about regular expressions, see Regular Expression Language Elements in the MSDN Library. This action is valid for text and link values.
must be unique	The selected attribute must be unique independently or in combination with defined attributes. Best Practice: Combine this action with a mandatory condition to ensure validity of index fields in subscribing systems. This action is valid for text, number, date, and link values.
must have one of the following values	The selected attribute must have one of the values specified in a list. This action is valid for text values.
must be greater than	The selected attribute must be greater than a specific attribute, a specific attribute value, or blank. This action is valid for text, number, and date values.
must be equal to	The selected attribute must be equal to a defined attribute value, another attribute, or blank. This action is valid for text, number, date, and link values.
must be greater than or equal to	The selected attribute must be greater than or equal to a specific attribute, a specific attribute value, or blank. This action is valid for text, number, and date values.

must be less than	The selected attribute must be less than a specific attribute, a specific attribute value, or blank. This action is valid for text, number, and date values.
must be less than or equal to	The selected attribute must be less than or equal to a specific attribute, a specific attribute value, or blank. This action is valid for text, number, and date values.
must be between	The selected attribute must be between two specific attribute or attribute values. This action is valid for text, number, and date values.
must have a minimum length of	The selected attribute must have a minimum length of the specified value. This action is valid for text and link values.
must have a maximum length of	The selected attribute must have a maximum length of the specified value. This action is valid for text and link values.

External Action

External actions interact with applications outside of Master Data Services.

Action Name	Description
start workflow	Initiates an external workflow. The data that caused this action to occur is passed to the workflow. For more information, see SharePoint Workflow Integration with Master Data Services . This action is valid for text, number, date, and link values.

See Also

Tasks

[Create and Publish a Business Rule \(Master Data Services\)](#)

Concepts

[Business Rule Conditions \(Master Data Services\)](#)

[Business Rules \(Master Data Services\)](#)


Change Tracking (Master Data Services)

SQL Server 2012

In Master Data Services, you can use change tracking groups to take action when an attribute value changes. Use change tracking when you don't know what the new value will be, but instead want to know if any change occurred.

▲ Configuring Change Tracking

To configure change tracking, you add an attribute to a change tracking group. The group can contain one or many attributes. Then, you create a business rule to define the action that is taken when any of the attributes in the group change.

 Note
Change tracking business rules consider staged (imported) data to be changed.

▲ Related Tasks

Task Description	Topic
Add attributes to a change tracking group.	Add Attributes to a Change Tracking Group (Master Data Services)
Create a business rule that initiates actions based on attribute changes.	Initiate Actions Based on Attribute Value Changes (Master Data Services)

▲ Related Content

- [Validation \(Master Data Services\)](#)
- [Business Rules \(Master Data Services\)](#)
- [Attributes \(Master Data Services\)](#)

Add Attributes to a Change Tracking Group (Master Data Services)

SQL Server 2012

In Master Data Services, add attributes to a change tracking group when you want to track changes to the attribute's values.

Note

After you add an attribute to a change tracking group, when values for the attribute change, the attribute is flagged as changed in the Master Data Services database. Create a business rule to take action based on the change.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- Attributes must exist to add to the change tracking group. For more information, see [Create a Text Attribute \(Master Data Services\)](#).

To add attributes to a change tracking group

1. In Master Data Manager, click **System Administration**.
2. On the **Model Explorer** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to track attribute values for.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page:
 - If the attribute is for leaf members, in the **Leaf attributes** pane, select the attribute and click **Edit leaf attribute**.
 - If the attribute is for consolidated members, in the **Consolidated attributes** pane, select the attribute and click **Edit consolidated attribute**.
 - If the attribute is for collections, in the **Collection attributes** pane, select the attribute and click **Edit collection attribute**.
7. Select the **Enable change tracking** check box.
8. In the **Change tracking group** box, type a number for the group.
9. Click **Save attribute**.
10. On the **Entity Maintenance** page, click **Save entity**.
11. Repeat this procedure for all attributes you want to include in the group. Use the same change tracking group number for each attribute in the group.

Next Steps

- [Initiate Actions Based on Attribute Value Changes \(Master Data Services\)](#)

See Also

Tasks

[Create a Text Attribute \(Master Data Services\)](#)

[Create a Domain-Based Attribute \(Master Data Services\)](#)

Initiate Actions Based on Attribute Value Changes (Master Data Services)

SQL Server 2012


In Master Data Services, create a business rule to initiate actions based on changes to attribute values. For example, when a specific attribute value changes, you may want to change a value, send a notification, or start an external workflow.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- Your attributes must be in a change tracking group. See [Add Attributes to a Change Tracking Group \(Master Data Services\)](#) for more information.

To create a business rule to initiate actions based on attribute value changes

1. In Master Data Manager, click **System Administration**.
 2. From the menu bar, point to **Manage** and click **Business Rules**.
 3. On the **Business Rule Maintenance** page, from the **Model** list, select a model.
 4. From the **Entity** list, select an entity.
 5. From the **Member Type** list, select a type of member for the business rule to apply to.
 6. From the **Attribute** list, select an attribute or leave the default of **All**.
 7. Click **Add business rule**.
 8. Click **Edit selected business rule**.
 9. In the **Components** pane, expand the **Conditions** node.
 10. Under the **Value comparison** node, drag **has changed** to the **IF** pane's **Conditions** label.
 11. In the **Attributes** pane, click an attribute and drag it to the **Edit Condition** pane's **Select attribute** label. This attribute has no effect on the rule, so select any available attribute.
 12. In the **Edit Condition** pane, in the **Change tracking group** box, type the number of the change tracking group that you assigned as part of the prerequisites.
 13. In the **Edit Condition** pane, click **Save item**.
 14. In the **Components** pane, expand the **Actions** node.
 15. Click an action and drag it to the **THEN** pane's **Action** label.
 16. In the **Attributes** pane, click an attribute and drag it to the **Edit Action** pane's **Select attribute** label.
 17. In the **Edit Action** pane, complete any required fields.
 18. In the **Edit Action** pane, click **Save item**.
 19. Click **Back**.
 20. Optionally, on the **Business Rules Maintenance** page, for the row that contains your business rule, double-click a cell in the **Name**, **Description**, or **Notification** column to update the value.
-  **Note**

Notifications are sent only for rules that include a validation action.
21. Click **Publish business rules**.
 22. On the confirmation dialog box, click **OK**. The rule's status changes to **Active**.

Next Steps

- Apply business rules to data by following one of these procedures:
 - [Validate Specific Members against Business Rules \(Master Data Services\)](#)
 - [Validate a Version against Business Rules \(Master Data Services\)](#)

▴ See Also

Tasks

[Add Attributes to a Change Tracking Group \(Master Data Services\)](#)

Concepts

[Business Rules \(Master Data Services\)](#)

Notifications (Master Data Services)

SQL Server 2012

Master Data Services can be configured to send an email notification when business rule validation fails or the status of a model version changes.

How Notifications Are Sent

You configure notifications in Master Data Services Configuration Manager. Notifications send email messages by using Database Mail on the instance of SQL Server Database Engine that hosts the Master Data Services database. For more information about Database Mail, see [Database Mail Configuration Objects](#) in SQL Server Books Online.

When Notifications Are Sent

After notifications are configured, automated email notifications can be sent in the following instances.

Instance	Description
Data fails business rule validation	Individual business rules must be configured to send email when an attribute value fails business rule validation. For more information, see Configure Business Rules to Send Notifications (Master Data Services) .
Model version status changes	Each time a model version's status changes, users that are model administrators receive notifications automatically. For more information, see Administrators (Master Data Services) .

System Settings

There are settings in Master Data Services Configuration Manager that affect notifications. You can adjust these settings in Master Data Services Configuration Manager or directly in the System Settings table in the Master Data Services database. For more information, see [System Settings \(Master Data Services\)](#).

Related Tasks

Task Description	Topic
Configure Master Data Services to send email notifications.	Configure Email Notifications (Master Data Services)
Configure Master Data Manager to send notifications when attribute values change.	Configure Business Rules to Send Notifications (Master Data Services)

Related Content

- [Business Rules \(Master Data Services\)](#)
- [Versions \(Master Data Services\)](#)
- [Troubleshooting Email Notifications \(Master Data Services\)](#)

Configure Email Notifications (Master Data Services)

SQL Server 2012

Configure notification emails when you want Master Data Services to send email messages automatically.

To configure notifications

1. In Master Data Services Configuration Manager, on the **Database** page, select your Master Data Services database.
2. In the **System Settings** section, click **Create Profile**.
3. Complete all required fields. For more information, see [Create Database Mail Profile and Account Dialog Box \(Master Data Services Configuration Manager\)](#).
4. Click **OK**.

Note

After you configure notifications, you cannot use Master Data Services Configuration Manager to make changes. You must make changes directly in the Master Data Services database. For more information, see [Database Mail Configuration Objects](#).

Next Steps

- There are settings in Master Data Services Configuration Manager that affect notifications. You can adjust these settings in Master Data Services Configuration Manager or directly in the System Settings table in the Master Data Services database. For more information, see [System Settings \(Master Data Services\)](#).

See Also

Tasks

[Configure Business Rules to Send Notifications \(Master Data Services\)](#)

Concepts

[Notifications \(Master Data Services\)](#)

Other Resources

[Troubleshooting Email Notifications \(Master Data Services\)](#)

Transactions (Master Data Services)

SQL Server 2012

In Master Data Services, a transaction is recorded each time action is taken on a member. Transactions can be viewed by all users and reversed by administrators. Transactions show the date, time, and user who took the action, along with other details. Users can add an annotation to a transaction, to indicate why a transaction took place.

When Transaction Are Recorded

Transactions are recorded when members:

- Are created, deleted, or reactivated.
- Have attribute values changed.
- Are moved in a hierarchy.

Transactions are not recorded when business rules change attribute values.

View and Manage Transactions

In the **Explorer** functional area, you can view and annotate (add comments to) the transactions that you made yourself.

In the **Version Management** functional area, administrators can view all transactions for all users for the models they have access to, and reverse any of these transactions.

System Settings

There is a setting in Master Data Services Configuration Manager that affects whether or not transactions are recorded when records are staged. This setting affects SQL Server 2008 R2 only. You can adjust this setting in Master Data Services Configuration Manager or directly in the System Settings table in the Master Data Services database. For more information, see [System Settings \(Master Data Services\)](#).

When importing data in this version of SQL Server, you can specify whether or not to log transactions when initiating the stored procedure. For more information, see [Staging Stored Procedure \(Master Data Services\)](#).

Concurrency

If a particular entity value is shown simultaneously in more than one Explorer session, concurrent edits to the same value are possible. Concurrent edits will not be detected automatically by MDS. This can occur when multiple users use the MDS Explorer in the Web browser from multiple sessions, for example from multiple computers, multiple browser tabs or windows, or multiple user accounts.

More than one user can update the same entity values without error despite transactions being enabled. Typically the last edit to the value in a sequence of time will take precedence. The duplicate edit conflict can be manually observed in the transaction history and can be reversed manually by the administrator. The transaction history will show the individual transactions for the **Prior value** and **New value** for the attribute in question from each session, but will not automatically resolve the conflict when multiple **New Values** exist for the same old value.

Related Tasks

Task Description	Topic
Undo an action by reversing a transaction (administrators only).	Reverse a Transaction (Master Data Services)

Related Content

- [Administrators \(Master Data Services\)](#)
- [Annotations \(Master Data Services\)](#)

Reverse a Transaction (Master Data Services)

SQL Server 2012

In Master Data Services, administrators can reverse a transaction when an action needs to be undone. Examples of transactions are attribute value changes, hierarchy moves, or member deletions.

Prerequisites

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To reverse a transaction

1. On the Master Data Manager home page, click **Version Management**.
2. On the menu bar, click **Transactions**.
3. On the **Transactions** page, from the **Model** list, select a model.
4. From the **Version** list, select a version.
5. Click the row in the grid for the transaction you want to reverse.
6. Click **Reverse Transaction**.
7. In the confirmation dialog box, click **OK**. Another transaction is added to the grid to record the reversed transaction.

See Also

Tasks

[Reactivate a Member or Collection \(Master Data Services\)](#)

Concepts

[Transactions \(Master Data Services\)](#)

Annotations (Master Data Services)

SQL Server 2012

In Master Data Services, annotations are comments that you enter to provide details about transactions.

You can annotate a transaction to provide information about why an action was taken. You must select a transaction to add a related annotation. Annotations are displayed along with transactions in:

- The **Explorer** functional area of the Master Data Manager web application.
- The **Version Management** functional area of the Master Data Manager web application.
- The MDS Add-in for Excel.

Related Tasks

Task Description	Topic
Add an annotation to a transaction.	Annotate a Transaction (Master Data Services)

Related Content

- [Transactions \(Master Data Services\)](#)

Annotate a Transaction (Master Data Services)

SQL Server 2012

In Master Data Services, annotate a transaction when you want to provide supporting details about the transaction for historical purposes.

 Note
You cannot delete annotations.

Prerequisites

- To annotate transactions that you created, you must have permission to access the **Explorer** functional area, and have a minimum of **Update** permission to the model object you want to annotate.
- To annotate transactions for all users, you must have permission to access the **Version Management** functional area and be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To annotate a transaction in Explorer

1. On the Master Data Manager home page, from the **Model** list, select a model.
2. From the **Version** list, select a version.
3. Click **Explorer**.
4. From the menu bar, point to **Entities** and click the entity that contains the member with a transaction you want to annotate.
5. In the grid, click the row of the member.
6. Click **View Transactions**.
7. In the **View Transactions** dialog box, click the transaction you want to annotate.
8. In the box at the bottom of the dialog box, type your annotation.
9. Click **Add Annotation**. The annotation is displayed in the **Annotations** pane.

To annotate a transaction in Version Management (administrators only)

1. On the Master Data Manager home page, click **Version Management**.
2. On the menu bar, click **Transactions**.
3. In the **Transactions** pane, click the row in the grid for the transaction you want to annotate.
4. In the **Transaction Annotations** pane, in the **Annotation** box, type your annotation.
5. Click **OK**.

See Also

Concepts

[Annotations \(Master Data Services\)](#)

[Transactions \(Master Data Services\)](#)

Hierarchies (Master Data Services)

SQL Server 2012

In Master Data Services, a hierarchy is a tree structure that you can use to:

- Group similar members for organizational purposes.
- Consolidate and summarize members for reporting and analysis.

What Hierarchies Contain

Each hierarchy contains members from one or more entities. When a member is added, changed, or deleted, all hierarchies are updated. This ensures that the data is accurate in all hierarchies. Hierarchies also help ensure that each member is counted once and only once.

If you want to create a grouping of a subset of members, consider using a collection. For more information, see [Collections \(Master Data Services\)](#).

Kinds of Hierarchies

You can create multiple hierarchies to view and organize your members in different ways. You can create:

- Ragged hierarchies from a single entity, which are called explicit hierarchies. For more information, see [Explicit Hierarchies \(Master Data Services\)](#).
- Level-based hierarchies from multiple entities, based on the existing relationships between entities and their attributes, which are called derived hierarchies. For more information, see [Derived Hierarchies \(Master Data Services\)](#).

 **Note**

All members in a hierarchy must be within the same model.

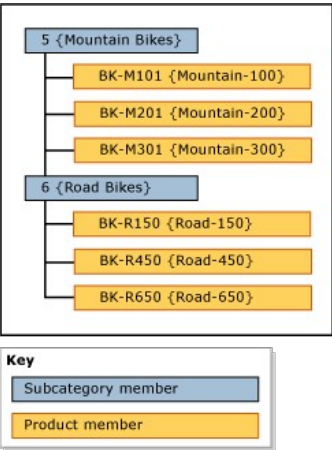
Hierarchies Are Not Taxonomies

A hierarchy is different from a taxonomy. A taxonomy organizes members by multiple attributes at the same time, while a hierarchy organizes members by one attribute at a time. A taxonomy can include the same member multiple times, while a hierarchy includes a member only once.

For example, the same bike can be included in a taxonomy twice: once because it's red, and once because it's a size 38. In a hierarchy, the bike is included only once, so you must decide whether to show it in relation to its color or its size.

Hierarchy Example

In the following example, product members are grouped by subcategory members.



Related Tasks

Task Description	Topic
Enable an entity for explicit hierarchies and collections.	Enable an Entity for Explicit Hierarchies and Collections (Master Data Services)
Create a explicit hierarchy.	Create an Explicit Hierarchy (Master Data Services)
Create a derived hierarchy.	Create a Derived Hierarchy (Master Data Services)

Hide or delete levels in an existing derived hierarchy.	Hide or Delete Levels in a Derived Hierarchy (Master Data Services)
---	---

Related Content

- [Explicit Hierarchies \(Master Data Services\)](#)
- [Derived Hierarchies \(Master Data Services\)](#)
- [Recursive Hierarchies \(Master Data Services\)](#)
- [Derived Hierarchies with Explicit Caps \(Master Data Services\)](#)
- [Collections \(Master Data Services\)](#)

Explicit Hierarchies (Master Data Services)

SQL Server 2012

In Master Data Services, an explicit hierarchy organizes members from a single entity in any way you specify. The structure can be ragged and unlike derived hierarchies, explicit hierarchies are not based on domain-based attribute relationships.

Consolidated Members Group Other Members

An explicit hierarchy uses consolidated members that you create for the purpose of grouping other members. These consolidated members can belong to only one explicit hierarchy at a time. An explicit hierarchy also includes all of the leaf members from the associated entity.

An explicit hierarchy can be ragged, which means that the hierarchy can end at different levels simultaneously. Each consolidated member can have an unlimited number of consolidated and leaf members underneath, or can have none. The leaf members can be under a single consolidated member or under multiple levels of consolidated members.

Note

Before you can create an explicit hierarchy, the entity must be enabled for explicit hierarchies. For more information, see [Enable an Entity for Explicit Hierarchies and Collections \(Master Data Services\)](#).

Types of Explicit Hierarchies

There are two types of explicit hierarchies: mandatory and non-mandatory.

Mandatory Explicit Hierarchy

A mandatory explicit hierarchy is a hierarchy in which all leaf members must be included in the hierarchy tree. By default, all members are included at the root of the tree. You can rearrange the members as needed.

Non-Mandatory Explicit Hierarchy

A non-mandatory explicit hierarchy is a hierarchy in which all leaf members are in a system-created **Unused** node. You can move members out of this node as you need them. The rest of the members can remain in the **Unused** node.

When you use non-mandatory explicit hierarchies, any reporting or analysis done on the hierarchy may not match reporting or analysis done on mandatory hierarchies.

Rules

The following rules apply to explicit hierarchies (both mandatory and non-mandatory).

- Each leaf member can be included in the hierarchy only once.
- All consolidated members must be included in a hierarchy.
- Consolidated members cannot be in more than one explicit hierarchy.
- Consolidated members in the hierarchy tree do not have to contain leaf members underneath them.
- If you delete an explicit hierarchy, all consolidated members that were used in the hierarchy are deleted.
- If you delete a consolidated member that was in an explicit hierarchy, all leaf members that were grouped by that consolidated member are moved to the root.

Explicit Hierarchies versus Derived Hierarchies

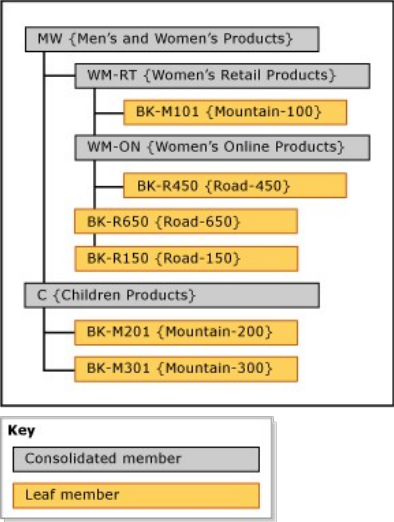
The following table shows some of the differences between explicit and derived hierarchies.

Explicit Hierarchies	Derived Hierarchies
Structure is defined by the user	Structure is derived from the relationships between domain-based attributes
Contains members from a single entity	Contains members from multiple entities
Uses consolidated members to group other members	Uses leaf members from one entity to group leaf members from another entity
Can be ragged	Always contains a consistent number of levels

Explicit Hierarchy Example

In the following example, the Product entity contains these leaf members: BK-M101 {Mountain-100}, BK-M201 {Mountain-200}, BK-M301 {Mountain-300}, BK-R150 {Road-150}, BK-R450 {Road-450}, and BK-R650 {Road-650}.

To summarize these leaf members at specific consolidation points, you can create consolidated members in the Product entity. Insert the consolidated members at levels in the hierarchy tree where you want to summarize the leaf members. There is no limitation on where you insert your consolidated members; however, each member (leaf or consolidated) can be used only once.



Consolidated members can be used to group members at any level, and leaf and consolidated members are sorted in the order you determine.

Related Tasks

Task Description	Topic
Enable an entity for explicit hierarchies and collections.	Enable an Entity for Explicit Hierarchies and Collections (Master Data Services)
Create a new explicit hierarchy.	Create an Explicit Hierarchy (Master Data Services)
Change the name of an existing explicit hierarchy.	Change an Explicit Hierarchy Name (Master Data Services)
Delete an existing explicit hierarchy.	Delete an Explicit Hierarchy (Master Data Services)

Related Content

- [Derived Hierarchies \(Master Data Services\)](#)
- [Collections \(Master Data Services\)](#)

Create an Explicit Hierarchy (Master Data Services)

SQL Server 2012

In Master Data Services, create an explicit hierarchy when you need a ragged hierarchy in which members can exist at any level. Explicit hierarchies contain members from a single entity.

After you create an explicit hierarchy, you can add members to it in the **Explorer** functional area.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- The entity must be enabled for explicit hierarchies and collections. For more information, see [Enable an Entity for Explicit Hierarchies and Collections \(Master Data Services\)](#).

To create an explicit hierarchy

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that you want to create an explicit hierarchy for.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page, in the **Explicit hierarchies** pane, click **Add explicit hierarchy**.
7. On the **Add Explicit Hierarchy** page, in the **Explicit hierarchy name** box, type a name for the hierarchy.
8. Optionally, clear the **Mandatory hierarchy** check box to create the hierarchy as a non-mandatory hierarchy. For more information about hierarchy types, see [Explicit Hierarchies \(Master Data Services\)](#).
9. Click **Save hierarchy**.
10. On the **Edit Entity** page, click **Save entity**.

Next Steps

- [Create a Consolidated Member \(Master Data Services\)](#)
- [Move Members within a Hierarchy \(Master Data Services\)](#)

See Also

Tasks

[Change an Explicit Hierarchy Name \(Master Data Services\)](#)

Concepts

[Explicit Hierarchies \(Master Data Services\)](#)

[Derived Hierarchies with Explicit Caps \(Master Data Services\)](#)

Change an Explicit Hierarchy Name (Master Data Services)

SQL Server 2012

In Master Data Services, you can change the name of an explicit hierarchy.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To change the name of an explicit hierarchy

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that contains the explicit hierarchy you want to rename.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page, click the explicit hierarchy you want to rename.
7. Click **Edit selected hierarchy**.
8. In the **Explicit hierarchy name** box, type the updated name of the hierarchy.
9. Click **Save hierarchy**.

See Also

Tasks

[Create an Explicit Hierarchy \(Master Data Services\)](#)

[Delete an Explicit Hierarchy \(Master Data Services\)](#)

Concepts

[Explicit Hierarchies \(Master Data Services\)](#)

Delete an Explicit Hierarchy (Master Data Services)

SQL Server 2012

In Master Data Services, delete an explicit hierarchy when you no longer need it.

Caution

When you delete an explicit hierarchy, all consolidated members in the hierarchy are deleted also. If you delete all explicit hierarchies for an entity, all collections for the entity are also deleted and the entity is no longer enabled for explicit hierarchies and collections.

▀ Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete an explicit hierarchy

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Entities**.
3. On the **Entity Maintenance** page, from the **Model** list, select a model.
4. Select the row for the entity that contains the explicit hierarchy you want to delete.
5. Click **Edit selected entity**.
6. On the **Edit Entity** page, in the **Explicit Hierarchies** pane, click the explicit hierarchy you want to delete.
7. Click **Delete selected hierarchy**.
8. In the confirmation dialog box, click **OK**.
9. In the additional confirmation dialog box, click **OK**.

▀ See Also

Tasks

[Create an Explicit Hierarchy \(Master Data Services\)](#)

Concepts

[Explicit Hierarchies \(Master Data Services\)](#)

Derived Hierarchies (Master Data Services)

SQL Server 2012

A Master Data Services derived hierarchy is derived from the domain-based attribute relationships that already exist between entities in a model.

You can create a derived hierarchy to highlight any of the existing domain-based attribute relationships in the model.

Leaf Members Group Other Leaf Members

In a derived hierarchy, the leaf members from one entity are used to group the leaf members of another entity. A derived hierarchy is based on the relationship between these entities. An explicit hierarchy, in contrast, is based on members from a single entity only and is structured in any way you specify.

You can change the structure of a derived hierarchy without affecting the underlying data. As long as the relationships still exist in the model, deleting a derived hierarchy has no effect on your master data.

Explicit Hierarchies versus Derived Hierarchies

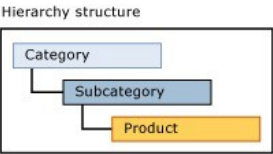
The following table shows some of the differences between explicit and derived hierarchies.

Explicit Hierarchies	Derived Hierarchies
Structure is defined by the user	Structure is derived from the relationships between domain-based attributes
Contains members from a single entity	Contains members from multiple entities
Uses consolidated members to group other members	Uses leaf members from one entity to group leaf members from another entity
Can be ragged	Always contains a consistent number of levels

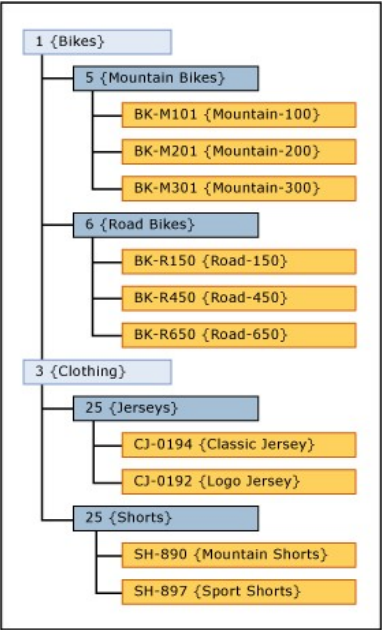
Derived Hierarchy Example

In the following example, leaf members of the Product entity are grouped by leaf members of the Subcategory entity, which are then grouped by leaf members of the Category entity. This hierarchy is possible because the Product entity has a domain-based attribute named Subcategory, and the Subcategory entity has a domain-based attribute named Category.

The hierarchy structure shows how the members are grouped. The entity with the most members is at the bottom.



In a derived hierarchy, you can highlight the relationship between Product and Subcategory, and then between Subcategory and Category. When you view the members in this hierarchy, each level in the tree contains members from the same entity.



This type of hierarchy prevents you from moving a member to a level that is not valid. For example, you can move the Road-650 bike from one subcategory, Road Bikes, to another, Mountain Bikes. You cannot move Road-650 directly under a category, like 1 {Bikes}. Each time you move a member in the hierarchy tree, the member's domain-based attribute value changes to reflect the move.

Notes

All members in a derived hierarchy tree are sorted by code. You cannot change the sort order.

If a member's domain-based attribute is blank and the attribute is used for a derived hierarchy, the member is not displayed in the hierarchy. Create business rules to require attributes to be populated. For more information, see [Require Attribute Values \(Master Data Services\)](#).

Related Tasks

Task Description	Topic
Create a new derived hierarchy.	Create a Derived Hierarchy (Master Data Services)
Hide or delete levels in an existing derived hierarchy.	Hide or Delete Levels in a Derived Hierarchy (Master Data Services)
Change the name of an existing derived hierarchy.	Change a Derived Hierarchy Name (Master Data Services)
Delete an existing derived hierarchy.	Delete a Derived Hierarchy (Master Data Services)

Related Content

- [Domain-Based Attributes \(Master Data Services\)](#)
- [Explicit Hierarchies \(Master Data Services\)](#)
- [Recursive Hierarchies \(Master Data Services\)](#)
- [Derived Hierarchies with Explicit Caps \(Master Data Services\)](#)
- [Collections \(Master Data Services\)](#)

Recursive Hierarchies (Master Data Services)

SQL Server 2012

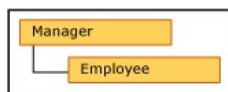
In Master Data Services, a recursive hierarchy is a derived hierarchy that includes a recursive relationship. A recursive relationship exists when an entity has a domain-based attribute based on the entity itself.

Recursive Hierarchy Example

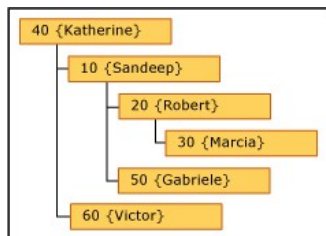
A typical recursive hierarchy example is an organizational structure. In Master Data Services, you would do this by creating an Employee entity with a domain-based attribute called Manager. The Manager attribute is populated from the list of employees. In this sample organization, all employees can be managers.

Name	Code	Manager
Sandeep	10	Katherine
Robert	20	Sandeep
Marcia	30	Robert
Katherine	40	
Gabriele	50	Sandeep
Victor	60	Katherine

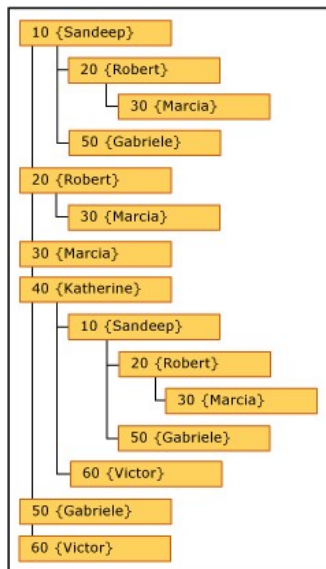
You can create a derived hierarchy that highlights the relationship between the Employee entity and the Manager domain-based attribute.



To include each member in the hierarchy only once, you can anchor null relationships. When you do, members with blank domain-based attribute values are displayed at the top level of the hierarchy.



If you do not anchor null relationships, members are included multiple times. All members are displayed at the top level. They are also displayed under members of which they are attributes.



In this example, Marcia is at the top level. She is not the manager of any employees because she is not used as a domain-based attribute value for any other Employee members. Robert, in contrast, has a level beneath him because Marcia has Robert as her Manager attribute value.

Rules

- A derived hierarchy cannot contain more than one recursive relationship. It can, however, have other derived relationships (for example, a derived hierarchy that contains a recursive Manager to Employee relationship can also have Country to Manager and Employee to Store relationships).
- You cannot assign member permissions (on the **Hierarchy Members** tab) to members in a recursive hierarchy.
- Recursive hierarchies cannot include circular relationships. For example, Katherine cannot be Sandeep's manager if Sandeep is her manager. Also, Katherine cannot manage herself.

Related Tasks

Task Description	Topic
Create a derived hierarchy.	Create a Derived Hierarchy (Master Data Services)
Change the name of an existing derived hierarchy.	Change a Derived Hierarchy Name (Master Data Services)
Delete an existing derived hierarchy.	Delete a Derived Hierarchy (Master Data Services)

Related Content

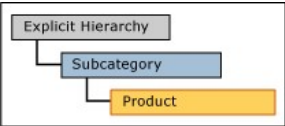
- [Domain-Based Attributes \(Master Data Services\)](#)
- [Derived Hierarchies \(Master Data Services\)](#)

Derived Hierarchies with Explicit Caps (Master Data Services)

SQL Server 2012

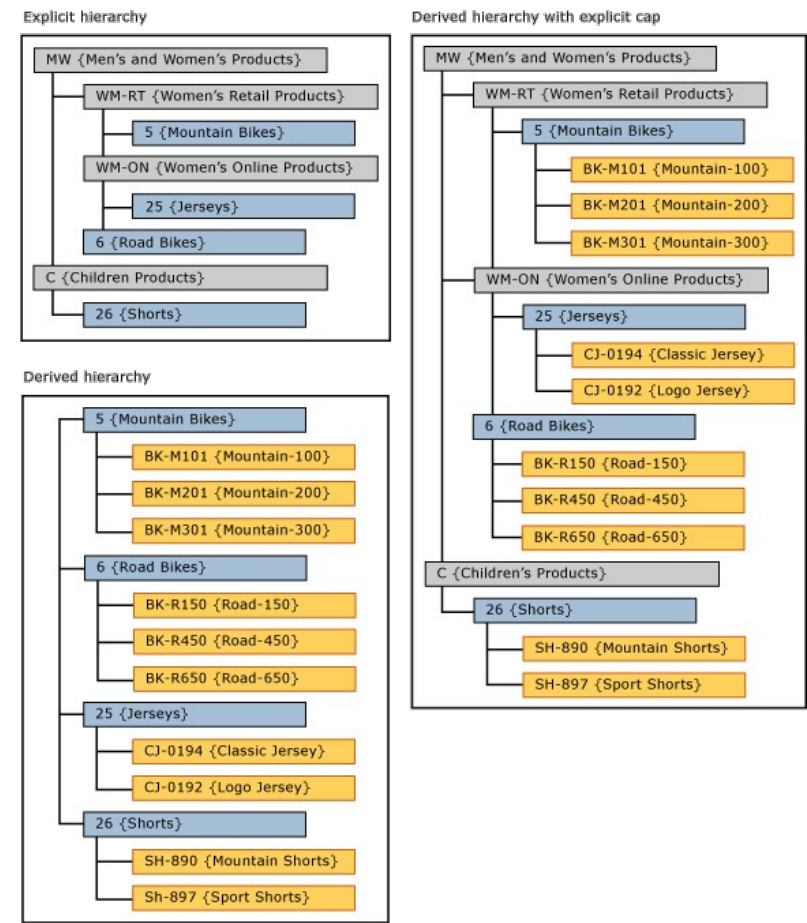
In Master Data Services, when the levels from an explicit hierarchy are used as the top levels of a derived hierarchy, this is called a derived hierarchy with an explicit cap. The explicit hierarchy must be based on the same entity as the entity at the top of the derived hierarchy.

In the Master Data Manager user interface (UI), you create this type of hierarchy by dragging an explicit hierarchy to the top of a derived hierarchy.



Derived Hierarchy with Explicit Cap Example

In this example, the members in the explicit hierarchy are from the Subcategory entity. In the derived hierarchy, the top-level members are also from the Subcategory entity.



By using the explicit hierarchy at the top of a derived hierarchy, the derived hierarchy becomes ragged.

Rules

- You cannot have more than one explicit hierarchy in a derived hierarchy with explicit cap.
- You can use the same explicit hierarchy as a cap for multiple derived hierarchies.
- You cannot assign hierarchy member permissions to derived hierarchies with explicit caps. If you assign permissions to either the explicit hierarchy or the derived hierarchy individually, the permissions affect both hierarchies.

Related Tasks

Task Description	Topic
Create a derived hierarchy.	Create a Derived Hierarchy (Master Data Services)

Create an explicit hierarchy.	Create an Explicit Hierarchy (Master Data Services)
Delete an existing derived hierarchy.	Delete a Derived Hierarchy (Master Data Services)

Related Content

- [Derived Hierarchies \(Master Data Services\)](#)
- [Explicit Hierarchies \(Master Data Services\)](#)

Create a Derived Hierarchy (Master Data Services)

SQL Server 2012

In Master Data Services, create a derived hierarchy when you want a level-based hierarchy that ensures that members exist at the correct level. Derived hierarchies are based on the domain-based attribute relationships that exist in a model.

Note

If a domain-based attribute value doesn't exist for a member, the member is not included in the derived hierarchy. See [Require Attribute Values \(Master Data Services\)](#) to require a domain-based attribute value for all members.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To create a derived hierarchy

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Derived Hierarchies**.
3. On the **Derived Hierarchy Maintenance** page, from the **Model** list, select a model.
4. Click **Add derived hierarchy**.
5. On the **Add Derived Hierarchy** page, in the **Derived hierarchy name** box, type a name for the hierarchy.

Tip

Use a name that describes the levels in the hierarchy, for example **Product to Subcategory to Category**.

6. Click **Save derived hierarchy**.
7. On the **Edit Derived Hierarchy** page, in the **Available Entities and Hierarchies** pane, click an entity or hierarchy and drag it to the **Current Levels** pane.
8. Continue dragging entities or hierarchies until your hierarchy is complete.
9. Click **Back**.

See Also

Concepts

[Derived Hierarchies \(Master Data Services\)](#)
[Derived Hierarchies with Explicit Caps \(Master Data Services\)](#)
[Domain-Based Attributes \(Master Data Services\)](#)

Hide or Delete Levels in a Derived Hierarchy (Master Data Services)

SQL Server 2012

In Master Data Services, hide a level in a derived hierarchy when you require the level for grouping but you do not need to show the level. Delete a level when you do not want to use it for grouping.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To hide or delete levels in a derived hierarchy

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Derived Hierarchies**.
3. On the **Derived Hierarchy Maintenance** page, from the **Model** list, select a model.
4. Select the row for the derived hierarchy you want to edit.
5. Click **Edit selected derived hierarchy**.
6. In the **Current Levels** pane:
 - To hide a level, click a level other than the top or bottom. From the **Visible** list, select **No**. Then click **Save selected hierarchy item**.
 - To delete the top level, click **Delete selected hierarchy item**. In the confirmation dialog box, click **OK**. You can delete the top level only.

See Also

Tasks

[Move Members within a Hierarchy \(Master Data Services\)](#)

Concepts

[Derived Hierarchies \(Master Data Services\)](#)

Change a Derived Hierarchy Name (Master Data Services)

SQL Server 2012

In Master Data Services, you can change the name of a derived hierarchy.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To change a derived hierarchy name

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Derived Hierarchies**.
3. On the **Derived Hierarchy Maintenance** page, from the **Model** list, select a model.
4. Select the row for the derived hierarchy that you want to rename.
5. Click **Edit selected derived hierarchy**.
6. On the **Edit Derived Hierarchy** page, click **Edit derived hierarchy name**.
7. In the **Derived hierarchy name** box, type the updated name of the hierarchy.
8. Click **Save derived hierarchy**.

See Also

Tasks

[Create a Derived Hierarchy \(Master Data Services\)](#)

[Delete a Derived Hierarchy \(Master Data Services\)](#)

Concepts

[Derived Hierarchies \(Master Data Services\)](#)

Delete a Derived Hierarchy (Master Data Services)

SQL Server 2012

In Master Data Services, delete a derived hierarchy when you are sure you no longer need it.

Note

Deleting a derived hierarchy has no effect on the attribute relationships that the hierarchy is based on.

Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete a derived hierarchy

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **Manage** and click **Derived Hierarchies**.
3. On the **Derived Hierarchy Maintenance** page, from the **Model** list, select a model.
4. Select the row for the derived hierarchy that you want to delete.
5. Click **Delete selected hierarchy**.
6. In the confirmation dialog box, click **OK**.

See Also

Tasks

[Create a Derived Hierarchy \(Master Data Services\)](#)

Concepts

[Derived Hierarchies \(Master Data Services\)](#)

Collections (Master Data Services)

SQL Server 2012

A collection is a group of leaf and consolidated members from a single entity. Use collections when you do not need a complete hierarchy and you want to view different groupings of members for reporting or analysis, or when you need to create a taxonomy.

What Collections Contain

Collections do not limit the number or type of members you can include, as long as the members are within the same entity. A collection can contain leaf and consolidated members from multiple mandatory and non-mandatory explicit hierarchies.

When you create a collection, you are not creating a hierarchical structure; you are creating a flat list of members. When you select a node from a hierarchy and add it to the collection, the consolidated member you selected is the only member added to the collection.

A collection can also contain other collections. You can use collections of collections to create taxonomies.

When you create a collection you are automatically listed as the owner. If you are an administrator, you can create other attributes for your collection as needed.

Note

Before you can create a collection, the entity must be enabled for explicit hierarchies. For more information, see [Enable an Entity for Explicit Hierarchies and Collections \(Master Data Services\)](#).

Subscription Views for Collections

There are two types of subscription views that show collections. The **Collection attributes** format shows a list of collections and any attributes related to the collections (like description or owner). The **Collections** format shows all members in all collections, as well as each members weight and sort order. For more information, see [Exporting Data \(Master Data Services\)](#).

If you set weight values for specific members in a collection, these values are available in related subscription views.

Related Tasks

Task Description	Topic
Enable an entity for explicit hierarchies and collections.	Enable an Entity for Explicit Hierarchies and Collections (Master Data Services)
Create a new collection.	Create a Collection (Master Data Services)
Add members to an existing collection.	Add Members to a Collection (Master Data Services)

Related Content

- [Explicit Hierarchies \(Master Data Services\)](#)
- [Exporting Data \(Master Data Services\)](#)

Create a Collection (Master Data Services)

SQL Server 2012

In Master Data Services, create a collection when you want to create flat lists of leaf and consolidated members. Collections do not need to include all members from the entity.

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.
- You must have a minimum of **Update** permission to the collection model object for the entity.
- The entity must be enabled for explicit hierarchies and collections. For more information, see [Enable an Entity for Explicit Hierarchies and Collections \(Master Data Services\)](#).

To create a collection

1. On the Master Data Manager home page, from the **Model** list, select a model.
2. From the **Version** list, select a version.
3. Click **Explorer**.
4. From the menu bar, point to **Collections** and click *entity_name*.
5. Click **Add collection**.
6. On the **Details** tab, in the **Name** box, type a name for the collection.
7. In the **Code** box, type a unique code for the collection.
8. Optionally, in the **Description** box, type a description for the collection.
9. Click **OK**.

Next Steps

- [Add Members to a Collection \(Master Data Services\)](#)

See Also

Tasks

[Delete a Member or Collection \(Master Data Services\)](#)

[Create an Explicit Hierarchy \(Master Data Services\)](#)

Concepts

[Collections \(Master Data Services\)](#)

Add Members to a Collection (Master Data Services)

SQL Server 2012

In Master Data Services, you can add leaf and consolidated members to a collection.

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.
- You must have a minimum of **Update** permission to the collection model object that you are adding members to.
- A collection must exist. For more information, see [Create a Collection \(Master Data Services\)](#).

To add members to a collection

1. On the Master Data Manager home page, from the **Model** list, select a model.
2. From the **Version** list, select a version.
3. Click **Explorer**.
4. From the menu bar, point to **Collections** and click *entity_name*.
5. In the grid, click the row for the collection you want to add members to.
6. Click the **Collection Members** tab.
7. Click **Edit Members**.
8. To filter the list of available members, select from the list on the left.
9. Click the row with the member you want to add and click **Add**.
10. Optionally, rearrange collection members by clicking **Up** or **Down**.
11. Optionally, set weight values by clicking the value in the **Weight** column.

See Also

Concepts

[Collections \(Master Data Services\)](#)

Exporting Data (Master Data Services)

SQL Server 2012

You can export Master Data Services data to subscribing systems by creating subscriptions views. Any subscribing system can then view the published data in the Master Data Services database. For more information about views, see [Views](#).

Subscription View Formats

When you create a view in Master Data Manager, you choose from a set of standard view formats that Master Data Services provides. You can use these formats to create views that show:

- All leaf members and their attributes.
- All consolidated members and their attributes.
- All collections and their attributes.
- The members explicitly added to a collection.
- The members in a derived hierarchy, in either a parent child or level format.
- The members in all explicit hierarchies for an entity, in either a parent child or level format.

Subscription Views Can Become Out-of-Date

After you create a subscription view for an entity or hierarchy, changes to the associated model objects are not automatically reflected in the view. You might need to regenerate a subscription view in Master Data Manager to reflect changes to model objects. The **Changed** column on the **Export** page is updated to **True** when model objects change. **True** indicates that you should edit the subscription view and save it, which regenerates the view.

Related Tasks

Task Description	Topic
Create a subscription view of your master data.	Create a Subscription View (Master Data Services)
Delete an existing subscription view.	Delete a Subscription View (Master Data Services)

Related Content

- [Subscription View Formats \(Master Data Services\)](#)
- [Views](#)

Create a Subscription View (Master Data Services)

SQL Server 2012

Create a subscription view when you want to create a view of your data in the Master Data Services database for use by subscribing systems.

Prerequisites

To perform this procedure:

- You must have permission to access the **Integration Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To create a subscription view

1. In Master Data Manager, click **Integration Management**.
2. From the menu bar, click **Create Views**.
3. On the **Subscription Views** page, click **Add subscription view**.
4. In the **Create Subscription View** pane, in the **Subscription view name** box, type a name for the view.
5. From the **Model** list, select a model.
6. Select either the **Version** or **Version Flag** option, and then select from the corresponding list.



Tip

Create a subscription view based on a version flag. When you lock a version, you can reassign the flag to an open version without updating the subscription view.

7. Select either the **Entity** or **Derived hierarchy** option, and then select from the corresponding list.
8. From the **Format** list, select a subscription view format.
9. If you chose **Explicit levels** or **Derived levels** from the **Format** list, type the number of levels in the hierarchy to include in the view.
10. Click **Save**.

See Also

Tasks

[Delete a Subscription View \(Master Data Services\)](#)

[Create a Version Flag \(Master Data Services\)](#)

Concepts

[Exporting Data \(Master Data Services\)](#)

Delete a Subscription View (Master Data Services)

SQL Server 2012

In Master Data Services, delete a subscription view when it is no longer needed. Deleting a subscription view in Master Data Manager removes the view from the Master Data Services database.

Prerequisites

To perform this procedure:

- You must have permission to access the **Integration Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete a subscription view

1. In Master Data Manager, click **Integration Management**.
2. From the menu bar, click **Create Views**.
3. On the **Subscription Views** page, select the row of the view that you want to delete.
4. Click **Delete subscription view**.
5. In the confirmation dialog box, click **OK**.

See Also

Tasks

[Create a Subscription View \(Master Data Services\)](#)

Concepts

[Exporting Data \(Master Data Services\)](#)

Subscription View Formats (Master Data Services)

SQL Server 2012

Based on the entity or derived hierarchy you select, the following formats are available for your subscription view.

Subscription View Formats

Name	Description
Leaf Members	Contains leaf members and their associated attribute values.
Consolidated Members	Contains consolidated members and their associated attribute values.
Collection Memberships	Contains a list of collections and their associated attribute values.
Collections	Contains a list of collections and the members in each, along with weight values and sort order.
Explicit Parent Child	Contains explicit hierarchy structures for an entity in a parent child format.
Explicit Levels	Contains explicit hierarchy structures for an entity in level format.
Derived Parent Child (Derived Hierarchy View)	Contains a derived hierarchy structure in parent child format.
Derived Levels (Derived Hierarchy View)	Contains a derived hierarchy structure in level format.

See Also

Tasks

[Create a Subscription View \(Master Data Services\)](#)

Concepts

[Exporting Data \(Master Data Services\)](#)

Versions (Master Data Services)

SQL Server 2012

In Master Data Services, you can create multiple versions of the master data within a model. Versions can be locked while you validate your data and committed after the data is validated. Committed versions form an auditable record of changes. Each version you create contains all members, attribute values, hierarchy members, hierarchy relationships, and collections for the model.

When to Use Versions

Use versions to:

- Maintain an auditable record of your master data as it changes over time.
- Prevent users from making changes while you ensure all data validates successfully against business rules.
- Lock down a model for use by subscribing systems.
- Test different hierarchies without implementing them right away.

Note

When you change the structure of your model, such as when you create a new entity or domain-based attribute, the change applies to all versions. If you view an earlier version of the model, the entity or attribute is displayed, but no data exists.

Version Flags

When a version is ready for users or for a subscribing system, you can set a flag to identify the version. You can move this flag from version to version as needed. Flags help users and subscribing systems identify which version of a model to use.

Workflow for Version Management

Use the following workflow for version management:

1. An initial version is created automatically when you create a model and populate the Master Data Services database with your company's master data. Based on permissions, users can make changes to this version as needed.
2. When you want to commit a version of a model, lock the version so that only model administrators can update the data. For more information, see [Administrators \(Master Data Services\)](#). If notifications are configured, an email notification is sent to model administrators each time the version's status changes. For more information, see [Configure Email Notifications \(Master Data Services\)](#).
3. Apply business rules to the locked version's data and review any validation issues. If necessary, you can fill in missing information or revert the transaction that caused the issue. You can also unlock the version for users to make changes.
4. When all the data passes validation, commit the version and flag it for use by subscribing systems. Changes cannot be made to a committed version.
5. Copy the committed version and notify users that they can begin working in a new version of the model.

Sequential or Simultaneous Versions

You can create sequential or simultaneous versions of your model.

- **Sequential versions.** Each time you commit a version, create a new copy and give the version the next sequential number. For example, you can copy **Version 7** of your model and name the copy **Version 8**.
- **Simultaneous versions.** Create simultaneous versions of your model when you want to work on two or more versions of your data at once. This is useful when your company has reorganizations or mergers that coincide with the normal course of business and you want to determine how the new master data might fit into your existing structures.

Note

A setting in Master Data Services Configuration Manager determines whether or not you can copy all versions or only those that are committed. To create simultaneous versions you must configure Master Data Services to allow you to copy all versions. This setting is also available in the System Settings table. For more information, see [System Settings \(Master Data Services\)](#).

Related Tasks

Task Description	Topic
------------------	-------

Change the name of an existing version.	Change a Version Name (Master Data Services)
Lock a version so only administrators can edit its data.	Lock a Version (Master Data Services)
Unlock a version so users can edit its data.	Unlock a Version (Master Data Services)
Commit a version after all data is validated.	Commit a Version (Master Data Services)
Create a new flag to mark a version.	Create a Version Flag (Master Data Services)
Change the name of an existing version flag.	Change a Version Flag Name (Master Data Services)
Assign an existing flag to a version.	Assign a Flag to a Version (Master Data Services)
Create a new copy of an existing version	Copy a Version (Master Data Services)
Delete an existing version.	Delete a Version (Master Data Services)

Related Content

- [Reverse a Transaction \(Master Data Services\)](#)
- [Notifications \(Master Data Services\)](#)
- [Business Rules \(Master Data Services\)](#)

Change a Version Name (Master Data Services)

SQL Server 2012

In Master Data Services, change the version name to customize it for your organization.

Prerequisites

To perform this procedure:

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To change the version name

1. In Master Data Manager, click **Version Management**.
2. On the **Manage Versions** page, in the grid, double-click a cell in the **Name** column.
3. Type the new name.
4. Press ENTER.

See Also

Tasks

[Validate a Version against Business Rules \(Master Data Services\)](#)

[Lock a Version \(Master Data Services\)](#)

[Commit a Version \(Master Data Services\)](#)

Concepts

[Versions \(Master Data Services\)](#)

Lock a Version (Master Data Services)

SQL Server 2012

In Master Data Services, lock a version of a model to prevent changes to the model's members and their attributes.

Note

When a version is locked, model administrators can continue to add, edit, and remove members. Other users with permission to the model can view members only.

Prerequisites

To perform this procedure:

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- The version's status must be **Open**.

To lock a version

1. In Master Data Manager, click **Version Management**.
2. On the **Manage Versions** page, select the row for the version that you want to lock.
3. Click **Lock**.
4. In the confirmation dialog box, click **OK**.

Next Steps

- [Validate a Version against Business Rules \(Master Data Services\)](#)
- [Commit a Version \(Master Data Services\)](#)

See Also

Tasks

[Unlock a Version \(Master Data Services\)](#)

Concepts

[Versions \(Master Data Services\)](#)

Unlock a Version (Master Data Services)

SQL Server 2012

In Master Data Services, unlock a version of a model to enable changes to the model's members and their attributes.

Prerequisites

To perform this procedure:

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- The version's status must be **Locked**. For more information, see [Lock a Version \(Master Data Services\)](#).

To unlock a version

1. In Master Data Manager, click **Version Management**.
2. On the **Manage Versions** page, select the row for the version you want to unlock.
3. Click **Unlock**.
4. In the confirmation dialog box, click **OK**.

Next Steps

- [Lock a Version \(Master Data Services\)](#)

See Also

Concepts
[Versions \(Master Data Services\)](#)

Commit a Version (Master Data Services)

SQL Server 2012

In Master Data Services, commit a version of a model to prevent changes to the model's members and their attributes. Committed versions cannot be unlocked.

Prerequisites

To perform this procedure:

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- The version's status must be **Locked**. For more information, see [Lock a Version \(Master Data Services\)](#).
- All members must have validated successfully.

To commit a version

1. In Master Data Manager, click **Version Management**.
2. On the **Manage Versions** page, on the menu bar, click **Validate Version**.
3. On the **Validate Version** page, select the model and version you want to commit.
4. Click **Commit**.
5. In the confirmation dialog box, click **OK**.

Next Steps

- [Create a Version Flag \(Master Data Services\)](#)
- [Assign a Flag to a Version \(Master Data Services\)](#)
- [Copy a Version \(Master Data Services\)](#)

See Also

Concepts
[Versions \(Master Data Services\)](#)

Create a Version Flag (Master Data Services)

SQL Server 2012

In Master Data Services, create a version flag to assign to a version. The flag can indicate the version that users or subscribing systems should use.

Prerequisites

To perform this procedure:

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To create a version flag

1. In Master Data Manager, click **Version Management**.
2. On the **Manage Versions** page, from the menu bar, point to **Manage** and then click **Flags**.
3. On the **Manage Version Flags** page, from the **Model** field, select the model for which you want to create a flag.
4. Click **Add**.
5. In the **Name** box, type a name.
6. In the **Description** box, type a description.
7. In the **Committed Versions Only** field, select **True** to indicate that the flag can be assigned to versions with a status of **Committed** only. Select **False** to indicate that the flag can be assigned to versions with any status.
8. Click **Save**.

Next Steps

- [Assign a Flag to a Version \(Master Data Services\)](#)

See Also

Tasks

[Change a Version Flag Name \(Master Data Services\)](#)

Concepts

[Versions \(Master Data Services\)](#)

Change a Version Flag Name (Master Data Services)

SQL Server 2012

In Master Data Services, you can change the name of a version flag.

Prerequisites

To perform this procedure:

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To change a version flag name

1. In Master Data Manager, click **Version Management**.
2. On the **Manage Versions** page, from the menu bar, point to **Manage** and then click **Flags**.
3. On the **Manage Version Flags** page, from the **Model** field, select a model.
4. In the grid, in the row for the flag name, double-click the **Name** field.
5. Type a new name for the flag.
6. Press ENTER.

See Also

Tasks

[Assign a Flag to a Version \(Master Data Services\)](#)

Concepts

[Versions \(Master Data Services\)](#)

Assign a Flag to a Version (Master Data Services)

SQL Server 2012

In Master Data Services, assign a flag to a version to indicate the version that users or subscribing systems should use.

Note

Version flags can be assigned to only one version at a time. If you assign a flag that is already assigned to another version, the flag is moved to the version you selected.

Prerequisites

To perform this procedure:

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- You must have created a version flag to assign. For more information, see [Create a Version Flag \(Master Data Services\)](#).

To assign a flag to a version

1. In Master Data Manager, click **Version Management**.
2. On the **Manage Versions** page, in the row for the version to which you want to assign a flag, double-click the cell in the **Flag** column.
3. From the list, select the flag you want to assign.

Note

If the flag you want is not available, the flag might be available for **Committed** versions only. To confirm, go to the **Manage Version Flags** page and view the **Committed Versions Only** field for the flag.

4. Press ENTER to save the change.

See Also

Tasks

[Create a Version Flag \(Master Data Services\)](#)

Concepts

[Versions \(Master Data Services\)](#)

Copy a Version (Master Data Services)

SQL Server 2012

In Master Data Services, copy a version of the model to create a new version of it.

Prerequisites

To perform this procedure:

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To copy a version

1. In Master Data Manager, click **Version Management**.
2. On the **Manage Versions** page, select the row for the version that you want to copy.

Note

Depending on a setting in Master Data Services Configuration Manager, you might be able to copy versions with the **Committed** status only. For more information, see [System Settings \(Master Data Services\)](#).

3. Click **Copy**.
4. In the confirmation dialog box, click **OK**.

Next Steps

- [Change a Version Name \(Master Data Services\)](#)

See Also

Concepts

[Versions \(Master Data Services\)](#)

Delete a Version (Master Data Services)

SQL Server 2012

In Master Data Services, delete a version when you are sure you no longer need the master data associated with the version. After you delete a version, you cannot retrieve the associated master data.

⚠ Caution

If a model has only one version and you delete it, the model becomes unusable.

▀ Prerequisites

To perform this procedure:

- You must have permission to view the `mdm.viw_SYSTEM_SCHEMA_VERSION` view and to execute the `mds.udpVersionDelete` stored procedure in the Master Data Services database. For more information, see [Database Object Security \(Master Data Services\)](#).

To delete a version

- Open SQL Server Management Studio and connect to the Database Engine instance for your Master Data Services database.
- Open the `mdm.viw_SYSTEM_SCHEMA_VERSION` view.
- Find the version of the model you want to delete and copy the value in the **ID** field.
- Create a new query.
- Type the following text, replacing *version_ID* with the value you copied in step 2.

```
EXEC [mdm].[udpVersionDelete] @VersionID='version_ID'
```

- Run the query.

📌 Note

You may have to wait a few minutes before the Web application reflects the change.

▀ See Also

Tasks

[Copy a Version \(Master Data Services\)](#)

Concepts

[Versions \(Master Data Services\)](#)

Validation (Master Data Services)

SQL Server 2012

In Master Data Services, data is validated to ensure its accuracy. Some validation occurs automatically, and other validation is based on business rules that are created by administrators.

When Data Validation Occurs

Validation occurs at different times, and is displayed differently in the Master Data Services web application.

Validation Type	Standards Determined by	When it Occurs	Displayed in the MasterData Manager web UI as	Displayed in the Add-in for Excel as	Is Data Saved to the MDS Repository?
Business rule validation	An MDS administrator	<ul style="list-style-type: none">Automatically when a user adds or edits data.Manually when a user applies business rules.Manually when an administrator in the Version Management functional area of the Master Data Manager web application validates a version against business rules.	Validation Errors	ValidationStatus	Yes
Data type and content validation	An MDS administrator, when creating model objects (for example, an attribute's length or data type)	Automatically when a user adds or edits data	Input Errors	InputStatus	No
Data type and content validation	SQL Server or Master Data Services	Automatically when a user adds or edits data	Input Errors	InputStatus	No

Related Tasks

Task Description	Topic
Create business rules and publish them, so that data is validated against them.	Create and Publish a Business Rule (Master Data Services)
Validate a version of data against business rules. Administrators only.	Validate a Version against Business Rules (Master Data Services)
Validate specific subsets of data against business rules. All users with permission to the Explorer functional area.	Validate Specific Members against Business Rules (Master Data Services)
Validate specific subsets of data against business rules. All users with permission to the Explorer functional area and using the Add-in for Excel.	Apply Business Rules (MDS Add-in for Excel)

See Also

Concepts
[Business Rules \(Master Data Services\)](#)

Validate a Version against Business Rules (Master Data Services)

SQL Server 2012

In Master Data Services, validate a version to apply business rules to all members in the model version.

This procedure explains how to use the Master Data Manager web application to validate data. If you have permission in the MDS database, you can use a stored procedure instead. For more information, see [Validation Stored Procedure \(Master Data Services\)](#).

Note

All members must pass validation before a version can be committed.

Prerequisites

To perform this procedure:

- You must have permission to access the **Version Management** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- The version's status must be **Open** or **Locked**.
- On the **Validate Versions** page, members must exist with a status other than **Validation succeeded**.

To validate a version

1. In Master Data Manager, click **Version Management**.
2. On the **Manage Versions** page, from the menu bar, click **Validate Version**.
3. On the **Validate Versions** page, select the model and version you want to validate.
4. Click **Validate**.
5. In the confirmation dialog box, click **OK**.

Note

When the progress indicator is no longer displayed, the version has finished validation.

Next Steps

- [Lock a Version \(Master Data Services\)](#)

See Also

Tasks

[Validate Specific Members against Business Rules \(Master Data Services\)](#)

Concepts

[Validation Statuses \(Master Data Services\)](#)

[Validation Stored Procedure \(Master Data Services\)](#)

[Versions \(Master Data Services\)](#)

[Business Rules \(Master Data Services\)](#)

Validation Statuses (Master Data Services)

SQL Server 2012

In the **Version Management** functional area, on the **Validate Version** page, the following statuses are possible.

Status	Description
Waiting to be validated	All newly created members that are waiting to be validated.
Waiting to be revalidated	Existing members that are waiting to be validated.
Validation succeeded	Members that passed business rule validation.
Validation failed	Members that failed business rule validation.
Waiting for dependent member revalidation	Updated consolidated members waiting to be validated along with child members.

See Also

Tasks

[Validate a Version against Business Rules \(Master Data Services\)](#)

Concepts


[Versions \(Master Data Services\)](#)

Validation Stored Procedure (Master Data Services)

SQL Server 2012

In Master Data Services, validate a version to apply business rules to all members in the model version.

This topic explains how to use the **mdm.udpValidateModel** stored procedure to validate data. If you are an administrator in the Master Data Manager web application, you can do validation in the UI instead. For more information, see [Validate a Version against Business Rules \(Master Data Services\)](#).

 **Note**

If you invoke validation before the staging process is complete, members that have not finished staging will not be validated.

Example

```
DECLARE @ModelName nvarchar(50) = 'Customer'
DECLARE @Model_ID int
DECLARE @UserName nvarchar(50) = 'DOMAIN\user_name'
DECLARE @User_ID int
DECLARE @Version_ID int

SET @User_ID = (SELECT ID
                FROM mdm.tblUser u
                WHERE u.UserName = @UserName)

SET @Model_ID = (SELECT Top 1 Model_ID
                 FROM mdm.viw_SYSTEM_SCHEMA_VERSION
                 WHERE Model_Name = @ModelName)

SET @Version_ID = (SELECT MAX(ID)
                  FROM mdm.viw_SYSTEM_SCHEMA_VERSION
                  WHERE Model_ID = @Model_ID)

EXECUTE mdm.udpValidateModel @User_ID, @Model_ID, @Version_ID, 1
```

Parameters

The parameters of this procedure are as follows:

Parameter	Description
UserID	The user ID.
Model_ID	The model ID.
Version_ID	The version ID.

See Also

- Tasks
- [Validate a Version against Business Rules \(Master Data Services\)](#)
- Concepts
- [Importing Data \(Master Data Services\)](#)

Metadata (Master Data Services)

SQL Server 2012

In Master Data Services, user-defined metadata is information that you use to describe model objects. For example, you may want to track the owners of a particular model or entity, or track the source systems that supply data to an entity.

User-defined metadata is managed by a model called **Metadata**. This model is automatically included when Master Data Services is installed, and it is similar to all other MDS models, except that you cannot create versions of it.

After you populate the Metadata model with user-defined metadata, you can include it in subscription views, so it can be consumed by subscribing systems.

Metadata Entities

The Metadata model includes five entities, each of which represents a type of master data model object that supports user-defined metadata. For example, the **Model Metadata Definition** entity contains members that represent models, and the **Attribute Metadata Definition** entity has members that represent all attributes in all models.

To define metadata for a model object, you populate one of these member's attributes. For example, in the **Entity Metadata Definition** entity, you can populate the Price member's Description attribute with the text: **The product price when sold to a customer**.

The members in the Metadata model are automatically updated whenever model objects that support user-defined metadata are added or deleted.

The Metadata model cannot be versioned, have version flags added or changed, or be saved as a model deployment package. However, it has all the same other functionality available to other master data models. For example, you might implement a set of business rules on the Metadata model to enforce data policies.

Customizing Your Metadata Model

Each metadata definition entity includes Name, Code, and Description attributes. You may want to create additional attributes to further describe your model objects.

For example, you might create:

- A domain-based attribute named Owner, which you use to track the owner of each model object.
- A free-form attribute named Last Review Date, which you use to track the date that an object was last reviewed by the owner.
- A domain-based attribute named Sources, which you use to track and manage the source systems that interact with the Master Data Services instance.

Related Tasks

Task Description	Topic
Add metadata to a model object.	Add Metadata (Master Data Services)

Related Content

- [Exporting Data \(Master Data Services\)](#)

Add Metadata (Master Data Services)

SQL Server 2012

In Master Data Services, add user-defined metadata to the **Metadata** model to describe your model objects.

Prerequisites

To perform this procedure:

- You must have permission to access the **Explorer** functional area.
- For the **Metadata** model, you must have **Add** permission to the metadata entity or leaf member type that you want to add.
- The version's status must be **Open**.

To add metadata

1. On the Master Data Manager home page, from the **Model** list, select the **Metadata** model.
2. From the **Version** list, select a version.
3. Click **Explorer**.
4. From the menu bar, point to **Entities** and click the metadata entity you want to add.
5. Click the **Add Member** button. The **Adding New Member** page will be displayed on the right of this page.
6. Enter the new name, code, and description, and then click **OK**.

See Also

Concepts

[Metadata \(Master Data Services\)](#)

Deploying Models (Master Data Services)

SQL Server 2012

In Master Data Services, a package is an XML file that contains a deployable model structure, and optionally, data from the model. Use model packages to move copies of models from one MDS environment to another, or to create new models in your existing MDS environment.

⚠ Important

Packages can be deployed to the edition of SQL Server they were created in only. This means that packages created in SQL Server 2008 R2 cannot be deployed to SQL Server 2012.

Tools for Deploying Models

To work with model packages, you can use one of three tools, depending on your needs.

- **MDSModelDeploy tool:** To create and deploy model objects and data, use the MDSModelDeploy.exe tool. If you selected the default path when installing MDS, this tool is located on *drive:\Program Files\Microsoft SQL Server\110\Master Data Services\Configuration*.
- **Model Deployment wizard:** To create and deploy packages of the model structure only, use the wizard in the Master Data Manager web application. You cannot use this wizard to deploy data.
- **Model Package Editor:** To edit a model package, use the ModelPackageEditor.exe that launches the Model Package Editor wizard. You use this wizard to edit a package that was created by the MDSModelDeploy tool or the Model Deployment wizard. If you selected the default path when installing MDS, this tool is located on *drive:\Program Files\Microsoft SQL Server\110\Master Data Services\Configuration*.

⚠ Important

You can use the MDSDeployModel to create a new model, create a clone of a model, or update an existing model and its data. If you use the MDSModelDeploy tool to update an existing model and its data, and the package does not contain an entity, attribute, or member that exists in the destination model, MDSModelDeploy will not delete that entity, attribute, or member from the model.

What Packages Contain

A model package is an XML file that is saved with the .pkg extension. When you create a deployment package, you can decide whether or not to include data. If you decide to include data, you must select a version of the data to include.

All model objects are included in a package. These objects are:

- Entities
- Attributes
- Attribute groups
- Hierarchies
- Collections
- Business rules
- Version flags
- Subscription views

User-defined metadata, file attributes, and user and group permissions are not included. After you deploy a model, you must update these manually.

Sample Packages

Sample package files are included when you install Master Data Services. These package files are in the Master Data Services\Samples\Packages directory where you installed Master Data Services. When you deploy these sample packages by using the MDSModelDeploy tool, sample models are created and populated with data.

Related Tasks

Task Description	Topic
Create a new deployment package of model objects and/or data by using the MDSModelDeploy tool.	Create a Model Deployment Package by Using MDSModelDeploy
Create a new deployment package of model objects only by using the wizard.	Create a Model Deployment Package by Using the Wizard
Deploy a package of model objects and data by using the MDSModelDeploy tool.	Deploy a Model Deployment Package by Using

	MDSModelDeploy
Deploy a package of model objects only by using the wizard.	Deploy a Model Deployment Package by Using the Wizard
Edit a model deployment package to deploy selected parts of a model, rather than the entire model.	Edit a Model Deployment Package

Related Content

- [Model Deployment Options \(Master Data Services\)](#)

Create a Model Deployment Package by Using the Wizard

SQL Server 2012

Use the Master Data Manager model deployment wizard to create a package of the model objects only. If you need to include data in the package, see [Create a Model Deployment Package by Using MDSModelDeploy](#).

Prerequisites

To perform this procedure:

- In the Master Data Manager web application, you must have permission to access the **System Administration** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- A model must exist for you to create a package of. For more information, see [Create a Model \(Master Data Services\)](#).

To create a model deployment package

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **System** and click **Deployment**.
3. On the **Model Deployment Wizard**, click **Create**.
4. On the **Create Package** page, select a model from the **Model** list.
5. Click **Next**.
6. Click **Download**.
7. Save the file.
8. Click **Close** to close the wizard.

Next Steps

- [Deploy a Model Deployment Package by Using the Wizard](#)

See Also

Concepts

[Deploying Models \(Master Data Services\)](#)

Create a Model Deployment Package by Using MDSModelDeploy

SQL Server 2012

In Master Data Services, use the MDSModelDeploy tool to create a package. Depending on the commands you specify, the package can contain either:

- Model objects only.
- Model objects and data.

If you want to deploy a package that contains model objects only, you can use the model deployment wizard in the Master Data Manager web application instead. For more information, see [Create a Model Deployment Package by Using the Wizard](#).

Prerequisites

To perform this procedure:

1. The basic permissions required to run the MDSModelDeploy tool are the following:
 - The same Windows permission as the MDS Configuration Manager (administrator of Windows)
 - DBA permission on the MDS database.
2. The permissions required to create the package using the MDSModelDeploy tool are the following:
 - MDS model administrator permission on the data model.
 - MDS Integration Management function permission.
3. The permissions required to deploy a model using the MDSModelDeploy tool are the following:
 - MDS Explorer function permission
 - MDS Integration Management function permission
 - MDS System Administration function permission.
4. The permissions required to list models using the MDSModelDeploy tool are the following:
 - MDS Explorer function permission
 - MDS model administrator permission on the data model on order to see the model in the list.

A model must exist for you to create a package of. For more information, see [Create a Model \(Master Data Services\)](#).

For more information, see [Administrators \(Master Data Services\)](#).

To create a model deployment package by using MDSModelDeploy

1. Open a command prompt.
2. Navigate to the location of MDSModelDeploy.exe.
 - If MDS was installed in the default location, the file is in `drive:\Program Files\Microsoft SQL Server\110\Master Data Services\Configuration`.
 - If MDS was not installed in the default location, search the local computer for MDSModelDeploy.exe.
3. Optional. View options and help.
 - To display all available options, type `MDSModel Deploy` and press Enter.
 - To display help for an option, type the following, where *OptionName* is the name of the option: `MDSModel Deploy help OptionName`.
4. Optional. If you have multiple web applications, determine the name of the service you will deploy to by typing this command and pressing Enter:

```
MDSModel Deploy listservices
```

A list of values is returned, for example `MDS1, Default Web Site, MDS`. The first value in this list (in this case, `MDS1`) is needed to deploy the model.

5. To create a package that contains model objects and data, type the following, where *ModelName*, *VersionName*, *ServiceName*, and *PackageName* are the names of the model, version, service, and of the .pkg output file respectively:

```
MDSModelDeploy createpackage -model ModelName -version VersionName -service ServiceName -package PackageName -include
```

If you do not want to include data, do not use the `-version` and `-includedata` switches.

6. Press Enter. When the package is successfully created, a message stating "MDSModelDeploy operation completed successfully" is displayed.

Next Steps

- [Deploy a Model Deployment Package by Using MDSModelDeploy](#)

See Also

Reference

[Model Deployment Options \(Master Data Services\)](#)

Concepts

[Deploying Models \(Master Data Services\)](#)

Model Deployment Options (Master Data Services)

SQL Server 2012

In Master Data Services, when you deploy a model package file, you must decide whether to deploy a new or cloned model, or to update a model that was previously cloned.

Workflows

When working with model packages, there are two primary workflows.

- Create a package of a model in a test environment and deploy a clone of the model to a production environment. Over time, deploy updates from the test environment to the production environment.
- Create a package of a model and deploy it as a new model to the same environment. In this case, you must give the model a new name.

Options

In the MDS database, each model object has a unique identifier (ID). These IDs are included in model deployment packages. When you deploy the package, you must choose what to do with these IDs.

The following table should help you determine which choice to make when deploying a model by using either the System Administration model deployment wizard or the MDSModelDeploy tool.

Option	Description	Notes
New	Create a new model with a unique name. New identifiers are created for all model objects.	If you create a new model with new identifiers, you cannot use model deployment tools to apply updates to the model later. When using the wizard in the web application to deploy a model package, you have the option to create a new model only if a model with the same name or ID already exists.
Clone	Create a new model that is an exact clone of the model in the package. This works only if the model does not exist (by name or identifier) in the target environment. Use "clone" when you want to have the same model in multiple environments and update the cloned model over time.	This is the default behavior of the wizard in the web application. If a model with the same name or ID already exists, you are prompted to create a new model instead.
Update	Update an existing model with the model in the package. The identifiers must be the same in both models. This is used to update a model that you previously cloned.	You can only update models that were previously cloned. (The names and IDs must match.)

See Also

Tasks

[Deploy a Model Deployment Package by Using MDSModelDeploy](#)

[Deploy a Model Deployment Package by Using the Wizard](#)

Concepts

[Deploying Models \(Master Data Services\)](#)

Deploy a Model Deployment Package by Using the Wizard

SQL Server 2012

Use the Master Data Manager model deployment wizard to deploy packages that contain model objects only. If you need to deploy a package with data, see [Deploy a Model Deployment Package by Using MDSModelDeploy](#).

⚠ Important

Packages can be deployed to the edition of SQL Server they were created in only. This means that packages created in SQL Server 2008 R2 cannot be deployed to SQL Server 2012.

▣ Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area in the target Master Data Services environment.
- A model deployment package must exist. For more information, see [Create a Model Deployment Package by Using the Wizard](#).
- You must be an administrator in the environment where you are deploying the model. For more information, see [Administrators \(Master Data Services\)](#).

To deploy a model deployment package of model objects only

1. In Master Data Manager, click **System Administration**.
2. On the **Model View** page, from the menu bar, point to **System** and click **Deployment**.
3. On the **Model Deployment Wizard**, click **Deploy**.
4. Click **Browse**.
5. Find your deployment package (.pkg file) and click **Open**.
6. Click **Next**.
7. After the package is loaded, click **Next**.
8. If the model already exists, you can update it by selecting **Update the existing model**. To create a new model, select **Create a new model** and after you click **Next** you can type a name for the new model.
9. Click **Finish** to exit the wizard.

Notes:

- If a subscription view in the package has the same name as a subscription view in an existing model, the view is created as *modelname.subscriptionviewname*. If this name is already in use, the subscription view is not created.
- The deployment process has four steps:
 1. The model objects are created.
 2. Subscription views are created.
 3. Business rules are created.
 4. Master data is populated.
- When creating a new or cloned model, if the process fails during any step, the model is deleted.

When updating a model, if the process fails during any of the first three steps, it does not proceed beyond that step; however, changes that are already made are not rolled back. If the process fails in step 4, members that can be updated are updated.

▣ Next Steps

User-defined metadata, file attributes, and user and group permissions are not included in model deployment packages. After you deploy a model, you must update these manually. For more information, see:

- [Add Metadata \(Master Data Services\)](#)
- [Assign Model Object Permissions \(Master Data Services\)](#)

▣ See Also

Concepts

Deploying Models (Master Data Services)

© 2014 Microsoft. All rights reserved.

Deploy a Model Deployment Package by Using MDSModelDeploy

SQL Server 2012

In Master Data Services, use the MDSModelDeploy tool to deploy a package that contains either:

- Model objects only.
- Model objects and data.

If you want to deploy a package that contains model objects only, you can use the model deployment wizard in the Master Data Manager web application instead. For more information, see [Deploy a Model Deployment Package by Using the Wizard](#).

⚠ Important

Packages can be deployed to the edition of SQL Server they were created in only. This means that packages created in SQL Server 2008 R2 cannot be deployed to SQL Server 2012.

▣ Prerequisites

To perform this procedure:

- You must have permission to access the **System Administration** functional area in the target Master Data Services environment.
- A model deployment package must exist. For more information, see [Create a Model Deployment Package by Using MDSModelDeploy](#).
- You must be an administrator in the environment where you are deploying the model. For more information, see [Administrators \(Master Data Services\)](#).
- If you are updating a model with data, the version you're deploying to cannot be **Locked** or **Committed**.

To deploy a model deployment package

1. Determine whether you are deploying a new model, a clone of a model, or updating a previously-cloned model. For more information, see [Model Deployment Options \(Master Data Services\)](#).
2. Open a command prompt and navigate to MDSModelDeploy.exe.
 - If MDS is installed at the default location, the tool is available at `drive:\Program Files\Microsoft SQL Server\110\Master Data Services\Configuration\MDSModelDeploy.exe`
 - If MDS is not installed at the default location, search the local computer for MDSModelDeploy.exe.
3. Optional. View options and help.
 - To display all available options, type `MDSModel Deploy` and press Enter.
 - To display help for an option, type the following, where *OptionName* is the name of the option: `MDSModel Deploy help OptionName`.
4. Optional. If you have multiple web applications, determine the name of the service you will deploy to by typing this command and pressing Enter:

```
MDSModelDeploy listservices
```

A list of values is returned, for example `MDS1`, `Default Web Site`, `MDS`. The first value in this list (in this case, `MDS1`) is needed to deploy the model.

5. Depending on whether you are creating a model, cloning a model, or updating a model, at the command prompt, type the following and press Enter.
 - To create a new model:

```
MDSModelDeploy deploynew -package PackageName -model ModelName -service ServiceName
```

- To create a clone of a model:

```
MDSModelDeploy deployclone -package PackageName
```

- To update an existing model and its data:

```
MDSModelDeploy deployupdate -package PackageName -version VersionName
```

Important

If you use the MDSModelDeploy tool to update an existing model and its data, and the package does not contain an entity, attribute, or member that exists in the destination model, MDSModelDeploy will not delete that entity, attribute, or member from the model.

Where *PackageName* is the name of the package (.pkg) file, *ModelName* is the name of the new model, *VersionName* is the name of the version, and *ServiceName* is the name of the service that you returned in the previous step. Ensure that the model and version names match the exact case-sensitive names.

- When the package is successfully deployed, a message stating "MDSModelDeploy operation completed successfully" is displayed.

Notes:

- If a subscription view in the package has the same name as a subscription view in an existing model, the view is created as *modelName.subscriptionviewname*. If this name is already in use, the subscription view is not created.
- The deployment process has four steps:
 - The model objects are created.
 - Business rules are created.
 - Subscription views are created.
 - Master data is populated.
- When creating a new or cloned model, if the process fails during any step, the model is deleted.

When updating a model, if the process fails during the first three steps, it does not proceed; however, changes that are already made are not rolled back. If the process fails in step 4, members that can be updated are updated.

Next Steps

User-defined metadata, file attributes, and user and group permissions are not included in model deployment packages. After you deploy a model, you must update these manually. For more information, see:

- [Add Metadata \(Master Data Services\)](#)
- [Assign Model Object Permissions \(Master Data Services\)](#)

See Also

Concepts

[Deploying Models \(Master Data Services\)](#)

Edit a Model Deployment Package

SQL Server 2012

This topic describes how to deploy selected parts of a model in MDS, rather than an entire model. To do so, you edit an MDS model package using the Model Package Editor.

The Model Package Editor wizard enables you to select the specific entities, derived hierarchies, subscription views, and business rules in a model that you want to include in an MDS package, and then later deploy. You can leave out those parts of the model that you do not want to deploy. When you select an entity, all of the dependent objects in that entity are also automatically selected.

You use the Model Package Editor to select parts of a model in a package file that was created by either the MDSModelDeploy tool (which creates a package file that includes objects and data) or the Model Deployment wizard (which creates a file that includes only the model structure). After editing the model in the package, you use the MDSModelDeploy tool to deploy objects and data, or the Model Deployment wizard to deploy just the model structure.

Prerequisites

To perform this procedure:

- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).
- A model package must exist for you to edit. For more information, see [Deploying Models \(Master Data Services\)](#) and [Create a Model Deployment Package by Using the Wizard](#) or [Create a Model Deployment Package by Using MDSModelDeploy](#).

To edit a model deployment package

1. In Windows Explorer on the MDS server, move to *drive:\Program Files\Microsoft SQL Server\110\Master Data Services\Configuration*.
2. Execute ModelPackageEditor.exe.
3. In the Model Package Editor wizard, click **Browse**, move to the folder containing your packages, select a package, and then click **Open**. Click **Next**.
4. Select those entities, derived hierarchies, subscriptions views, or business rules that you want to deploy. Deselect those that you do not want to deploy. Click **Next**.
5. Verify the list of selections to deploy. To change, click **Back** and repeat step 4.
6. Click **Browse**, move to the folder that you want to save the partial package in, and then enter the file name of the partial package (with a .pkg extension). Click **Save**.
7. Click **Finish**.

Next Steps

- [Deploy a Model Deployment Package by Using the Wizard](#)
- [Deploy a Model Deployment Package by Using MDSModelDeploy](#)

Security (Master Data Services)

SQL Server 2012

In Master Data Services, use security to ensure that users have access to the specific master data necessary to do their jobs, and to prevent them from accessing data that should not be available to them.

You can also use security to make someone an administrator of a specific model and functional area (for example, to allow someone to create versions of the Customer model or to give someone the ability to set security permissions).

Master Data Services security is based on local or Active Directory domain users and groups. MDS security allows you to use a granular level of detail when determining the data a user can access. Because of the granularity, security can easily become complicated and you should use caution when using overlapping users and groups. For more information, see [Overlapping User and Group Permissions \(Master Data Services\)](#).

You can assign security access in the **User and Group Permissions** functional area of the Master Data Manager web application or by using the web service.

Types of Users

There are two types of users in Master Data Services:

- Those who access data in the **Explorer** functional area.
- Those who have the ability to perform administrative tasks in areas other than **Explorer**. These users are called [Administrators \(Master Data Services\)](#).

How to Set Security

To give a user or group permission to access data or functionality in MDS, you must assign:

- [Functional area access](#), which determines which of the five functional areas of the user interface a user can access.
- [Model object permissions](#), which determine the attributes a user can access, and the type of access (Read or Update) that the user has to those attributes.
- Optionally, [hierarchy member permissions](#), which determine the members a user can access, and the type of access (Read or Update) the user has to those members.

When you assign permissions to attributes and members, the permissions intersect and rules determine which permission takes precedence. For more information, see [How Permissions Are Determined \(Master Data Services\)](#).

Security in the Add-in for Excel

Security set in the Master Data Manager web application is also applied to the Add-in for Excel. Users are only able to view and work with data they have permission to. Administrators can perform administrative tasks.

The only caveat is that all security assigned in Master Data Manager does not take effect in Excel until a 20 minute interval passes. The interval is defined by the *MdsMaximumUserInfoCacheInterval* setting in the web.config file. To change the interval, you can change the setting and restart IIS.

Related Tasks

Task Description	Topic
Create a user who has full permission to a model.	Create a Model Administrator (Master Data Services)
Add an Active Directory group to Master Data Services; this is the first step in giving a group permission to access data in the Master Data Services web application.	Add a Group (Master Data Services)
Assign permission to a functional area of the Master Data Services web application.	Assign Functional Area Permissions (Master Data Services)
Assign permission to attribute values by assigning permission to model objects.	Assign Model Object Permissions (Master Data Services)
Assign permission to member values by assigning permission to hierarchy nodes.	Assign Hierarchy Member Permissions (Master Data Services)

See Also

Concepts

[Administrators \(Master Data Services\)](#)

Users and Groups (Master Data Services)
Functional Area Permissions (Master Data Services)
Model Object Permissions (Master Data Services)
Hierarchy Member Permissions (Master Data Services)
How Permissions Are Determined (Master Data Services)

Administrators (Master Data Services)

SQL Server 2012

In Master Data Services, there are two types of administrators: model administrators, and the Master Data Services system administrator.

Model Administrators

In Master Data Services, a model administrator is a user who has **Update** permission assigned to the top-level model object on the **Model Objects** tab and no other assigned permissions.

- If the user has access to the **Explorer** functional area, the user can add, delete, and update all master data in this area.
- If the user has access to other functional areas, the user can perform all administrative tasks available in the functional area.

Each model can have multiple administrators. Each user can be a model administrator for one, several, or all models in the Master Data Services deployment.

A user can be configured as a model administrator either in Master Data Manager or programmatically. For more information, see [Create a Model Administrator \(Master Data Services\)](#).

Master Data Services System Administrator

There is only one Master Data Services system administrator. The system administrator is the user specified for the **Administrator Account** when you create the Master Data Services database.

The Master Data Services system administrator:

- Automatically has access to all functional areas.
- Can add, delete, and update all master data for all models in the **Explorer** functional area.

You can change the user who is assigned as the system administrator. For more information, see [Change the System Administrator Account \(Master Data Services\)](#).

Comparing Administrator Types

Administrator Type	Description
Master Data Services system administrator	<ul style="list-style-type: none">• Permissions assigned in Master Data Manager have no effect on the administrator's access.• Automatically has Update permission to all models.• Automatically has access to all functional areas.• In <code>mdm.tblUser</code>, the value in the ID column is 1.
Model administrator	<ul style="list-style-type: none">• Permissions assigned in Master Data Manager determine whether or not the user is a model administrator.• Can be a model administrator based on permissions assigned explicitly or permissions inherited from a group.• Is an administrator only for models that have Update permission assigned to top-level model object, and no other permissions.• Has access only to functional areas that access is granted to.• In <code>mdm.tblUser</code>, the value in the ID column is not 1.

See Also

Tasks

[Create a Model Administrator \(Master Data Services\)](#)

[Change the System Administrator Account \(Master Data Services\)](#)

[Create a Master Data Services Database](#)

Concepts

[Notifications \(Master Data Services\)](#)

Create a Model Administrator (Master Data Services)

SQL Server 2012

In Master Data Services, create a model administrator when you want a group or user to have **Update** permission to all objects in one or more models.

Tip

To simplify administration, create a Windows or local group and configure it as a model administrator. You can then add and remove users from the group without accessing Master Data Manager.

Prerequisites

To perform this procedure:

- You must have permission to access the **User and Group Permissions** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To create a model administrator

1. In Master Data Manager, click **User and Group Permissions**.
2. On the **Users** or **Groups** page, select the row for the user or group that you want to edit.
3. Click **Edit selected user**.
4. Click the **Models** tab.
5. Optionally, select a model from the **Model** list.
6. Click **Edit**.
7. Click the model you want to grant permission to.
8. From the menu, select **Update**.
9. Complete steps 7 and 8 for each model you want the group or user to be an administrator for.
10. Click **Save**.

Remarks

Do not assign any other permissions to model objects or hierarchy members. If you do, the user is no longer an administrator and cannot view the model in any functional area other than **Explorer**.

There is one exception: if the user has **Update** permission assigned to a hierarchy **Root** on the **Hierarchy Members** tab, the user is still considered a model administrator.

See Also

Tasks

[Assign Model Object Permissions \(Master Data Services\)](#)
[Assign Hierarchy Member Permissions \(Master Data Services\)](#)

Concepts

[Administrators \(Master Data Services\)](#)
[Model Object Permissions \(Master Data Services\)](#)
[Hierarchy Member Permissions \(Master Data Services\)](#)

Change the System Administrator Account (Master Data Services)

SQL Server 2012

You can change the user account that is designated as the Master Data Services system administrator.

⚠ Caution

When you complete this procedure, the former system administrator user account is deleted.

▴ Prerequisites

To perform this procedure:

- You must add the new administrator's user name to the Master Data Manager Users list. For more information, see [Add a User \(Master Data Services\)](#).
- You must have permission to view `mdm.tblUser` and to execute the `mdm.udpSecurityMemberProcessRebuildModel` stored procedure in the Master Data Services database. For more information, see [Database Object Security \(Master Data Services\)](#).

To change the administrator account

1. Open SQL Server Management Studio and connect to the Database Engine instance for your Master Data Services database.
2. In `mdm.tblUser`, find the user that will be the new administrator and copy the value in the **SID** column.
3. Create a new query.
4. Type the following text, replacing `DOMAIN\user_name` with the new administrator's user name and `SID` with the value you copied in step 2.

```
EXEC [mdm].[udpSecuritySetAdministrator] @UserName='DOMAIN\user_name', @SID = 'SID', @PromoteNonAdmin = 1
```

5. Run the query.

▴ See Also

Concepts

[Administrators \(Master Data Services\)](#)

Users and Groups (Master Data Services)

SQL Server 2012

To access the Master Data Manager web application a user must have a Windows domain account or an account on the server computer where Master Data Services is installed. To grant access to Master Data Manager you can either:

- Add the user account to a domain or local group and then add the group to the list of groups in Master Data Manager.
- Add the user account to the list of users in Master Data Manager.

Note

When a user belongs to a group that has access to Master Data Manager, the user's name is automatically added to the list of users the first time the user accesses Master Data Manager or the MDS Add-in for Excel.

To take action within the **Explorer** functional area of the UI, the group or user must be assigned access to the **Explorer** functional area and assigned permission to model objects.

If a user or group needs access to other functional areas, the user or group must be an administrator. For more information, see [Administrators \(Master Data Services\)](#).

Best Practice

To simplify administration, create groups and assign each group permission to functional areas and model objects. You can then add and remove users from the groups without accessing the Master Data Manager UI.

Do not assign additional permissions to an individual user, and do not include a user in multiple groups that have access to Master Data Manager. In addition, do not use hierarchy member permissions unless you want a group to have limited access to specific members.

See Also

Tasks

[Add a User \(Master Data Services\)](#)

[Add a Group \(Master Data Services\)](#)

[Delete Users or Groups \(Master Data Services\)](#)

[Test a User's Permissions \(Master Data Services\)](#)

Add a User (Master Data Services)

SQL Server 2012

Add a user to the **Users** list in Master Data Manager to begin the process of assigning permission to the Web application. Before a user in the list can access Master Data Manager, you must give the user permission to one or more functional areas and model objects.

Prerequisites

To perform this procedure:

- You must have permission to access the **Users and Group Permissions** functional area.

To add a user

1. In Master Data Manager, click **User and Group Permissions**.
2. On the **Users** page, click **Add users**.
3. Type the user's name preceded by the Active Directory domain name or by the server computer's name, as in *domain\user_name* or *computer\user_name*.
4. Optionally, click **Check names**.
5. Click **OK**.

Next Steps

- [Assign Functional Area Permissions \(Master Data Services\)](#)

See Also

Concepts

[Security \(Master Data Services\)](#)

Add a Group (Master Data Services)

SQL Server 2012

Add a group to the **Groups** list in Master Data Manager to begin the process of assigning permission to the Web application. Before a user in the group can access Master Data Manager, you must give the group permission to one or more functional areas and model objects.

Prerequisites

To perform this procedure:

- You must have permission to access the **Users and Group Permissions** functional area.

To add a group

- In Master Data Manager, click **User and Group Permissions**.
- On the **Users** page, from the menu bar, click **Manage Groups**.
- Click **Add groups**.
- Type the group's name preceded by the Active Directory domain name or by the server computer's name, as in *domain\group_name* or *computer\group_name*.
- Optionally, click **Check names**.
- Click **OK**.

Note

When the user first accesses Master Data Manager, the user's name is added to the Master Data Manager list of users.

Next Steps

- [Assign Functional Area Permissions \(Master Data Services\)](#)

See Also

Concepts

[Security \(Master Data Services\)](#)

Change a User's Email Address (Master Data Services)

SQL Server 2012

In Master Data Services, you can change the email address that notifications are sent to, and the default format of notification emails.

Prerequisites

To perform this procedure:

- You must have permission to access the **Users and Groups** functional area.

To change a user's email address or default email format

1. In Master Data Manager, click **User and Group Permissions**.
2. On the **Users** page, select the row for the user that you want to edit.
3. Click **Edit selected user**.
4. Above the **General** tab, click **Edit**.
5. Update the email address or email format.
6. Click **Save**.

Next Steps

- [Assign Functional Area Permissions \(Master Data Services\)](#)

See Also

Concepts

[Security \(Master Data Services\)](#)

Delete Users or Groups (Master Data Services)

SQL Server 2012

Delete users or groups when you no longer want them to access Master Data Manager.

Note the following behavior when deleting users and groups:

- If you delete a user who is a member of a group that has access to Master Data Manager, then the user can still access Master Data Manager until you remove the user from the Active Directory or local group.
- If you delete a group, all users from the group who have accessed Master Data Manager are displayed in the **Users** list until you delete them.
- Changes to security are not propagated to the MDS Add-in for Excel for 20 minutes.

Prerequisites

To perform this procedure:

- You must have permission to access the **Users and Group Permissions** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete users or groups

1. In Master Data Manager, click **User and Group Permissions**.
2. To delete a user, remain on the **Users** page. To delete a group, from the menu bar, click **Manage Groups**.
3. In the grid, select the row for the user or group that you want to delete.
4. Click **Delete selected user** or **Delete selected group**.
5. On the confirmation dialog box, click **OK**.

See Also

Concepts

[Security \(Master Data Services\)](#)

Functional Area Permissions (Master Data Services)

SQL Server 2012

You can assign permission to each of the functional areas of the Master Data Manager user interface (UI). There are five functional areas:

- **Explorer**
- **Version Management**
- **Integration Management**
- **System Administration**
- **User and Group Permissions**

When you assign permission to a functional area, you are making an area of the UI visible to the user or group.

Within the **Explorer** functional area, additional permissions assigned to model objects and hierarchy members determine which data a user can access. Within all other functional areas, a user must be a model administrator to view a model and act on it. For more information, see [Administrators \(Master Data Services\)](#).

A user or group must have permission to at least one functional area and one model on the **Models** tab in order to access Master Data Manager.

See Also

Tasks

[Assign Functional Area Permissions \(Master Data Services\)](#)

Concepts

[Model Object Permissions \(Master Data Services\)](#)

[Hierarchy Member Permissions \(Master Data Services\)](#)

[How Permissions Are Determined \(Master Data Services\)](#)

Assign Functional Area Permissions (Master Data Services)

SQL Server 2012

Assign functional area permission to grant users or groups access to the functional areas of Master Data Manager.

Prerequisites

To perform this procedure:

- You must have permission to access the **Users and Group Permissions** functional area.

To assign functional area permissions

1. In Master Data Manager, click **User and Group Permissions**.
2. On the **Users** or **Groups** page, select the row for the user or group that you want to edit.
3. Click **Edit selected user**.
4. Click the **Functions** tab.
5. Click **Edit**.
6. Click a functional area and click the **Add** arrow.
7. When you are done, click **Save**.

Next Steps

- [Assign Model Object Permissions \(Master Data Services\)](#)

See Also

Tasks

[Create a Model Administrator \(Master Data Services\)](#)

Concepts

[Security \(Master Data Services\)](#)

Model Object Permissions (Master Data Services)

SQL Server 2012

Model object permissions are mandatory. They determine the attributes a user can access in the **Explorer** functional area of the UI.

For example, if you assign a user **Update** permission to the Product entity, the user can update all of the attributes of the Product entity. If you assign **Update** permission to a single attribute, the user can update that attribute only.

To determine security assigned on each individual attribute value, model object permissions are combined with hierarchy member permissions, which determine the members a user can access.

To give a user access to a functional area other than **Explorer**, the user must be a model administrator, which also involves assigning model object permissions. For more information, see [Administrators \(Master Data Services\)](#).

Model object permissions are assigned in the Master Data Manager user interface (UI), in the **User and Group Permissions** functional area on the **Models** tab. On this tab, the model is represented as a tree structure. When you assign permission to an object in the tree, all objects below inherit that permission. You can override that inheritance by assigning permission to individual objects.

You can assign **Read-only**, **Update**, or **Deny** permission to model objects. If you do not assign any permissions on the **Models** tab, the user cannot view any models or data in Master Data Manager.

▲ Best Practice

In general, you should assign **Update** permission to the model object, and then explicitly assign permission to objects underneath. If you do not assign permission to objects underneath, the permissions are inherited and the user is an administrator.

▲ See Also

Tasks

[Assign Model Object Permissions \(Master Data Services\)](#)

Concepts

[Model Permissions \(Master Data Services\)](#)

[Functional Area Permissions \(Master Data Services\)](#)

[Hierarchy Member Permissions \(Master Data Services\)](#)

[How Permissions Are Determined \(Master Data Services\)](#)

Assign Model Object Permissions (Master Data Services)

SQL Server 2012

In Master Data Services, assign permissions to model objects when you need to give a user or group access to data in the **Explorer** functional area of Master Data Manager, or when you need to make a user or group an administrator.

Note

When you assign permission to a model, permission to all other models is implicitly denied. If you do not assign model object permissions, the user or group cannot access any data in **Explorer**.

Prerequisites

To perform this procedure:

- You must have permission to access the **Users and Group Permissions** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To assign model object permissions

1. In Master Data Manager, click **User and Group Permissions**.
2. On the **Users** or **Groups** page, select the row for the user or group that you want to edit.
3. Click **Edit selected user**.
4. Click the **Models** tab.
5. Optionally, select a model from the **Model** list.
6. Click **Edit**.
7. Expand the tree, and click the model object you want to assign permissions to.
8. From the menu, select **Read-only**, **Update**, or **Deny**.
9. Click **Save**.

Next Steps

- (Optional) [Assign Hierarchy Member Permissions \(Master Data Services\)](#)

See Also

Tasks

[Delete Model Object Permissions \(Master Data Services\)](#)

[Create a Model Administrator \(Master Data Services\)](#)

Concepts

[Model Object Permissions \(Master Data Services\)](#)

Delete Model Object Permissions (Master Data Services)

SQL Server 2012

In Master Data Services, delete model object permissions to remove any assignments that have been made.

Prerequisites

To perform this procedure:

- You must have permission to access the **Users and Group Permissions** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete model object permissions

1. In Master Data Manager, click **User and Group Permissions**.
2. On the **Users** or **Groups** page, select the row for the user or group that you want to edit.
3. Click **Edit selected user**.
4. Click the **Models** tab.
5. Optionally, select a model from the **Model** list.
6. In the **Model Permission Summary** pane, select the row for the permission that you want to delete.
7. Click **Delete selected permission**.

Note

You cannot remove a permission from a user if the permission is inherited from a group. You must remove the permission from the group instead.

See Also

Tasks

[Assign Model Object Permissions \(Master Data Services\)](#)


Concepts

[Model Object Permissions \(Master Data Services\)](#)

Model Permissions (Master Data Services)

SQL Server 2012

Model permissions apply to all entities, derived hierarchies, explicit hierarchies, and collections that exist within the model. Permissions assigned to the model can be overridden for any individual object.

 Note	
If a user is a model administrator, the model is displayed in all functional areas of the user interface. Otherwise, the model is displayed in the Explorer functional area only. For more information, see Administrators (Master Data Services) .	
Permission	Description
Read-only	In Explorer , the model is displayed but the user cannot add or remove members, and cannot update attribute values, hierarchy memberships, or collection memberships.
Update	In Explorer , the model is displayed and the user can add and remove members, can update attribute values, hierarchy memberships, and collection memberships.
Deny	The model is not displayed.

When you assign permission to a model, the user gets access to all versions of the model. You cannot assign permission to an individual version.

See Also

Tasks

[Assign Model Object Permissions \(Master Data Services\)](#)

Concepts

[Model Object Permissions \(Master Data Services\)](#)

[Entity Permissions \(Master Data Services\)](#)

[Collection Permissions \(Master Data Services\)](#)

Entity Permissions (Master Data Services)

SQL Server 2012

Entity permissions apply to:

- All of the entity's attributes, including **Name** and **Code**, for both leaf and consolidated members.
- All of the entity's collections.
- Explicit hierarchy memberships and relationships.

When you have permission to an entity, you can add and remove members from the entity, its explicit hierarchies, and its collections.

Note	
These permissions apply to the Explorer functional area of the user interface only.	
Permission	Description
Read-only	The entity is displayed but the user cannot add, remove, or change members.
Update	The entity is displayed and the user can add, remove, and change members.
Deny	The entity is not displayed.

See Also

Tasks

[Assign Model Object Permissions \(Master Data Services\)](#)

Concepts

[Model Object Permissions \(Master Data Services\)](#)

[Entities \(Master Data Services\)](#)

Leaf Permissions (Master Data Services)

SQL Server 2012

Leaf permissions apply to the attribute values for all leaf members of an entity.

For entities without explicit hierarchies enabled, assigning permission to **Leaf** is the same as assigning permission to the entity.


Notes:

- Leaf permissions apply to the **Explorer** functional area of the user interface only.
- Permissions assigned to **Name** and **Code** attributes are not enforced.

Permission	Description
Read-only	Leaf members are displayed but the user cannot add, remove, or change them. If consolidated members exist, the names and codes are displayed but the user cannot add, remove, or change them.
Update	Leaf members are displayed and the user can add, remove, and change them. If consolidated members exist, the names and codes are displayed but the user cannot add, remove, or change them.
Deny	Leaf members for the entity are not displayed.

Attribute Permissions

Attribute permissions apply to the attribute's values for the specific entity. Users with attribute permissions only cannot add or remove members.


Permission	Description
Read-only	The attribute is displayed but the user cannot change attribute values.
Update	The attribute is displayed and the user can change attribute values.
Deny	<div>The attribute is not displayed.</div> <div> Note</div> <div>You cannot explicitly deny access to Name and Code attributes.</div>

Example

For the Product entity, assign **Update** permission to Subcategory attribute. Deny permission to all other attributes.

Name	Code	Subcategory (Update)
Mountain-100	BK-M101	{5} Mountain Bikes
Mountain-100	BK-M201	{5} Mountain Bikes

In **Explorer**, you can update any attribute value in the Subcategory column. If you do not have permission to an attribute, the attribute is not displayed.

 Note
In this example, Subcategory is a domain-based attribute, based on the SubcategoryList entity. You can select a different subcategory for Mountain-100 but you cannot add members to or delete members from the SubcategoryList entity.

See Also

Tasks

[Assign Model Object Permissions \(Master Data Services\)](#)

Concepts

[Consolidated Permissions \(Master Data Services\)](#)

[Model Object Permissions \(Master Data Services\)](#)

[Members \(Master Data Services\)](#)

[Attributes \(Master Data Services\)](#)

Consolidated Permissions (Master Data Services)

SQL Server 2012

Consolidated permissions apply to the attribute values for all consolidated members of an entity.

Consolidated permissions apply only to entities that are enabled for explicit hierarchies and collections.


Notes:

- Leaf permissions apply to the **Explorer** functional area of the user interface only.
- Permissions assigned to **Name** and **Code** attributes are not enforced.

Permission	Description
Read-only	Consolidated members are displayed but the user cannot add, remove, or change them.
Update	Consolidated members are displayed and the user can add, remove, and change them.
Deny	Consolidated members for the entity are not displayed.

Attribute Permissions

Attribute permissions apply to the attribute's values for the specific entity. Users with only attribute permissions cannot add or remove members.

Permission	Description
Read-only	The attribute is displayed but the user cannot change attribute values.
Update	The attribute is displayed and the user can change attribute values.
Deny	<div>The attribute is not displayed.<div> Note You cannot explicitly deny access to Name and Code attributes.</div></div>

See Also

Tasks

[Assign Model Object Permissions \(Master Data Services\)](#)

Concepts

[Leaf Permissions \(Master Data Services\)](#)

[Model Object Permissions \(Master Data Services\)](#)


[Members \(Master Data Services\)](#)

[Attributes \(Master Data Services\)](#)

Collection Permissions (Master Data Services)

SQL Server 2012

Collection permissions apply to all collections for an entity. You cannot give permission to a specific collection; permissions apply to all collections.

 Note	
These permissions apply to the Explorer functional area of the user interface only.	
Permission	Description
Read-only	All collections and collection members are displayed. The user cannot add or remove collections, and cannot add members to the collection or remove members from it.
Update	All collections are displayed and the user can add and remove collections. The user can also add and remove members from the collection.
Deny	All collections for the entity are not displayed.

See Also

Tasks

[Assign Model Object Permissions \(Master Data Services\)](#)

Concepts

[Collections \(Master Data Services\)](#)

[Model Object Permissions \(Master Data Services\)](#)

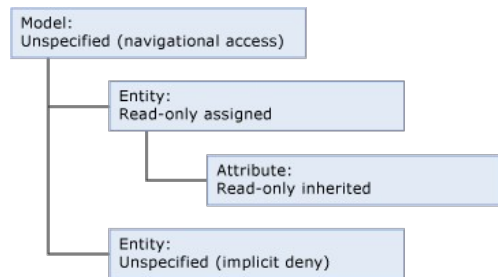
Navigational Access (Master Data Services)

SQL Server 2012

Navigational access applies to model object security, which is assigned on the **Models** tab.

Navigational access is the access you get to levels higher than where you've assigned security.

In this example, permissions are assigned to an entity, and so navigational access is granted at the model level.



Entities

When you assign permission to an entity, its leaf members, or its consolidated members, navigational access means you can read or update the name and code for all members. You can also read the model name.

Attributes

When you assign permission to an attribute, navigational access means you can read or update the name and code for all members in the entity. You can also read the model name.

Collections

When you assign permissions to collections, you can read or update the name, code, description and owner ID. You can also read the model name.

See Also

Concepts

[How Permissions Are Determined \(Master Data Services\)](#)

Hierarchy Member Permissions (Master Data Services)

SQL Server 2012

Hierarchy member permissions are optional and should be used only when you want a user to have limited access to specific members. If you do not assign permissions on the **Hierarchy Members** tab, then the user's permissions are based solely on the permissions assigned on the **Models** tab.

Hierarchy member permissions are assigned in the Master Data Manager user interface (UI), in the **User and Group Permissions** functional area on the **Hierarchy Members** tab. These permissions determine which members a user can access in the **Explorer** functional area of the UI.

On the **Hierarchy Members** tab, each hierarchy is represented as a tree structure. When you assign permission to a node in the tree, all children inherit that permission unless permission is explicitly assigned at a lower level.

Note

When you assign permission to a node in a hierarchy, all members in other nodes at the same level or higher are implicitly denied.

In **Explorer**, the member permissions are applied everywhere the member is displayed. For example, a member with **Read-only** permission is read-only in any entities, hierarchies, and collections to which it belongs.

Hierarchy member permissions apply to the model version you assign them to, and to any future copies of the version. They do not apply to versions earlier than the one you assign them to.

Permission	Description
Read-only	The members are displayed but the user cannot change them. The user also cannot move the members in any explicit hierarchies or collections that the members belong to.
	<div><div>Note</div><p>If you assign Read-only permission to Root, the members under Root are read-only; however, in explicit hierarchies and collections, the user can move members to Root and can add new members to Root.</p></div>
Update	The members are displayed and the user can change them. The user can also move the members in any explicit hierarchies or collections the members belong to.
Deny	The members are not displayed.

On the **Hierarchy Members** tab, the permissions you assign do not take effect immediately. The frequency that the permissions are applied depends on the **Member security processing interval setting** in the System Settings table in the Master Data Services database. You can apply member permissions immediately by following the steps in [Immediately Apply Member Permissions \(Master Data Services\)](#).

Note

You cannot assign hierarchy member permissions to recursive hierarchies, derived hierarchies with explicit caps, and derived hierarchies with hidden levels.

Note

In SQL Server 2012, when a user has no permissions for a leaf member of a derived hierarchy, the user will not be able to see leaf members under the invisible leaf member in the Hierarchy Explorer, even when they have Update or Read-only permissions for those members. This occurs when the user does not have explicit permissions on the Root. However, the user can see the leaf member in the Attribute Explorer or by searching for it in the search box of the Hierarchy Explorer. In SQL Server 2008 R2, the leaf under the invisible leaf will be displayed, and leaves at the same level as that leaf will be shown as duplicates.

Possible Overlapping Permissions

When assigning permission to members, you may have to resolve overlapping permissions.

When a member belongs to multiple hierarchies

Two or more hierarchies can contain the same member.

- If one hierarchy node is assigned **Update** permission and another is assigned **Read-only**, then the members in the node are **Read-only**.
- If one hierarchy node is assigned **Update** or **Read-only** permission and another node is assigned **Deny**, then the members in the node are not displayed.

See Also

Tasks

[Assign Hierarchy Member Permissions \(Master Data Services\)](#)

[Immediately Apply Member Permissions \(Master Data Services\)](#)

Concepts

[How Permissions Are Determined \(Master Data Services\)](#)

[Members \(Master Data Services\)](#)

[Hierarchies \(Master Data Services\)](#)

Assign Hierarchy Member Permissions (Master Data Services)

SQL Server 2012

Assign permissions to hierarchy members to give users or groups access to view data in the **Explorer** functional area of Master Data Manager.

Hierarchy member permissions are optional. They provide added granularity to model object permissions, which are required.


Prerequisites

To perform this procedure:

- You must have permission to access the **Users and Group Permissions** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To assign hierarchy member permissions

1. In Master Data Manager, click **User and Group Permissions**.
2. On the **Users** or **Groups** page, select the row for the user or group that you want to edit.
3. Click **Edit selected user**.
4. Click the **Hierarchy Members** tab.
5. From the **Model** list, select a model.
6. From the **Version** list, select a version.
7. From the **Hierarchy** list, select a hierarchy.
8. Click **Edit**.
9. Expand the tree, and click the hierarchy node you want to assign permissions to.
10. From the menu, select **Read-only**, **Update**, or **Deny**.
11. Click **Save**.

 **Note**

Hierarchy member permissions do not take effect immediately. See [Immediately Apply Member Permissions \(Master Data Services\)](#) for more information.

See Also

Tasks

[Delete Hierarchy Member Permissions \(Master Data Services\)](#)
[Assign Model Object Permissions \(Master Data Services\)](#)

Concepts

[Hierarchy Member Permissions \(Master Data Services\)](#)

Immediately Apply Member Permissions (Master Data Services)

SQL Server 2012

In Master Data Services, instead of waiting for member security to be applied at regular intervals, you can apply member permissions immediately.

Prerequisites

To perform this procedure:

- You must have permission to execute the `mdm.udpSecurityMemberProcessRebuildModel` stored procedure in the Master Data Services database. For more information, see [Database Object Security \(Master Data Services\)](#).

To immediately apply hierarchy member permissions

- Open SQL Server Management Studio and connect to the Database Engine instance for your Master Data Services database.
- Create a new query.
- Type the following text, replacing *database* with the name of your database and *Model_Name* with the name of the model.

```
USE [database];

GO

DECLARE @Model_ID INT;
SELECT @Model_ID = ID FROM mdm.tblModel WHERE [Name] = N'Model_Name';
EXEC [mdm].[udpSecurityMemberProcessRebuildModel] @Model_ID=@Model_ID, @ProcessNow=1;
GO
```

- Run the query.

See Also

Tasks

[Assign Hierarchy Member Permissions \(Master Data Services\)](#)

Concepts

[Hierarchy Member Permissions \(Master Data Services\)](#)

Delete Hierarchy Member Permissions (Master Data Services)

SQL Server 2012

In Master Data Services, delete model object permissions to remove any assignments that have been made.

Prerequisites

To perform this procedure:

- You must have permission to access the **Users and Group Permissions** functional area.
- You must be a model administrator. For more information, see [Administrators \(Master Data Services\)](#).

To delete hierarchy member permissions

1. In Master Data Manager, click **User and Group Permissions**.
2. On the **Users** or **Groups** page, select the row for the user or group that you want to edit.
3. Click **Edit selected user**.
4. Click the **Hierarchy Members** tab.
5. From the **Model** list, select a model.
6. From the **Version** list, select a version.
7. In the **Hierarchy Member Permission Summary** pane, select the row for the permission that you want to delete.
8. Click **Delete selected permission**.

Note

You cannot remove a permission from a user if the permission is inherited from a group. You must remove the permission from the group instead.

See Also

Tasks

[Assign Hierarchy Member Permissions \(Master Data Services\)](#)

Concepts

[Hierarchy Member Permissions \(Master Data Services\)](#)

How Permissions Are Determined (Master Data Services)

SQL Server 2012

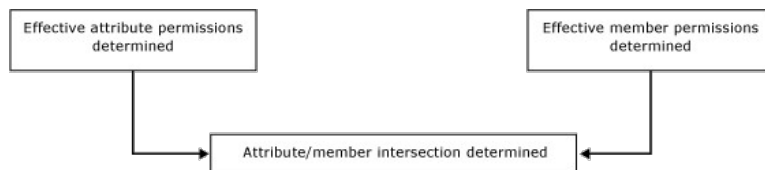
In Master Data Services, the simplest way to configure security is to assign model object permissions to a group that the user is a member of.

Security becomes more complex when:

- Both model object and hierarchy member permissions are assigned.
- The user belongs to groups and permission is assigned to both the user and groups.
- The user belongs to groups and permission is assigned to multiple groups.

Permissions assigned to a single group or user

If you assign permissions to a single group or user, permissions are determined based on the following workflow.

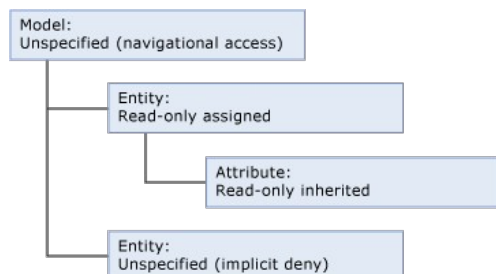


Step 1: Effective attribute permissions are determined.

The following list describes how effective attribute permissions are determined:

- Permissions assigned to model objects determine which attributes a user can access.
- All model objects automatically inherit permission from the closest object at a higher level in the model structure.
- Any objects at the same level as the entity are implicitly denied.
- Any objects at a higher level are given navigational access. For more information about navigational access, see [Navigational Access \(Master Data Services\)](#).

In this example, **Read-only** permission is assigned to an entity and that permission is inherited by its attribute, which is at a lower level in the model structure. The model provides navigational access to this entity and its attribute. The other entity in the model has no explicit permission assigned and does not inherit any permissions, so it is implicitly denied.

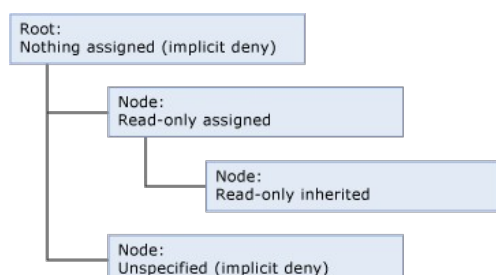


Step 2: If hierarchy member permissions are assigned, effective member permissions are determined.

The following list describes how effective hierarchy member permissions are determined:

- Permissions assigned to hierarchy nodes determine which members a user can access.
- All nodes in a hierarchy automatically inherit permission from the closest object at a higher level in the hierarchy structure.
- Any nodes at the same level are implicitly denied.
- Any nodes at higher levels that do not have permissions assigned are implicitly denied.

In this example, **Read-only** permission is assigned to one node of the hierarchy and that permission is inherited by a node at a lower level in the hierarchy structure. The root has no permission assigned, so it is implicitly denied. The other node in the hierarchy structure has no explicit permission assigned and does not inherit any permissions, so it is implicitly denied.

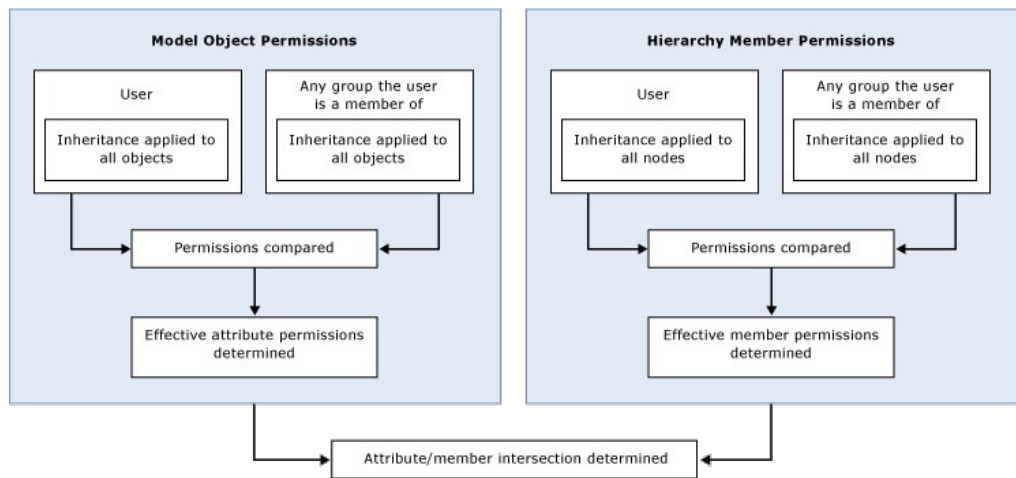


Step 3: The intersection of attribute and member permissions is determined.

If the effective attribute permissions are different than the effective member permissions, permissions must be determined for each individual attribute value. For more information, see [Overlapping Model and Member Permissions \(Master Data Services\)](#).

Permissions assigned to multiple groups

If a user belongs to one or more groups and permissions are assigned to both the user and the groups, the workflow becomes more complex.



In this case, overlapping user and group permissions must be resolved before model object and hierarchy member permissions can be compared. For more information, see [Overlapping User and Group Permissions \(Master Data Services\)](#).

See Also

Concepts

[Overlapping User and Group Permissions \(Master Data Services\)](#)

[Overlapping Model and Member Permissions \(Master Data Services\)](#)

Overlapping Model and Member Permissions (Master Data Services)

SQL Server 2012

Permission assigned to a member can overlap with permission assigned to a model object. When overlaps occur, the more restrictive permission takes effect.

If a member has permission that is different than its corresponding model object, the following rules apply:

- **Deny** overrides all other permissions.
- **Read-only** overrides **Update**.

The following image shows which permissions take effect on an individual attribute value when attribute permissions are different than member permissions.

	Attribute 1 (Update)	Attribute 1 (Read-only)	Attribute 3 (Deny)	Attribute 4 (unspecified)
Member 1 (Update)	Update	Read-only	Deny (not shown)	No access (not shown)
Member 2 (Read-only)	Read-only	Read-only	Deny (not shown)	No access (not shown)
Member 3 (Deny)	Deny (not shown)	Deny (not shown)	Deny (not shown)	No access (not shown)
Member 4 (unspecified)	No access (not shown)	No access (not shown)	No access (not shown)	No access (not shown)

Example 1

Attribute: Update	+	Member: Update	=	Effective: Update
----------------------	---	-------------------	---	----------------------

On the **Models** tab, the Product entity has **Update** permission assigned. All attributes in the entity inherit that permission.

On the **Hierarchy Members** tab, the Mountain Bikes subcategory node in a derived hierarchy has **Update** permission assigned.

Result: In **Explorer**, the user has **Update** permission to all attribute values for all members in the Mountain Bikes node. All other members and attributes are hidden.

Name	Code	Subcategory	StandardCost	ListPrice
Mountain-100	BK-M101	Mountain Bikes {5}	1912.15	3399.99
Mountain-200	BK-M201	Mountain Bikes {5}	1265.62	2319.99
Road-150	BK-R150	Road Bikes {6}	2171.29	3578.27
Road-450	BK-450	Road Bikes {6}	884.71	1457.99

Example 2

Attribute: Update	+	Member: Read-only	=	Effective: Read-only
----------------------	---	----------------------	---	-------------------------

On the **Models** tab, the Subcategory attribute has **Update** permission assigned.

On the **Hierarchy Members** tab, the Mountain Bikes subcategory node in a derived hierarchy is explicitly assigned **Read-only** permission.

Result: In **Explorer**, the user has **Read-only** permission to the Subcategory attribute values for the members in the Mountain Bikes node. All other members and attributes are hidden.

Name	Code	Subcategory	StandardCost	ListPrice
Mountain-100	BK-M101	Mountain Bikes {5}	1912.15	3399.99
Mountain-200	BK-M201	Mountain Bikes {5}	1265.62	2319.99
Road-150	BK-R150	Road Bikes {6}	2171.29	3578.27
Road-450	BK-450	Road Bikes {6}	884.71	1457.99

Example 3

Attribute: Read-only	+	Member: Update	=	Effective: Read-only
-------------------------	---	-------------------	---	-------------------------

On the **Models** tab, the Subcategory attribute has **Read-only** permission assigned.

On the **Hierarchy Members** tab, the Mountain Bikes subcategory in a derived hierarchy is explicitly assigned **Update** permission.

Result: In **Explorer**, the user has **Read-only** permission to the attribute values. All other members and attributes are hidden.

Name	Code	Subcategory	StandardCost	ListPrice
Mountain-100	BK-M101	Mountain Bikes {5}	1912.15	3399.99
Mountain-200	BK-M201	Mountain Bikes {5}	1265.62	2319.99
Road-150	BK-R150	Road Bikes {6}	2171.29	3578.27
Road-450	BK-450	Road Bikes {6}	884.71	1457.99

▴ See Also

Concepts

[How Permissions Are Determined \(Master Data Services\)](#)

[Overlapping User and Group Permissions \(Master Data Services\)](#)

Overlapping User and Group Permissions (Master Data Services)

SQL Server 2012

A user's permissions are based on:

- Permissions from group memberships.
- Permissions assigned explicitly to the user.

If a user is a member of multiple groups, and those groups have access to Master Data Manager, the following rules apply:

- **Deny** overrides all other permissions.
- **Update** overrides **Read-only**.

These rules apply to both the **Models** and **Hierarchy Members** tabs. Permissions are resolved for each tab and then combined. For more information, see [How Permissions Are Determined \(Master Data Services\)](#).

Note

You can view the resolution of user and group overlapping permissions in the user interface. Both the **Models** and **Hierarchy Members** tab have a drop-down list from which you can choose **Effective** to view effective permissions.

Example 1

User: Read-only	+	Group 1: Update	+	Group 2: Read-only	=	Effective: Update
--------------------	---	--------------------	---	-----------------------	---	----------------------

The user belongs to Group 1 and Group 2.

The user has **Read-only** permission to the Product entity.

Group 1 has **Update** permission to the Product entity.

Group 2 has **Read-only** permission to the Product entity.

Result: The user's effective permission is **Update** to the Product entity.

Example 2

User: Read-only	+	Group 1: Update	+	Group 2: Deny	=	Effective: Deny
--------------------	---	--------------------	---	------------------	---	--------------------

The user belongs to Group 1 and Group 2.

The user has **Read-only** permission to the Product entity.

Group 1 has **Update** permission to Product entity.

Group 2 has **Deny** permission to the Product entity.

Result: The user's effective permission is **Deny** to the Product entity.

Example 3

User: Update	+	Group 1: Read-only	+	Group 2: Read-only	=	Effective: Update
-----------------	---	-----------------------	---	-----------------------	---	----------------------

The user belongs to Group 1 and Group 2.

The user has **Update** permission to a group of members in a hierarchy node.

Group 1 has **Read-only** permission to a group of members in a hierarchy node.

Group 2 has **Read-only** permission to a group of members in a hierarchy node.

Result: The user's effective permission is **Update** to the members.

See Also

Concepts

[How Permissions Are Determined \(Master Data Services\)](#)

Test a User's Permissions (Master Data Services)

SQL Server 2012

In Master Data Services, you can create a test user and log into the web application to test permissions. When a user attempts to access the Master Data Manager URL, the user's credentials are authenticated. In Internet Explorer, security settings control whether this occurs automatically or if the user must enter a user name and password. To change these settings, complete the following steps:

To test a user's security

1. In Internet Explorer 7 and later, click **Tools, Internet Options**, and then click the **Security** tab.
2. Click **Local Intranet** and then the **Custom Level** button.
3. In the **User Authentication** section, choose **Prompt for user name and password**.
4. The next time you open the browser window, you will be prompted for a user name and password.

See Also

Concepts

[Security \(Master Data Services\)](#)

