

**Jeremy Kashel  
Tim Kent  
Martyn Bullerwell**

**Chapter No.2  
"Master Data Services Overview"**

# 2

## Master Data Services Overview

In this chapter, we will provide an overview of SQL Server 2008 R2 Master Data Services (MDS), outlining the main features of the product as well as covering the architecture. In addition to covering the architecture, the purpose of this chapter is to get the reader comfortable with navigating around the main MDS interface, namely **Master Data Manager**, before we launch into the detailed modeling topics in further chapters.

In this chapter, we will cover the following:

- Master Data Services overview
- Master Data Services architecture
- Master Data Manager

### Master Data Services overview

Master Data Services is Microsoft's Master Data Management product that ships with SQL Server 2008 R2. Much like other parts of SQL Server, such as Analysis Services (SSAS) or Reporting Services (SSRS), MDS doesn't get installed along with the database engine, but is a separate product in its own right.

Unlike SSAS or SSRS, it's worth noting that MDS is only available in the Enterprise and Data Centre Editions of SQL Server, and that the server must be 64-bit.

MDS is a product that has grown from acquisition, as it is based on the +EDM product that Microsoft obtained when they bought the Atlanta-based company called Stratature in 2007. A great deal of work has been carried out since the acquisition, including changing the user interface and putting in a web service layer.

At a high level, the new product has the following features:

- Entity maintenance – MDS supports data stewardship by allowing users to add, edit, and delete members. The tool is not specific to a particular industry or area, but instead is generic enough to work across a variety of subject domains.
- Modeling capability – MDS contains interfaces that allow administrative users to create data models to hold entity members.
- Hierarchy management – Relationships between members can be utilized to produce hierarchies that users can alter.
- Version management – Copies of entity data and related metadata can be archived to create an entirely separate version of the data.
- Business rules and workflow – A comprehensive business rules engine is included in order to enforce data quality and assist with data stewardship via workflow. Alerts can be sent to users using e-mail when the business rules encounter a particular condition.
- Security – A granular security model is included, where it is possible, for example, to prevent a given user from accessing certain entities, attributes, and members.

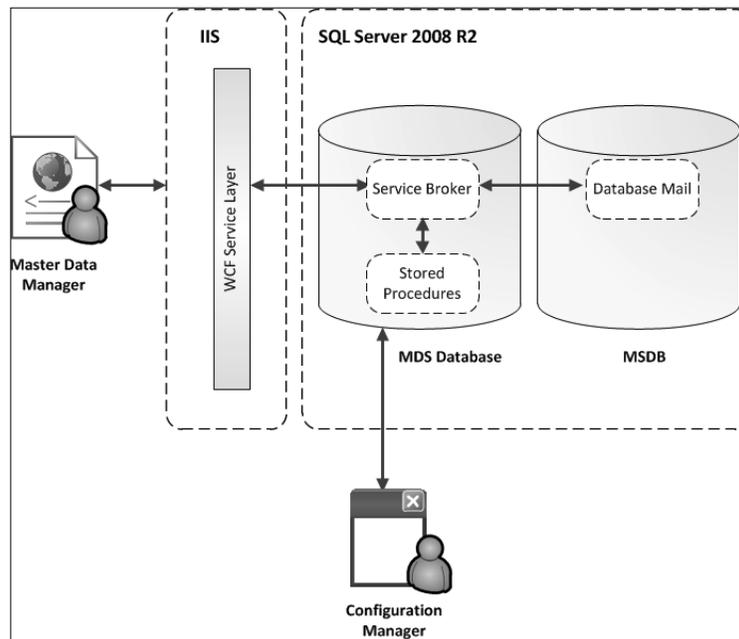
## **Master Data Services architecture**

Technically, Master Data Services consists of the following components:

- SQL Server Database – The database holds the entities such as Customer or Product, whether they are imported from other systems or created in MDS.
- Master Data Manager – A web-based data stewardship and administration portal that amongst many other features allows data stewards to add, edit, and delete entity members.
- Web Service Layer – All calls to the database from the front-end go through a WCF (Windows Communication Foundation) web service. Internet Information Services (IIS) is used to host the web services and the Master Data Manager application.

- Workflow Integration Service – A Windows service that acts as a broker between MDS and SharePoint in order to allow MDS business rules to use SharePoint workflows.
- Configuration Manager – A windows application that allows key settings to be altered by an administrator.

The following diagram shows how the components interact with one another:



## MDS SQL Server database

The MDS database uses a mix of components in order to be the master data store and to support the functionality found in Master Data Manager, including stored procedures, views, and functions.

Separate tables are created both for entities and for their supporting objects, all of which happens on the fly when a new object gets created in Master Data Manager. The data for all entities across all subject areas are stored in the same database, meaning that the database could get quite big if several subject domains are being managed. The tables themselves are created with a code name. For example, on my local installation, the Product entity is not stored in a table called "Product" as you might expect, but in a table called "tbl\_2\_10\_EN".



#### Locating entity data

The exact table that contains the data for a particular entity can be found by writing a select statement against the view called `viw_SYSTEM_SCHEMA_ENTITY` in the `mdm` schema.

As well as containing a number of standard SQL Server table-valued and scalar functions, the MDS database also contains a handful of .Net CLR (Common Language Runtime)-based functions, which can be found in the **mdq** schema. The functions utilize the `Microsoft.MasterDataServices.DataQuality` assembly and are used to assist with data quality and the merging, de-duplication, and survivorship exercises that are often required in a master data management solution.

Some of the actions in MDS, such as the e-mail alerts or loading of large amounts of data, need to happen in an asynchronous manner. Service Broker is utilized to allow users to continue to use the front-end without having to wait for long running processes to complete.

Although strictly outside the MDS database, SQL Server Database Mail, which resides in the system **msdb** database, is used as the mechanism to send e-mail alerts to subscribing users.

In addition to the tables that hold the master data entities, the MDS database also contains a set of staging tables that should be used when importing data into MDS. Once the staging tables have been populated correctly, the staged data can be loaded into the master data store in a single batch.

## Internet Information Services (IIS)

During the initial configuration of MDS, an IIS Web Application will get created within an IIS website. The name of the application that gets created is called "MDS", although this can be over-ridden if needed.

The Web Application contains a Virtual Directory that points to the physical path of `<Drive>:\<install location>\WebApplication`, where the various components of the Master Data Manager application are stored.

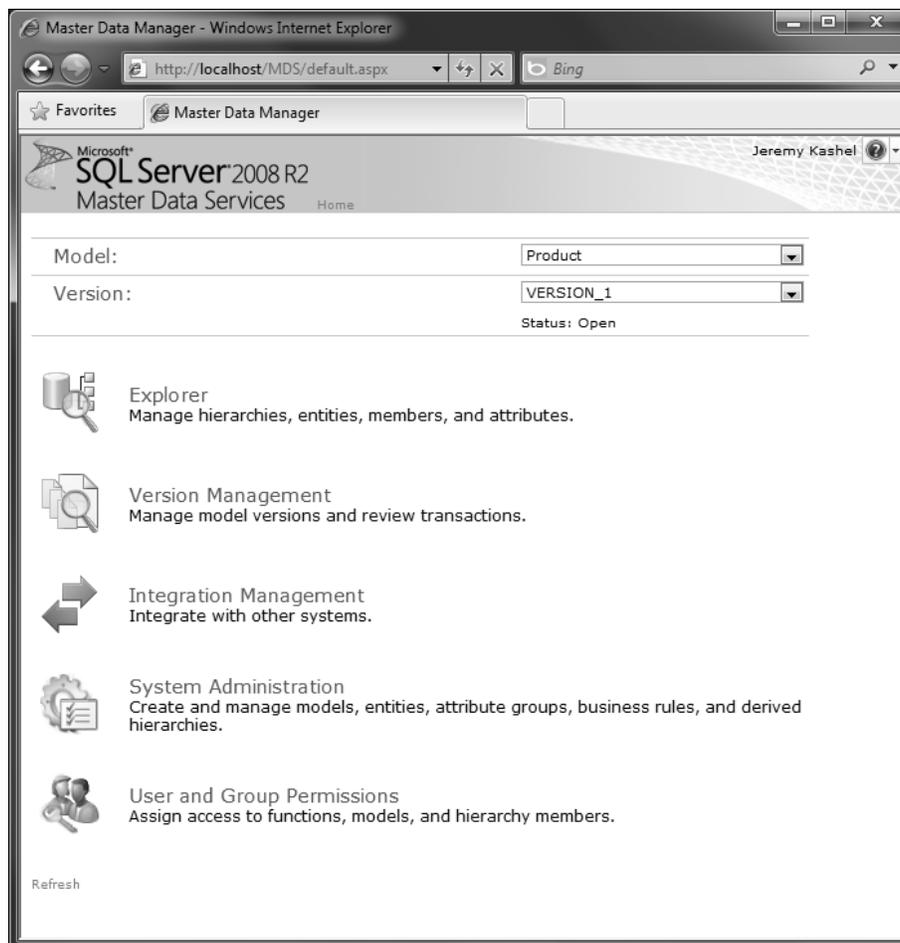
The MDS WCF Service called `Service.svc` is also located in the same directory. The service can be exposed in order to provide a unified access point to any MDS functionality (for example, creating an entity or member, retrieving all entity members) that is needed by other applications. Master Data Manager connects to the WCF service, which then connects to the database, so this is the route that should be taken by other applications, instead of connecting to the database directly.

# Master Data Manager

Master Data Manager is the primary mechanism for accessing all the Master Data Services functionality and is therefore the focus of the earlier parts of this book. We will introduce Master Data Manager in this section, before covering the detailed functionality in further chapters.

## Accessing Master Data Manager

With the default installation settings, Master Data Manager is accessed in a web browser by typing `http://<servername>/MDS/`. This takes us to the front page, as shown in the following screenshot:



As indicated in the previous screenshot, Master Data Manager has five functional areas that users of the application may access, namely:

- **Explorer**
- **Version Management**
- **Integration Management**
- **System Administration**
- **User and Group Permissions**

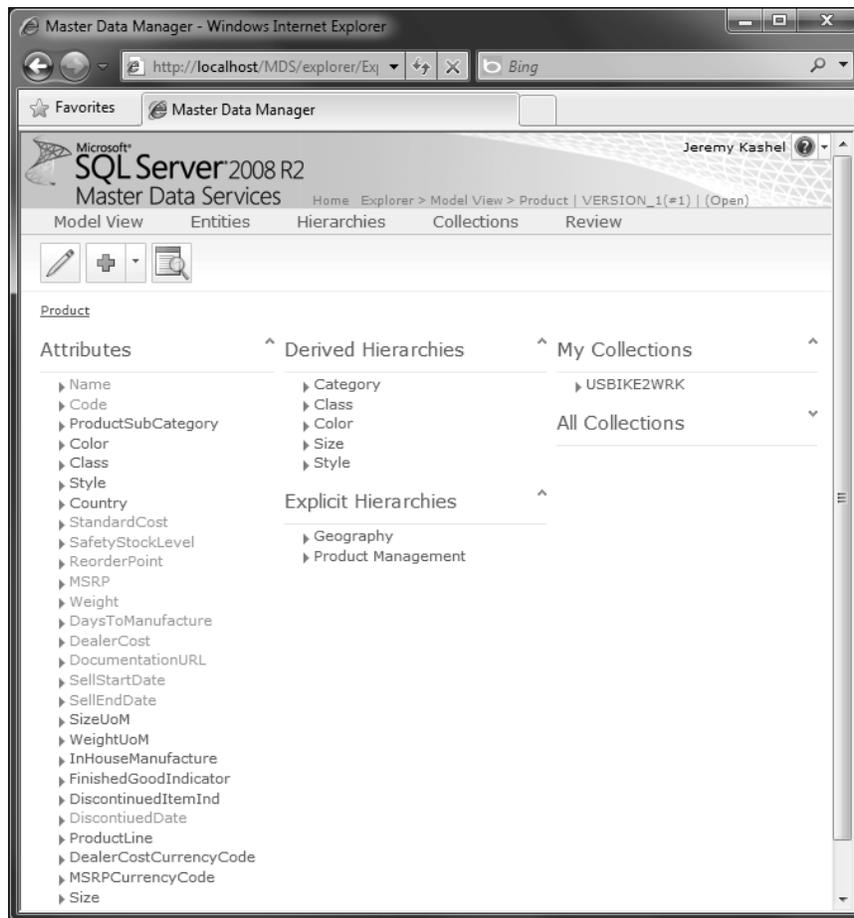
We will now look at each of these functions in turn, and explore how they can help with an MDM initiative.

## Explorer

The Explorer part of the application will be the primary interface used by data stewards as it allows a user to carry out the following:

- Add, edit, or delete entity members.
- Manage hierarchies.
- View transaction details, for example, who added a particular member and when.
- Reverse transactions that have previously been made.
- Run business rules.
- Export data to Microsoft Excel.
- Create and manage collections, which are user-defined groups of members that exist within an entity. Collections will be explained in detail in *Chapter 4, Creating and Using Models*, along with hierarchies and other objects that are key to MDS.

Clicking on the **Explorer** item on the front page takes us to the **Model View** screen, which gives an overview of the MDS view of our current subject area, for example, **Product**. The **Model View** of the sample **Product** model is shown next:

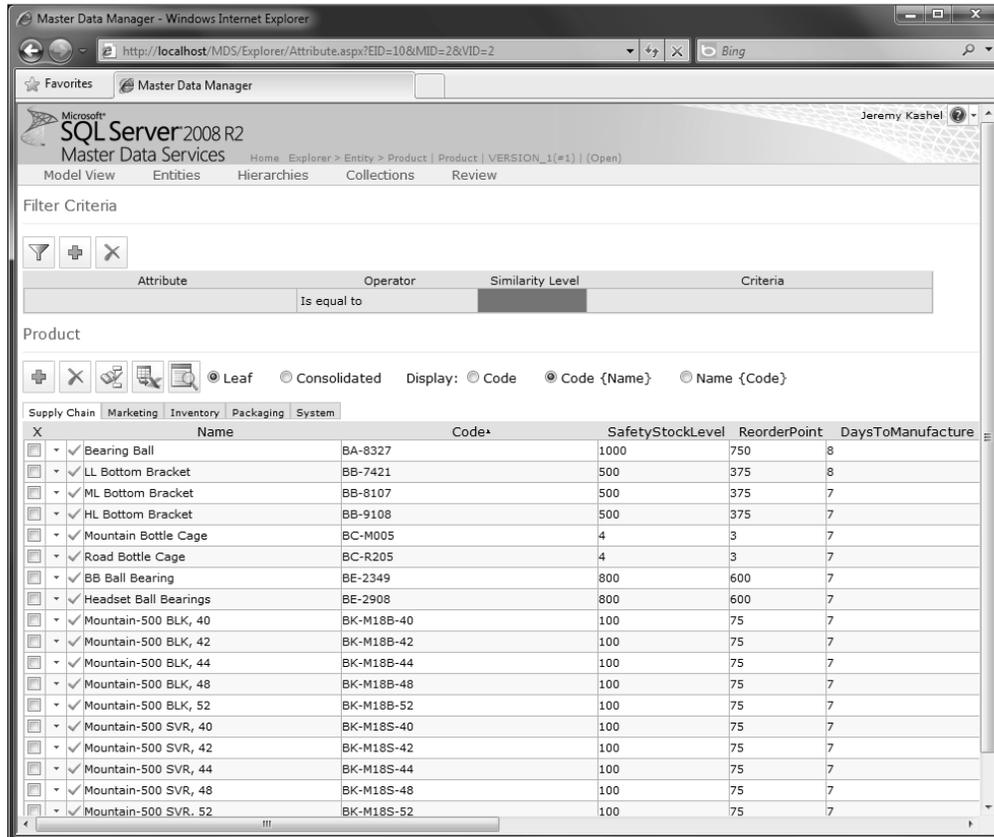


The above screenshot shows all the objects that are used by **Product**, namely a number of attributes, hierarchies, and collections. We can click on some of these objects to take us to the object itself. For example, clicking on the hierarchy called **Category** will take us to the hierarchy itself. Alternatively, we can also navigate to the various items using the menus shown at the top of the screen.

One of the most common tasks for a data steward to carry out will be to go to the Explorer Grid, which is a table structure that shows all the members in the chosen entity. Clicking on the "pencil" icon at the top of the screen is the next step to carry out:



Once this has been carried out, we will get to the Explorer Grid page, as shown in the following screenshot:



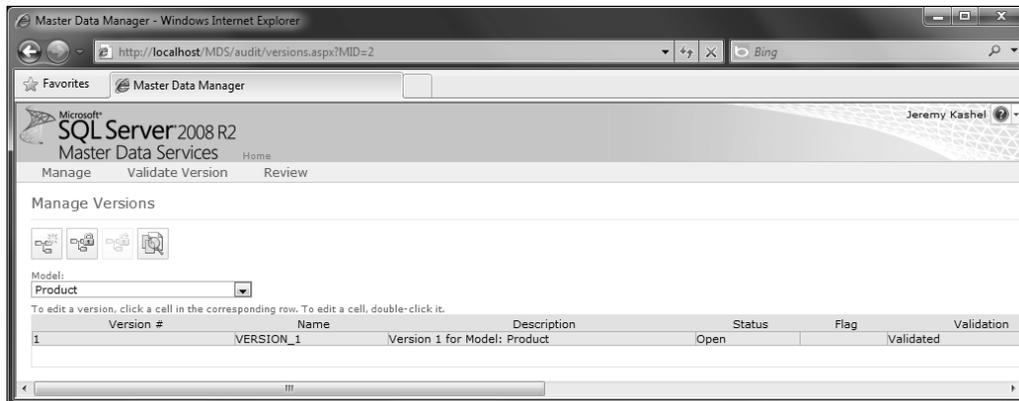
The **Filter Criteria** section at the top of the screen allows us to filter on a chosen entity attribute, in order to make searching easier, whereas the grid at the bottom of the screen contains the members themselves.

Within the grid, the columns are the attributes of the **Product** Entity, such as **Name** or **ReorderPoint**, whereas the rows are the members.

It is within this grid that the adding, editing, and deleting of members can be carried out.

## Version Management

Clicking on the Master Data Services logo in the top right-hand corner takes us back to the Master Data Manager home page. From here, if we pick the **Version Management** option, we're taken to the **Manage Versions** page, as shown in the following screenshot:



With **Version Management**, it is possible to carry out the following:

- Create a copy of an entity and its associated data, which is known as a version
- Control access to the version
- Validate the version
- Review any issues with the version
- View and reverse transactions that have occurred against a specific version

Versions provide an easy mechanism to create physical copies of all the members that exist within a given entity. This may be useful in an MDM program in order to record the historical changes to master data at a given point in time or to record a major data change, such as organizational realignment.

A new version is created by copying an old version. At this point, MDS can be configured to ensure that only the new version is available for editing, to prevent alterations to historical data.

Versions also contain the concept of validation. If a version is validated, then it meets the business requirements for the given master data entity, and can be safely extracted by any downstream systems that need the master data. Business rules run when a version is validated and will fail if their conditions are not met.

Validation is a key concept within Master Data Services and is covered in detail in Chapter 5, *Version Management*.

## Integration Management

The third option available on the front page of Master Data Manager is Integration Management. Within Integration Management it is possible to:

- Import data into MDS
- Export data out of MDS

Clicking on the **Integration Management** option defaults the screen to the **Import** page, as shown in the following screenshot:

The screenshot shows the Master Data Manager web interface in Internet Explorer. The browser address bar shows the URL: `http://localhost/MDS/importexport/process.aspx?MI`. The page title is "Master Data Manager" and the user is logged in as "Jeremy Kasha". The page displays the "Import" and "Export" options. Below these, there is a section titled "Staging Batches" containing a table with the following data:

ID	Name	Model	Version	Status	Started	Completed	Records	Errors
117		CustomerSample	VERSION_1	Not Running	08/01/2011 14:06:03	08/01/2011 14:07:35	109704	0
140	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 17:10:53	13/01/2011 17:11:02	1	0
142	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 17:19:13	13/01/2011 17:19:13	1	0
144	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 17:25:10	13/01/2011 17:25:10	2	0
145	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 17:27:42	13/01/2011 17:27:43	1	0
146	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 17:38:54	13/01/2011 17:38:55	2	0
148	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 17:40:48	13/01/2011 17:40:48	2	0
151	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 17:43:44	13/01/2011 17:43:45	1	0
155	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 17:52:22	13/01/2011 17:52:22	1	0
157	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 19:44:55	13/01/2011 19:44:56	1	0
158	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 19:51:15	13/01/2011 19:51:16	1	0
159	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 19:52:16	13/01/2011 19:52:17	1	1
160	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 19:54:27	13/01/2011 19:54:27	1	0
161	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 19:56:05	13/01/2011 19:56:05	1	0
162	BRTEST unbatched	BRTest	VERSION_1	Not Running	13/01/2011 19:59:06	13/01/2011 19:59:07	1	0
164	SUPPLIER unbatched	Supplier	VERSION_1	Not Running	13/01/2011 22:33:16	13/01/2011 22:33:24	1	0
166	SUPPLIER unbatched	Supplier	VERSION_1	Not Running	13/01/2011 23:04:48	13/01/2011 23:04:49	1	0
167	SUPPLIER unbatched	Supplier	VERSION_1	Not Running	13/01/2011 23:35:00	13/01/2011 23:35:01	1	0
168	SUPPLIER unbatched	Supplier	VERSION_1	Not Running	13/01/2011 23:35:20	13/01/2011 23:35:20	1	0
169	SUPPLIER unbatched	Supplier	VERSION_1	Not Running	13/01/2011 23:35:33	13/01/2011 23:35:33	1	0

Below the table, there is a section titled "Unbatched Staging Records" with a search icon. It includes the following fields and values:

- Model: Product
- Version: VERSION\_1
- Total member records: 0
- Total attribute records: 0
- Total relationship records: 0

The grid at the top of the screen, under **Staging Batches**, shows the historical data that has been imported into the MDS database, as well as any data that is currently importing or is queued to import.

Importing is initiated by the button at the bottom of the screen. This causes Service Broker to kick off the loading of any data that exists within the staging tables, whereas the user can monitor the load of data by refreshing the grid at the top of the page.

Unlike the **Import** page, the **Export** page does not actually export master data to subscribing systems, but instead allows the creation of objects that external systems can connect to.

It is therefore entirely down to the solution architect as to how master data gets propagated to systems that need it. Depending on the MDM architecture style that needs to be used, subscribing systems may need to store a local copy of the data.

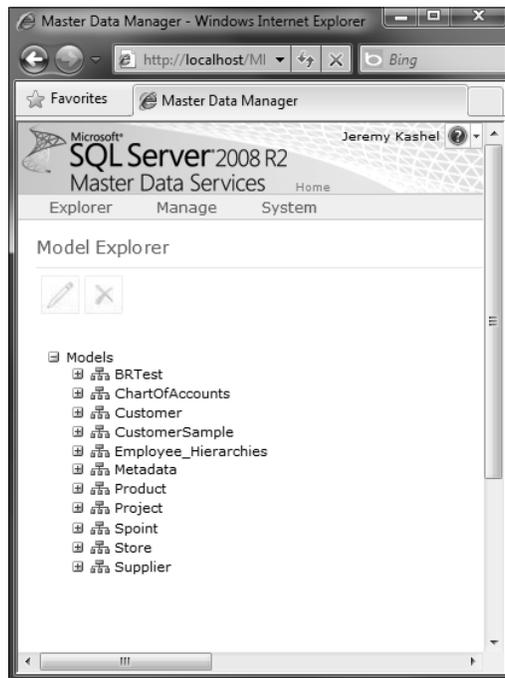
If data does need to be transferred to external systems, then another product within SQL Server is a good choice to help with this requirement. SQL Server Integration Services (SSIS) is Microsoft's ETL (Extract Transform Load) tool and it contains specific functionality for moving large volumes of data from a source to a target.

Integration Services is covered in more detail in *Chapter 6, Importing Data Into Master Data Services*.

## System Administration

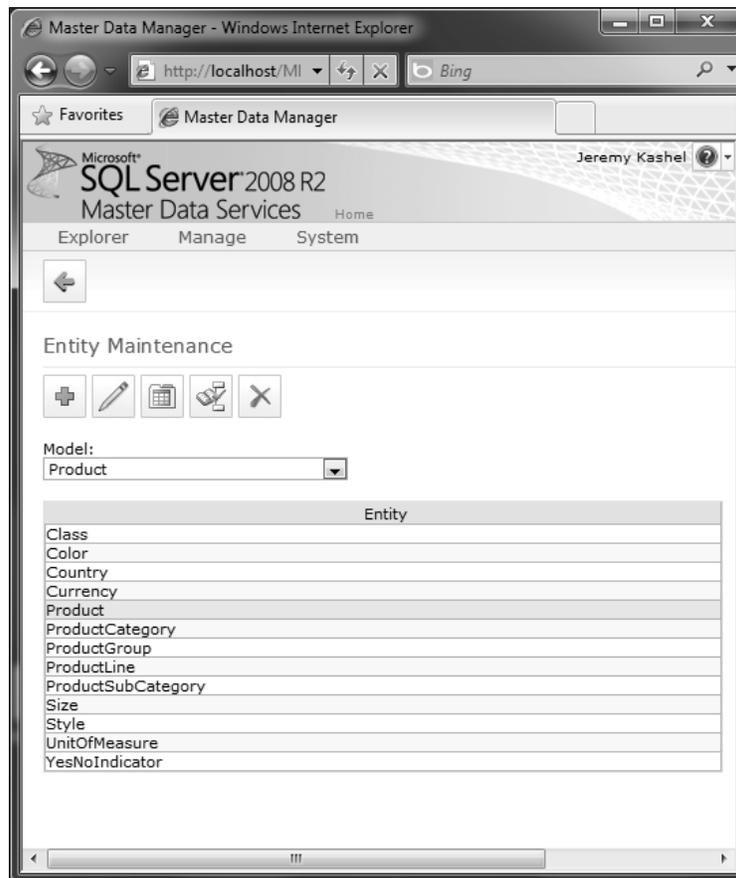
System Administration is the place within Master Data Manager where the MDS entities and all their related objects actually get created. For those of you who have developed operational or BI solutions using SQL Server, it may come as a surprise that this is not carried out in SQL Server Management Studio or Visual Studio.

Clicking on the **System Administration** option gives the following screen:



The **System Administration** screen provides two ways in which to manage the entities and their related objects. This can either be carried out by navigating the above tree or by utilizing the **Manage** menu item at the top of the screen.

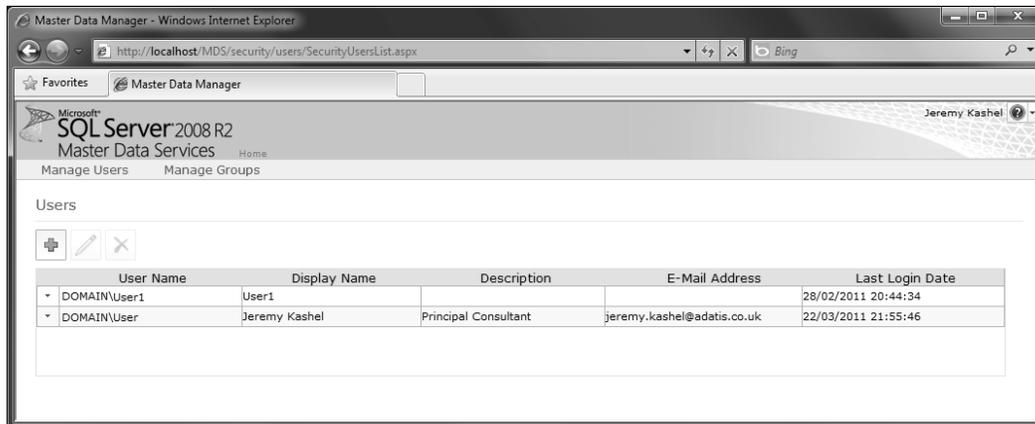
The various sub screens that exist within **System Administration** allow for the creation, editing, and deleting of the MDS objects. As shown next, the familiar plus and pencil icons are used to add and edit objects respectively:



Finally the System menu option that's shown in the above image is the place whereby a deployment package can be created. Deployment packages are useful for deploying the entity structures from a test to a production environment, for example.

## User and Group Permissions

As master data is one of the organization's prime data assets, it stands to reason that there should be some way of securing this data. This process is carried out within Master Data Manager by clicking on the **User and Group Permissions** option, the result of which is shown below:



MDS allows a given Windows user access to Master Data Manager if the user has been added to the MDS list of users, which is done by adding the user in the form of DOMAIN\User or COMPUTER\_NAME\User. Alternatively, Windows groups can also be added to MDS, albeit on the adjacent menu option called **Manage Groups**.

Once the Windows user or group has been added to MDS, a variety of permissions can be applied. At a high level, access can be granted to one or more of the five Master Data Manager functions.

Detailed permissions can also be applied to a user or group, to the extent that read and write permissions can be altered for an entity, attribute, or even a single member. We will cover security in *Chapter 10, Master Data Services Security*.

## Summary

In this chapter, we've gained an overview of Master Data Services and have learned the following key points:

- Master Data Services ships with the Enterprise and Data Center Editions of SQL Server, allowing organizations to carry out Master Data Management at a much lower cost.
- MDS assists with an MDM solution by providing generic entity modeling and maintenance capability.
- The underlying SQL Server database stores entity data as well as any required metadata.
- Master Data Manager is the main application that ships with MDS, providing the following features:
  - Data stewardship using the **Explorer** function
  - Versioning of entity data using the **Version Management** function
  - The ability to import data using the **Integration Management** function, as well as the ability to create objects to assist with data export
  - The creation and maintenance of entities and their related objects through the **System Administration** function
  - Management of security through the **User and Group Permissions** function

In the next chapter, we will learn how to install and configure Master Data Services.

## Where to buy this book

You can buy Microsoft SQL Server 2008 R2 Master Data Services from the Packt Publishing website: <http://www.packtpub.com/microsoft-sql-server-2008-r2-master-data-services/book>.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our [shipping policy](#).

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.



[www.PacktPub.com](http://www.PacktPub.com)