

**Tutorial: Enterprise Information Management using SSIS, MDS, and DQS Together**

SQL Server Technical Article

**Writer:** Sreedhar Pelluru, Jaime Alva Bravo

**Technical Reviewer:** Carla Sabotta, Jim van de Erve, Kumar Vivek, Matt Masson, Matthew Roche

**Published:** October 2012

**Applies to:** SQL Server 2012

**Summary:** Managing information in an enterprise typically involves integrating data from across the enterprise and beyond, cleansing the data, matching the data to remove any duplicates, standardizing the data, enriching the data, making the data conform to legal and compliance requirements, and then storing the data in a centralized location with all the necessary security settings.

In this tutorial, you will learn how to use SQL Server Integration Services (SSIS), Master Data Services (MDS), and Data Quality Services (DQS) together to implement a sample Enterprise Information Management (EIM) solution. First, you will use DQS to create a knowledgebase that contains knowledge about the data (metadata), cleanse the data in an Excel file using the knowledge base, and match the data to identify and remove duplicates in the data. Next, you will use the MDS Add-in for Excel to upload the cleansed and matched data to MDS. Then, you will automate the whole process using an SSIS solution.

Copyright

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. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious.  No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2011 Microsoft. All rights reserved.

Contents

[Overview 5](#_Toc338958126)

[Prerequisites 6](#_Toc338958127)

[Lessons 7](#_Toc338958128)

[Lesson 1: Creating the Suppliers DQS Knowledge Base 7](#_Toc338958129)

[Task 1: Creating a Knowledge Base and Domains 8](#_Toc338958130)

[Task 2: Adding Domain Values Manually 12](#_Toc338958131)

[Task 3: Importing Domain Values from an Excel File 12](#_Toc338958132)

[Task 4: Setting Domain Rules 13](#_Toc338958133)

[Task 5: Setting Term-Based Relations 15](#_Toc338958134)

[Task 6: Setting Synonyms 15](#_Toc338958135)

[Task 7: Creating a Composite Domain 16](#_Toc338958136)

[Task 8: Creating a Composite Domain Rule 17](#_Toc338958137)

[Task 9: Configuring a Reference Data Service 18](#_Toc338958138)

[Task 10: Configuring Composite Domain to Use Reference Data Service 19](#_Toc338958139)

[Task 11: Publishing the Knowledge Base 20](#_Toc338958140)

[Task 12: Discovering Knowledge (Knowledge Discovery) 21](#_Toc338958141)

[Lesson 2: Cleansing Supplier Data using the Suppliers Knowledge Base 24](#_Toc338958142)

[Task1: Creating a Data Quality Project 24](#_Toc338958143)

[Task 2: Mapping Excel Columns to DQS Domains 25](#_Toc338958144)

[Task 3: Cleansing Data against the Supplier Knowledge Base 27](#_Toc338958145)

[Task 4: Managing and Viewing Results 28](#_Toc338958146)

[Task 5: Exporting Cleansing Results to an Excel File 31](#_Toc338958147)

[Task 6: Importing Values from the Cleanse Supplier List Project 32](#_Toc338958148)

[Lesson 3: Matching Data to Remove Duplicates from Supplier List 33](#_Toc338958149)

[Task 1: Defining a Matching Policy 33](#_Toc338958150)

[Task 2: Testing and Publishing the Matching Policy 36](#_Toc338958151)

[Task 3: Creating and Running a Data Quality Project for Matching 38](#_Toc338958152)

[Task 4: Exporting the Results from Matching Activity to an Excel File 39](#_Toc338958153)

[Lesson 4: Storing Supplier Data in MDS 40](#_Toc338958154)

[Task 1: Creating Suppliers Model using Master Data Manager 42](#_Toc338958155)

[Task 2: Uploading Supplier Data to MDS using MDS Add-in for Excel 43](#_Toc338958156)

[Task 3: Verifying the Data in Master Data Manager 46](#_Toc338958157)

[Task 4 (Optional): Combining, Matching, and Publishing New Set of Data 47](#_Toc338958158)

[Task 5: Creating a Domain-Based Attribute from Excel 50](#_Toc338958159)

[Task 6: Verify that the Domain-Based Attribute is Created using Master Data Manager 52](#_Toc338958160)

[Task 7: Viewing Updates Made using Master Data Manager in Excel 54](#_Toc338958161)

[Task 8: Adding a New Value for State Entity in Excel 55](#_Toc338958162)

[Task 9: Creating a Derived Hierarchy using Master Data Manager 57](#_Toc338958163)

[Lesson 5: Automating the Cleansing and Matching using SSIS 60](#_Toc338958164)

[Task 1 (Prerequisite): Removing Supplier Data in MDS 61](#_Toc338958165)

[Task 2 (Optional): Creating a MDS Subscription View using Master Data Manager 62](#_Toc338958166)

[Task 3 (Optional): Reviewing the Subscription Views 63](#_Toc338958167)

[Task 4: Creating an SSIS Project using SQL Server Data Tools 63](#_Toc338958168)

[Task 5: Adding Data Flow Task 65](#_Toc338958169)

[Task 6: Adding Excel Source to the Data Flow 66](#_Toc338958170)

[Task 7: Adding DQS Cleansing Transform to the Data Flow 67](#_Toc338958171)

[Task 8: Adding Conditional Split Transform to Split Cleansing Output 69](#_Toc338958172)

[Task 9: Adding Union All Transform to Combine Correct and Corrected Records 71](#_Toc338958173)

[Task 10: Adding Fuzzy Group Transform to Identify Duplicates 73](#_Toc338958174)

[Task 11: Adding Conditional Split Transform to Filter Duplicates 75](#_Toc338958175)

[Task 12: Adding Derived Column Transform to Add Columns Required by MDS 75](#_Toc338958176)

[Task 13: Adding OLE DB Destination to Write Data to MDS Staging Table 77](#_Toc338958177)

[Task 14: Adding Execute SQL Task to Control Flow to Run the Stored Procedure for MDS 79](#_Toc338958178)

[Task 15: Building and Running the SSIS Project 81](#_Toc338958179)

[Task 16: Verifying with Master Data Manager 83](#_Toc338958180)

[Task 17: Reviewing DQS Cleansing Project Created by the SSIS package 83](#_Toc338958181)

[Conclusion 84](#_Toc338958182)

## Overview

Managing information in an enterprise typically involves integrating data from across the enterprise and beyond, cleansing the data, matching the data to remove any duplicates, standardizing the data, enriching the data, making the data conform to legal and compliance requirements, and then storing the data in a centralized location with all the necessary security settings.

SQL Server 2012 provides all the components needed for an effective Enterprise Information Management (EIM) solution in a single product. Key components of SQL Server 2012 that help you build an EIM solution are:

* SQL Server Integration Services
* SQL Server Data Quality Services
* SQL Server Master Data Services

SQL Server Integration Services (SSIS) provides a powerful, extensible platform for integrating data from a variety of sources in a comprehensive extract, transform, and load (ETL) solution that supports business workflows, a data warehouse, or master data management. See [Integration Services Overview](http://msdn.microsoft.com/library/ms141263(SQL.105).aspx) topic for a quick overview and typical uses of SSIS.

SQL Server Data Quality Services (DQS) enables you to cleanse, match, standardize, and enrich data, so you can deliver trusted information for business intelligence, a data warehouse, and transaction processing workloads. See [Introducing Data Quality Services](http://msdn.microsoft.com/library/ff877917.aspx) topic for the business need for DQS and how DQS answers the need.

SQL Server Master Data Services (MDS) provides a central data hub that ensures that the integrity of information and consistency of data is constant across different applications. See [Master Data Services Overview](http://msdn.microsoft.com/library/ff487003.aspx) topic for brief descriptions of important features of MDS.

See [Enterprise Information Management with SQL Server 2012](http://download.microsoft.com/download/B/F/8/BF81264C-413F-42FC-8FBF-171D97B3C8A7/SQL_Server_2012_Enterprise_Information_Management_Whitepaper.pdf) and [Cleansing and Matching Master Data using EIM Technologies](http://msdn.microsoft.com/en-us/library/hh403491.aspx) whitepapers for a comprehensive guidance on implementing an EIM solution using these Microsoft EIM technologies together and watch [Enterprise Information Management (EIM): Bringing together SSIS, DQS, and MDS](http://go.microsoft.com/fwlink/?LinkId=258672) video for a cool demonstration of an EIM scenario.

In this tutorial, you will learn how to use SSIS, MDS, and DQS together to implement a sample Enterprise Information Management (EIM) solution. First, you will use DQS to create a knowledgebase that contains knowledge about the data (metadata), cleanse the data in an Excel file using the knowledge base, and match the data to identify and remove duplicates in the data. Next, you will use the MDS Add-in for Excel to upload the cleansed and matched data to MDS. Then, you will automate the whole process using an SSIS solution. The SSIS solution in this tutorial reads the input data from an Excel file, but you can extend it to read from a variety of sources such as Oracle, Teradata, DB2, and SQL Azure.

### Prerequisites

* Microsoft SQL Server 2012 with the following components installed.
  + Integration Services (SSIS).
  + Master Data Services (MDS)
  + Data Quality Services (DQS)
  + SQL Server Data Tools

See [SQL Server 2012 Installation Guide](http://msdn.microsoft.com/library/bb500469.aspx) for details about installing the product.

* [Configure MDS using Master Data Services Configuration Manager](http://msdn.microsoft.com/en-us/library/ee633884.aspx)

Use the Configuration Manager to create and configure a Master Data Services database. After you create the MDS database, create a Web application for MDS in a Web site (for example: <http://localhost/MDS>) and associate the MDS database with the MDS Web application. Note that, to create an MDS Web application, you need to have IIS installed on your computer. See [Web Application Requirements (Master Data Services)](http://msdn.microsoft.com/library/ee633744.aspx) and [Database Requirements (Master Data Services)](http://msdn.microsoft.com/library/ee633767.aspx) for details about the prerequisites for configuring MDS database and Web application.

* [Install and Configure DQS using Data Quality Server Installer](http://msdn.microsoft.com/en-us/library/hh231682.aspx). Click Start, click All Programs, click Microsoft SQL Server 2012, click Data Quality Services, and then click Data Quality Server Installer.
* Microsoft Excel 2010 (32-bit is preferred)
* Install **Master Data Services Add-in for Excel** (32-bit or 64-bit based on the version of Excel you have on your computer) from [here](http://www.microsoft.com/en-us/download/details.aspx?id=29064). To find the version of Excel installed on your computer, run **Excel**, click **File** on menu bar and click **Help** to see the version in the right pane. Note that you need to install [Visual Studio 2010 Tools for Office Runtime](http://www.microsoft.com/download/details.aspx?id=20479) prior to installing the Excel Add-in.
* (Optional) Create an account with [Windows Azure](https://datamarket.azure.com/) Marketplace. One of the tasks in the tutorial requires you to have an **Azure Marketplace** (originally named **Data Market**) account. You can skip this task if you want and proceed with the next task.
* DQS does not allow you to export the cleansing or matching results to an Excel file if you are using **64-bit version of Excel**. This is a known issue. To work around the issue, do the following:
  + Install [SQL Server 2012 SP1](http://www.microsoft.com/en-us/download/details.aspx?id=34700) (on 64-bit computers with 64-bit Excel).
  + Run **DQLInstaller.exe –upgrade**. If you installed the default instance of SQL Server, the DQSInstaller.exe file will be available at C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Binn. Double-click the DQSInstaller.exe file.
  + In **Master Data Services Configuration Manager**, click **Select Database**, select existing **MDS** database, and then click **Upgrade**.

### Lessons

This tutorial includes the following lessons:

|  |  |  |
| --- | --- | --- |
| **Lesson** | **Brief description** | **Estimated time to complete (in minutes)** |
| [Lesson 1: Creating the Suppliers DQS Knowledge Base](#_Lesson_1:_Creating) | In this lesson, you will create a DQS knowledge base named **Suppliers**. | 45 |
| [Lesson 2: Cleansing Supplier Data using the Suppliers Knowledge Base](#_Lesson_2:_Cleansing) | In this lesson you will create and run a DQS project to cleanse the supplier data in an Excel file using the **Suppliers** KB you created in the first lesson. | 30 |
| [Lesson 3: Matching Data to Remove Duplicates from Supplier List](#_Lesson_3:_Matching) | In this lesson, you will create a DQS project to perform matching activity to identify and remove duplicates from the cleansed suppler list. | 30 |
| [Lesson 4: Storing Supplier Data in MDS](#_Lesson_4:_Storing) | In this lesson, you will upload the cleansed and matched supplier data to Master Data Services (MDS) by using the **MDS Add-in for Excel**. | 30 |
| [Lesson 5: Automating the Cleansing and Matching using SSIS](#_Lesson_5:_Automating) | In this lesson, you will create an SSIS solution that cleanses input data using DQS, matches the cleansed data to remove duplicates, and stores the cleansed and matched data on MDS in an automated manner. | 75 |

## Lesson 1: Creating the Suppliers DQS Knowledge Base

In this lesson, you will create a DQS knowledge base named **Suppliers** with the knowledge (metadata) about supplier data. You will use the knowledge base to perform cleansing and matching activities on input supplier data. The cleansing activity identifies incorrect/invalid data, corrects the incorrect data or proposes corrections/suggestions, standardizes the data, and enriches the data with additional information. The matching activity compares data and identifies similar records (may be slightly different) in the data that helps you perform de-duplication (remove duplicates) on the data.

You can use both interactive and computer-assisted processes to create, build, and manage a knowledge base. Knowledge in a knowledge base is maintained in domains, each of which is specific to a data field in the data that you want to cleanse and/or match.

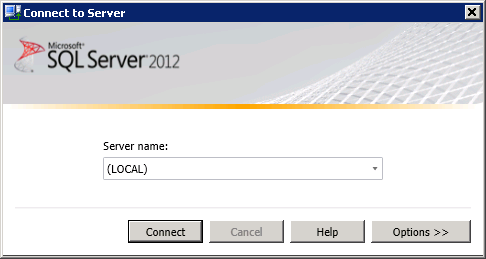
In this lesson, you will perform the following tasks to create the **Suppliers** knowledge base:

* Create a DQS knowledge base named **Suppliers**. You can create a knowledge base in several ways. You can build a KB from scratch or build it based on an existing knowledge base or by importing a DQS file (.dqs) that contains a pre-built and exported knowledge base, or by performing a knowledge discovery activity on sample data. In this tutorial, you will create the KB from scratch.
* Create domains in the Suppliers knowledge base that you will use for cleansing data, and matching data to identify duplicates. You should create domains for data fields that you want to use in cleansing and matching activities, not for all the data fields in the data.
* Add values to a domain by adding values manually, importing values from an excel file, by performing a knowledge discovery activity on sample data, and by importing project values from a cleansing project. You can also import domain values by importing a DQS file containing domain properties and values, which you will not perform in the tutorial.
* Set rules for a domain. A domain rule is a condition that will be used by DQS to validate, correct, and standardize domain values.
* Set term-based relationships for a domain. A term-based relationship enables you to make a correction to a term that is part of a value in a domain. For example, in the value **Contoso Inc.**, **Inc.** is a term that can be defined as Incorporated. This helps in standardizing the data as well as in identifying duplicates. For example, **Contoso Inc.** and **Contoso Incorporated** can be considered duplicates.
* Specify synonyms in domain values. You can set two or more values as synonyms and set one of them as a leading value, which replaces its synonym values during a cleansing activity to standardize the data.
* Create a composite domain named Address Validation that comprises Address Line, City, State, and Zip domains. A composite domain is a domain that consists of one or more single domains. It lets you create a rule that involves multiple domains. For example, you can define a rule: if City is Los Angeles, State must be CA, where City and State are two separate domains.
* Configure and use a reference data provider/service. The Reference Data Service feature in Data Quality Services (DQS) enables you to subscribe to third-party reference data providers, and to easily cleanse and enrich your business data by validating it against their high-quality data. You can use services from leading data quality service providers from within DQS to standardize, correct, or enrich your data during the cleansing process. In this tutorial, you will learn how to configure your DQS environment to use a reference data service on Windows Azure Marketplace and use the service associated with the Address Validation composite domain to cleanse address data.
* Publish the knowledge base so that the KB can be used in cleansing and matching activities.

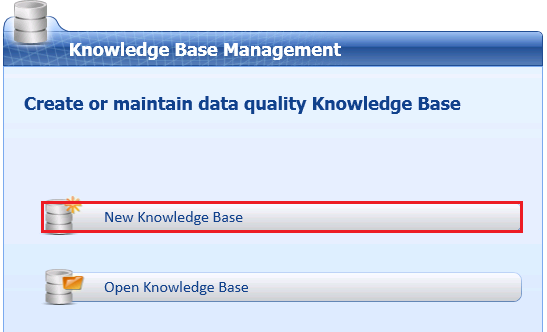
### Task 1: Creating a Knowledge Base and Domains

In this task, you will create the **Suppliers** knowledge base and create domains that will be used for cleansing data and matching data to remove duplicates.

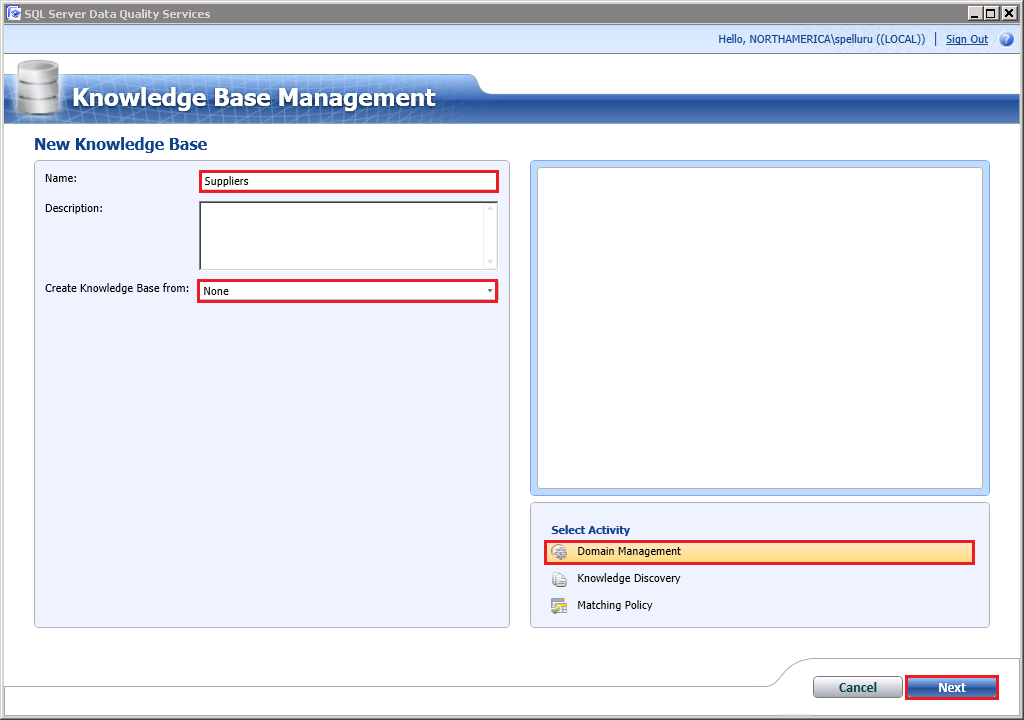
1. Launch **Data Quality Client**. Click **Start**, point to **All Programs**, click **Microsoft SQL Server 2012**, click **Data Quality Services**, and then click **Data Quality Client**.
2. In the **Connect to Server** dialog box, select the database server instance on which the DQS is installed, and click **Connect**.



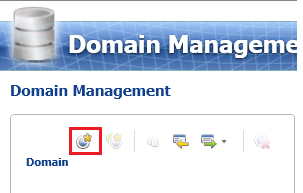
1. In the Data Quality Client home page, in the **Knowledge Base Management** pane, click **New Knowledge Base**.



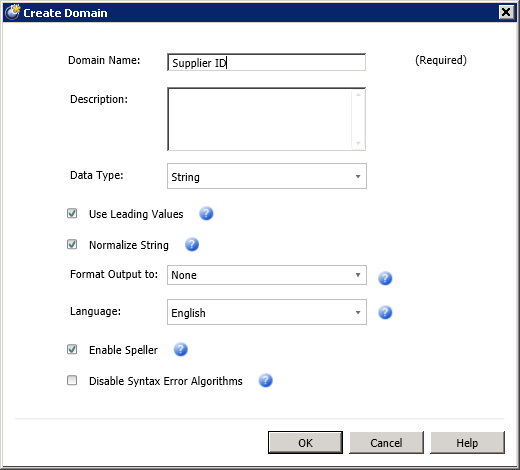
1. Enter **Suppliers** for **Name** of the knowledge base.



1. Confirm that **Create Knowledge Base** **from** field is set to **None** since you are creating the **Suppliers** knowledge base from scratch.
2. Confirm that **Domain Management** is selected for the **Activity** and click **Next**. The Domain Management activity lets you create and manage domains in the knowledge base.
3. In the **Domain Management** window, click **Create a domain** toolbar button to create a domain.



1. In the **Create Domain** dialog box, type **Supplier ID** for the **Domain Name**, and click **OK**.

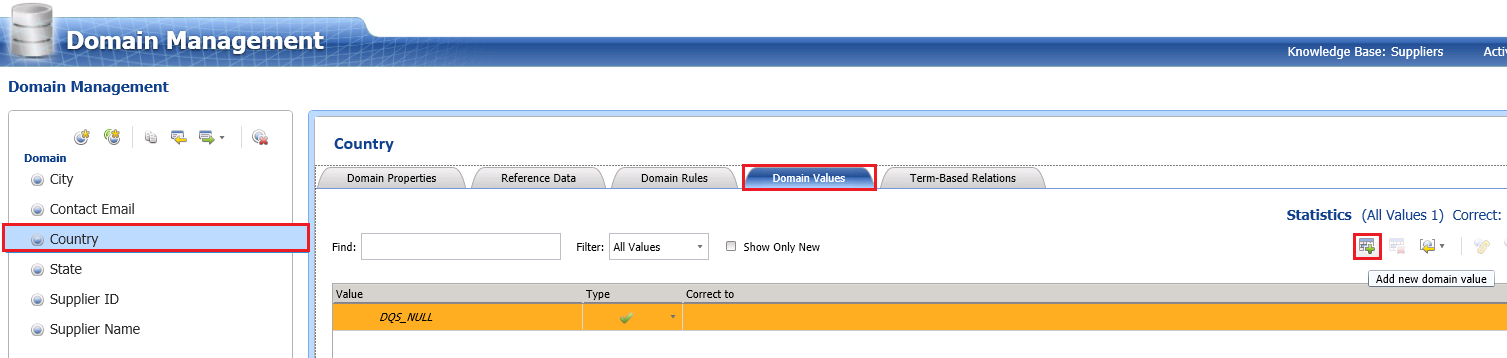


1. Repeat previous step to create the following domains with all the default settings. To keep the tutorial simple, you will keep the **Data Type** of all the domains as **String**. The other allowed data types are: Integer, Decimal, and Date. When the **Use Leading Values** option is selected (default), all synonyms are replaced with the leading value of the synonym group in the output. Setting **Normalize String** option (default) removes any special characters in the domain values. The **Format Output to** option lets you select the formatting that will be applied when the data values in the domain are output. Select **Enable Speller** (default) to run Speller on all string values when populating the domain. The **Language** setting specifies which language version of the **Speller** you want to apply. Select **Disable Syntax Error Algorithms** to populate the domain without checking string values for syntax errors. See [Create a Domain](http://msdn.microsoft.com/library/hh510401.aspx) topic in the MSDN library for more details.
   1. Supplier Name
   2. Contact Email
   3. Address Line
   4. City
   5. State
   6. Country
   7. Zip

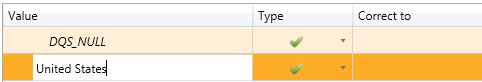
### Task 2: Adding Domain Values Manually

In this task, you will add a value for the **Country** domain manually. See [Change Domain Values](http://msdn.microsoft.com/library/hh510408.aspx) topic for more details about the fields on this page.

1. Click **Country** domain in the **Domain list**.
2. In the right pane, switch to the **Domain Values** tab.
3. Click **Add new domain value** button on the toolbar in the right pane.



1. Type **United States** for the **Value** field and press **ENTER**. You can see that, by default, the **Type** is set to **Correct** (green check). The Type can be set to **Error** (red cross) or **Invalid** (orange triangle), and a correct value can be entered in the **Correct To** field.



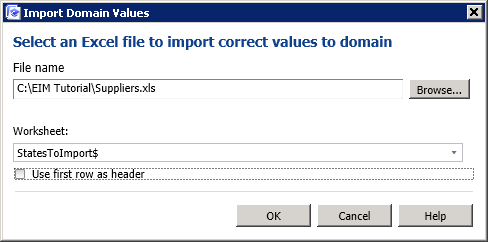
### Task 3: Importing Domain Values from an Excel File

In this task, you will import values for the **State** domain from a worksheet of an Excel file.

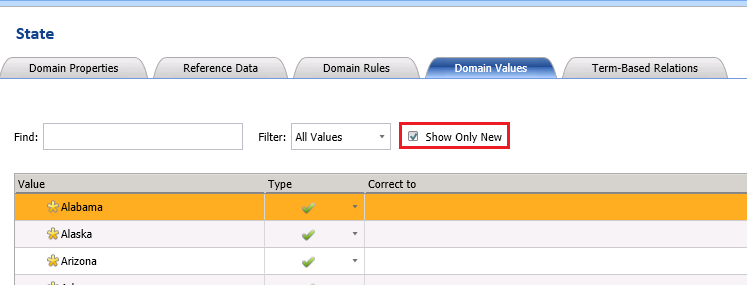
1. Click **State** domain in the **Domain list**.
2. Ensure that the **Domain Values** tab is active in the right pane.
3. In the right pane, from the toolbar, click **down arrow** next to the **Import Values** button, and click **Import Valid Values from Excel**.



1. Click **Browse**, select **Suppliers.xls**, and click **Open**.
2. Select **StatesToImport$** for the **Worksheet**.



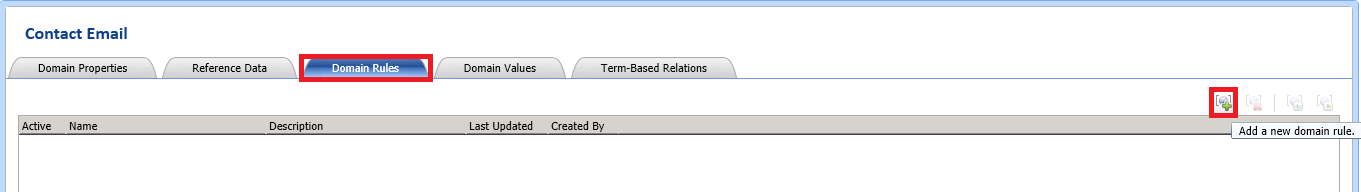
1. Click **OK** to close the **Import Domain Values** dialog box. You should see all the names of states you imported in the list. Notice that **Show Only New** option is automatically selected after importing. When you import values and you don’t see the old values in the list, it is because this option is automatically enabled after importing. To see all the values, just clear the check box. If you import the same set of values again, none of the values will be imported as they already exist in the domain.



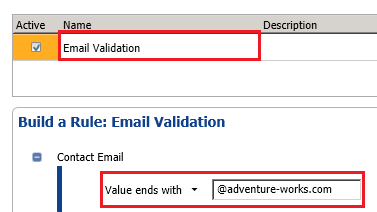
### Task 4: Setting Domain Rules

In this task, you will create a rule for the **Contact Email** domain to verify if the email address ends with **@adventure-works.com**. See [Creating a Domain Rule](http://msdn.microsoft.com/library/hh510397.aspx) topic for more details on the page.

1. Click **Contact Email** in the **Domain list**.
2. Switch to the **Domain Rules** tab in the right pane.



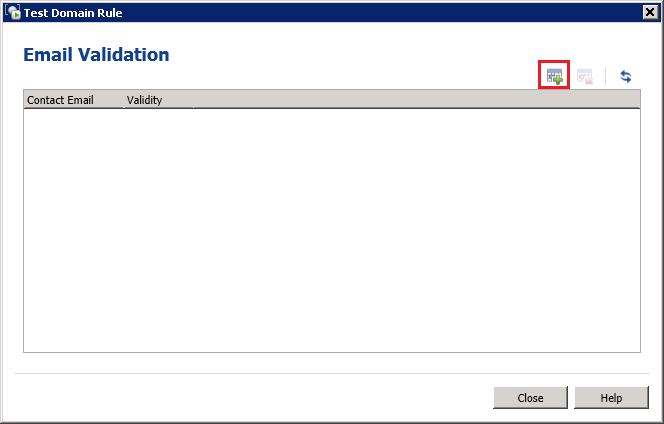
1. In the right pane, click **Add a new domain rule** toolbar button (see the image) to add a new rule.
2. Type **Email Validation** for the **rule name** and press **ENTER**. The **Active** check box should be checked by default. This control allows you to deactivate a rule temporarily.
3. In the **Build a Rule** pane, click **down arrow**, and select **Value ends with**.
4. Type **@adventure-works.com** in the text box and press **TAB**. You can add more conditions by clicking **Add a new condition to the selected clause** toolbar button in the **Build a Rule** pane. In this scenario, you will not be adding another condition.



1. Click **Run the selected domain rule on test data** button on the toolbar in the right pane to test the rule against sample data.



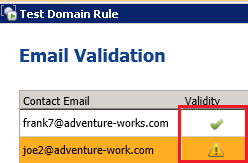
1. In the **Test Domain Rule** dialog box, click **Adds a new testing term for the domain rule** button on the toolbar.



1. Type **frank7@adventure-works.com** (a valid value) in the **Contact Email** column.
2. Repeat previous two steps to add **joe2@adventure-work.com** (an invalid value with no ‘s’).
3. Click the last button (**Test the domain rule on all the terms**) on the toolbar to test the input data against the rule.



1. Notice that the first entry is shown as a valid item and the second one as an invalid item.



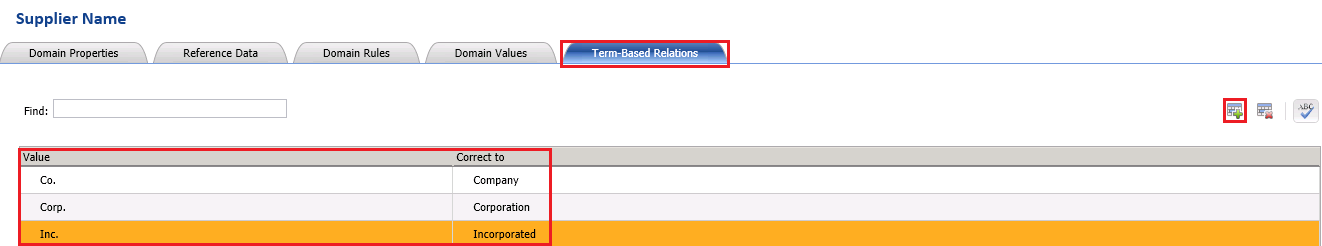
1. Click **Close** to close the **Test Domain Rule** dialog box.

### Task 5: Setting Term-Based Relations

In this task, you will define a few term based relations for values for the **Supplier Name** domain. A term-based relation enables you to make a correction to a term that is part of a value in a domain. It enables multiple values that are identical except for the spelling of a common part of them to be considered identical synonyms. For example, **Inc.** can be corrected to **Incorporated**. DQS will use these relations in the knowledge discovery, cleansing, or matching processes. See [Create Term-based Relations](http://msdn.microsoft.com/library/hh510404.aspx) for more details.

1. Select **Supplier Name** in the **Domain list**.
2. Switch to the **Term-Based Relationships** tab in the right pane.
3. Click **Add new relation** button on the toolbar to add a new relation to the table.
4. Type **Co.** for the **Value** field and **Company** for the **Correct To** field.
5. Repeat the previous two steps for the following values:

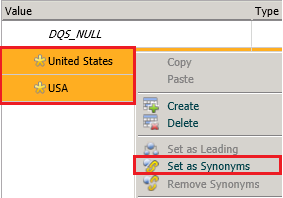
|  |  |
| --- | --- |
| **Value** | **Correct To** |
| Corp. | Corporation |
| Inc. | Incorporated |



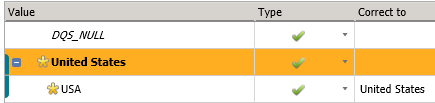
### Task 6: Setting Synonyms

In this task, you will set two domain values, **USA** and **United States**, of the **Country** domain as synonyms with **United States** as the leading value. Since the **Use Leading Values** option was selected when creating the **Country** domain, any **USA** values for the **Country** domain will be output as **United States** (as this is the leading value). See [Change Domain Values](http://msdn.microsoft.com/library/hh510408.aspx) for more details.

1. Select **Country** from the list of domains.
2. Switch to the **Domain Values** tab.
3. Click **Add new domain value** button on the toolbar.
4. Type **USA** for the value and press **ENTER**.
5. Multi-select **United States** and **USA** using CTRL or SHIFT keys, right-click the selected items, and then click **Set as Synonyms**. DQS will group these values and designate one of the values as the leading value that the other values will be replaced with.



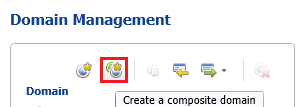
1. Notice that **United States** is set as the leading value. If you want USA to be the leading value, you can right-click on USA and select **Set as Leading** option. For this tutorial, we will use **United States** as the leading value.



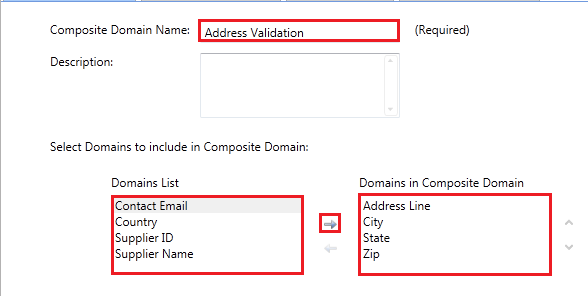
### Task 7: Creating a Composite Domain

In this task, you will create a composite domain, **Address Validation**, which comprises **Address Line**, **City**, **State** and **Zip** domains. A composite domain lets you define a cross-domain rule that involves multiple domains in a rule. There are other advantages to a composite domain such as being able to parse a field value into multiple domains. For example, a value for a Full Name field can be parsed into separate First Name, Middle Name, and Last Name domains. In this tutorial, we will only be defining a cross-domain rule. See [Managing a Composite Domain](http://msdn.microsoft.com/library/hh510399.aspx) for more details.

1. In the left pane, click **Create a composite domain** toolbar button.



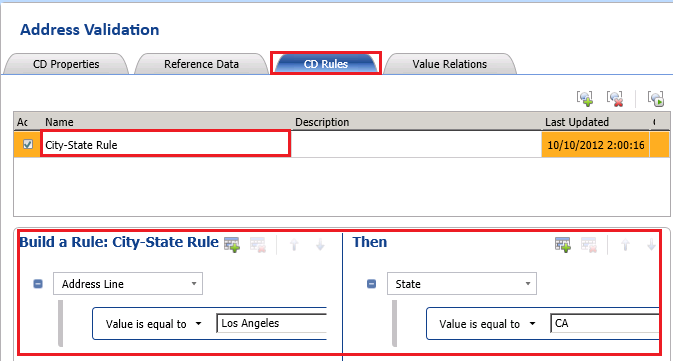
1. Enter **Address Validation** for the **Composite Domain Name**.



1. From the domain list select **Address Line**, **City**, **State, and Zip** and click **right arrow** to add them to the **Domains in Composite Domain** list.
2. Click **OK** to close the dialog box.

### Task 8: Creating a Composite Domain Rule

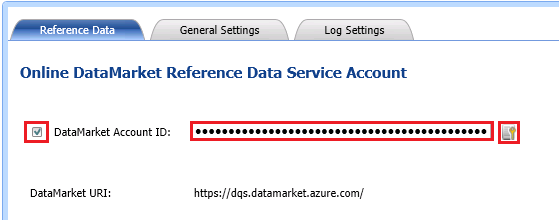
In this task, you will create a rule for the **Address Validation** composite domain. You will define a cross-domain rule: if **City** is **Los Angeles**, **State** must be **CA** where **City** and **State** are two domains.

1. In the right pane, switch to the **CD Rules** tab.
2. Click **Add a new domain** **rule** from the toolbar.
3. Type **City-State Rule** for **Name** and press **ENTER**.
4. In the **Build a Rule** pane, select **City** in the domain list, and select condition **Value is equal to** and type **Los Angeles** for the value.
5. In the **Then** pane, select **State** in the domain list, and select **Value is equal to**, type **CA** for the value, and press **TAB**. 
6. Click **Close** button at the bottom of the page to switch to the main page of DQS Client. You will publish the knowledge base in the next lesson. Notice that the KB is in locked state (lock icon).

### Task 9: Configuring a Reference Data Service

In this task, you will configure DQS to use a Reference Data Service on Windows Azure Marketplace. In the next task, you will be configuring the **Address Validation** domain to use this service. At runtime, during cleansing activity, DQS passes the values of domains in the **Address Validation** domain to the service for cleansing. See [Configure DQS to Use Reference Data](http://msdn.microsoft.com/library/hh213070.aspx) for more details.

1. In the main page of **DQS Client**, in the **Administration** pane, click **Configuration**.
2. Ensure that **Reference Data** tab is active.
3. In the **Network Settings** area, type appropriate values in the **Proxy Server** and **Port** fields if you need to use a proxy server to connect to internet.
4. Type your **Windows Azure** **Data Market (Marketplace) Account Key** for the **DataMarket Account ID** field.

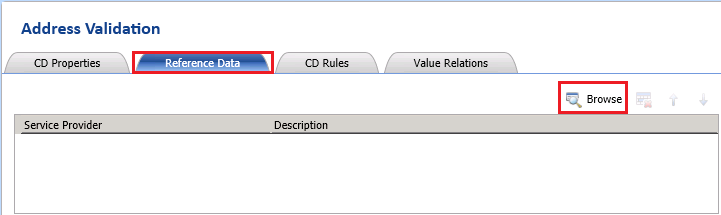


1. Click **Validate** button next to the text box to validate the account ID.
2. Click **OK** on the message box.
3. Click **Close** at the bottom of the page to switch to the main page of DQS Client.

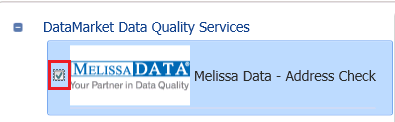
### Task 10: Configuring Composite Domain to Use Reference Data Service

In this task, you will configure the **Address Validation** composite domain to use the **Melissa Data – Address Check** service. At runtime, during cleansing activity, DQS passes the values of domains in the Address Validation domain to the service for cleansing. See [Map Domain/Composite Domain to Reference Data](http://msdn.microsoft.com/library/hh213030.aspx) for more details.

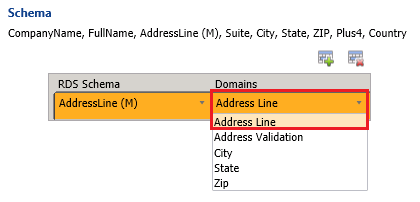
1. In the main page of **DQS Client**, click **Suppliers (Domain Management)** under **Recent Knowledge Bases** to launch the **Domain Management** page.
2. Select the **Address Validation** composite domain is selected in the **list of domains**.
3. In the right pane, switch to the **Reference Data** tab.



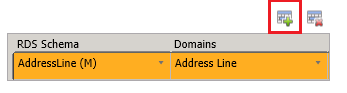
1. Click **Browse** button on the toolbar.
2. On the **Online Reference Data Providers Catalog** dialog box, select **check box** next to **Melissa Data – Address Check**.



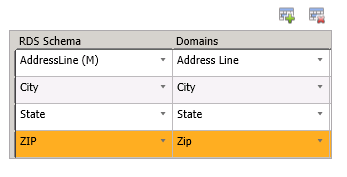
1. In the right pane, in the **Schema** section, map **Address Line** domain to the **Address Line (M)** schema item using the drop-down list.



1. Click **Add Schema Entry (+)** button on the toolbar to create a new entry in the list.



1. Map the following DQS domains using the drop-down lists as shown in the following picture.



1. Click **OK** to close the dialog box.

### Task 11: Publishing the Knowledge Base

In this task, you will publish the knowledge base. A published knowledgebase can be used for cleansing or matching activity in data quality project.

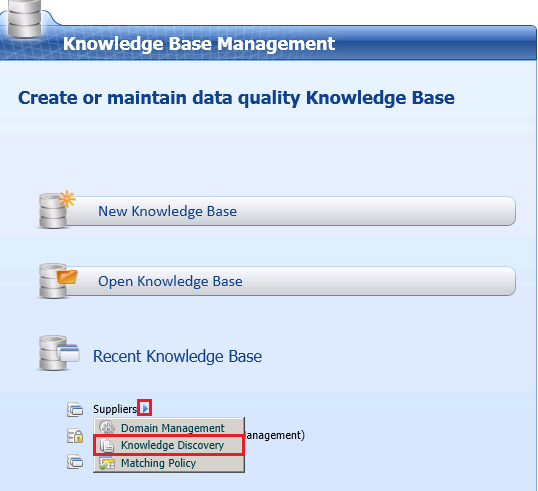
1. Click **Finish** button at the bottom of the window.
2. Click **Publish** in the **SQL Server Data Quality Services** dialog box.
3. Click **OK** to close the message box.

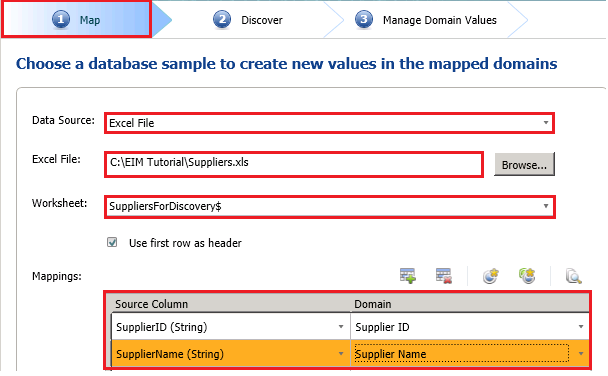
### Task 12: Discovering Knowledge (Knowledge Discovery)

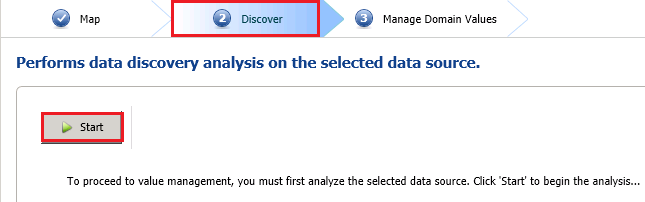
In this task, you will perform the **Knowledge Discovery** activity on **Supplier ID** and **Supplier Name** domains. In this scenario, the knowledge discovery process mainly imports values for these two domains.

In this tutorial, you started building knowledge base from scratch. You can also start creating a knowledge base by performing a knowledge discovery activity. When you click **Create a Knowledge** Base in the main page, DQS client takes you to a page with **Domain Management** activity selected for the activity. You can change the **activity** to **Knowledge Discovery** and then in the next page you can create domains as part of the knowledge discovery process. See [Perform Knowledge Discovery](http://msdn.microsoft.com/library/hh510398.aspx) for more details.

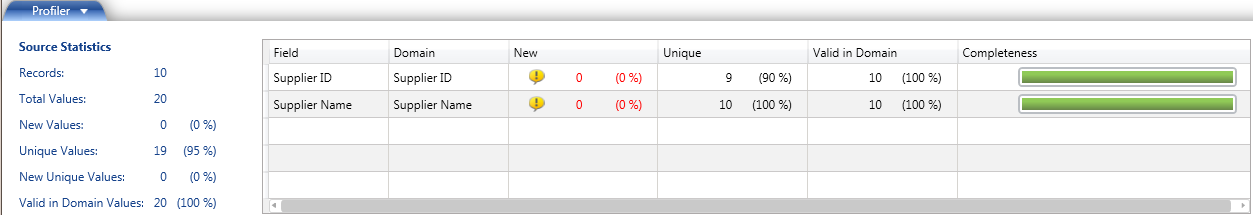
1. In the main page of DQS Client, in the **Recent Knowledge Base** section, click **right-arrow** next to the **Suppliers** KB and click **Knowledge Discovery**. Alternatively, you can click **Open Knowledge Base**, select **Suppliers** from the **KB list**, select **Knowledge Discovery** as **activity** and click **Next**.



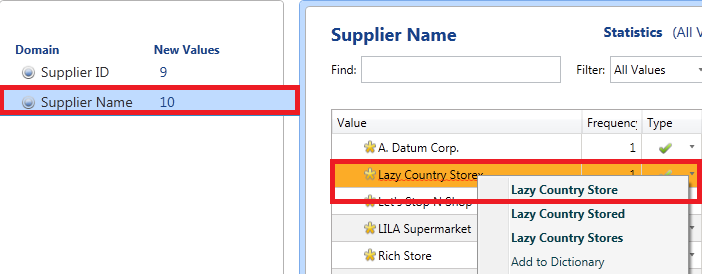
1. Select **Excel File** for **Data Source**.
2. Click **Browse**, navigate and select **Suppliers.xls**, and click **Open**.
3. Select **Suppliers for Discovery** for **Worksheet**.
4. In the **Mappings** section, map **SupplierID** column from the **Excel** file to the **Supplier ID** domain and **Supplier Name** column to the **Supplier Name** domain by using **drop-down lists**. The Excel file has sample data for the **Supplier ID** and **Supplier Name** domains. In the discovery process, you can select the domains for which you want to discover the values. Note that you create domains on this page and then map the source columns to those domains. It is not uncommon to create domains during knowledge discovery activity instead of creating domains during domain management activity.
5. Click **Next** to switch to the **Discover** page.
6. On the **Discover** page, click **Start** to start the discovery process. Discovery is performed on the columns **SupplierID** and **Supplier Name** in the **Suppliers.xls** file. The **Supplier ID** and **Supplier Name** domains will be populated with the knowledge drawn from the discovery.



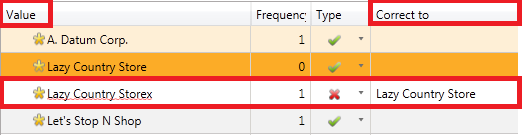
1. After the analysis has completed, review the **Source Statistics** in the **Profiler tab** at the bottom of the page. Notice that 10 new records with total 20 values (**SupplierID** and **Supplier Name** values from the **Excel worksheet**) were discovered. You will also see how many of the values are new, unique, new and unique, and valid. In the list box to the right, you can see more details for each domain involved in the discovery process. If you hover the mouse over the status bar in the Completeness column, you can see if there are any missing values in the columns in the source.



1. Click **Next** to switch to the **Manage Domain Values** page.
2. In the **Manage Domain Values** page, click **Supplier Name** domain from the list of domains.
3. In the right pane, right-click **Lazy Country Storex** (notice ‘x’ at the end), and select **Lazy Country Store**. DQS suggests this change after running the spell checker on the domain. By default, speller is enabled on the domains you create.



1. In the domain values list, confirm that the value **Lazy Country Storex** is set as an error (red **X** mark) with **Lazy Country Store** as the correction and also the **Lazy Country Store** is also added as a valid value.



1. Click **Finish**.
2. On **SQL Server Data Quality Services** dialog box, click **Publish**.
3. Click **OK** on the success message box.

## Lesson 2: Cleansing Supplier Data using the Suppliers Knowledge Base

In this lesson you will cleanse the supplier data in an Excel file using the **Suppliers** KB you have created in the first lesson. Data cleansing in DQS includes a **computer-assisted process** that analyzes how data conforms to the knowledge in a knowledge base, and an **interactive process** that enables you to review and modify results from the computer-assisted process. The data cleansing feature identifies incorrect data in your data source, and then corrects or suggests corrections for the incorrect data. It also standardizes and enriches customer data by using domain values, leading values for synonyms, domain rules, term-based relations, and reference data. You can interactively approve or reject changes proposed by the computer-assisted process. See [Data Cleansing](http://msdn.microsoft.com/library/gg524800.aspx) for more details.

The computer-assisted process uses the following threshold values that you can configure using the Configuration option on the DQS Client main page.

* **Min score for suggestions**: The minimum score or confidence level that will be used by DQS for suggesting replacement for a value.
* **Min score for auto corrections**: The minimum score or confidence level that will be used by DQS for automatically correcting a value.

See [Configure Threshold Values for Cleansing and Matching](http://msdn.microsoft.com/library/hh510415.aspx) for details on how to configure these settings.

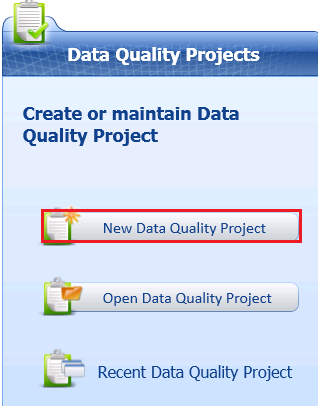
In this lesson, you will perform the following tasks to cleanse the input data using the Suppliers KB.

1. Create a Data Quality Project for Cleansing, select the Suppliers KB as the KB to use to analyze and cleanse the source data in an Excel file, and select the Cleansing activity.
2. Map Excel columns to be cleansed to appropriate DQS domains/composite domains in the knowledge base.
3. Run the computer-assisted cleansing activity. The computer-assisted process displays data quality information in the Data Quality Client that you can use to interactively cleanse the data.
4. View and manage the results of the cleansing activity. You can review the values that are found by the computer-assisted process to be correct, incorrect but corrected, incorrect with a suggested change, or invalid. You can interactively approve or reject changes, correcting or overriding the suggestion from the computer-assisted process using the Correct To field.
5. Export the results from the cleansing process to an Excel file.
6. Import the values from the cleansing project into domains to augment the knowledge in the knowledge base with new rules, values, corrections etc…

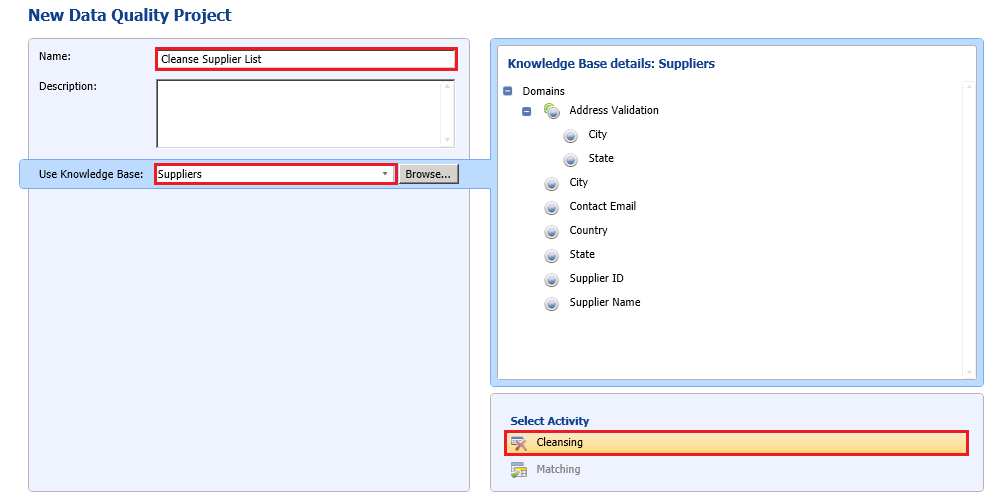
### Task1: Creating a Data Quality Project

In this task, you will create a Data Quality Project for cleansing the supplier data in an Excel file against the Suppliers knowledge base you created earlier in this tutorial.

1. In the **Data Quality Project** pane on the main page, click **New Data Quality Project**.



1. Type **Cleanse Supplier List** for the **name** of project.
2. **Important:** Select **Suppliers** for the **Use Knowledge Base** field. You will be cleansing input supplier data against the **Suppliers** knowledge base you created earlier in this tutorial.
3. Ensure that **Cleansing** is selected as the **activity** at the bottom of the right pane and click **Next**.

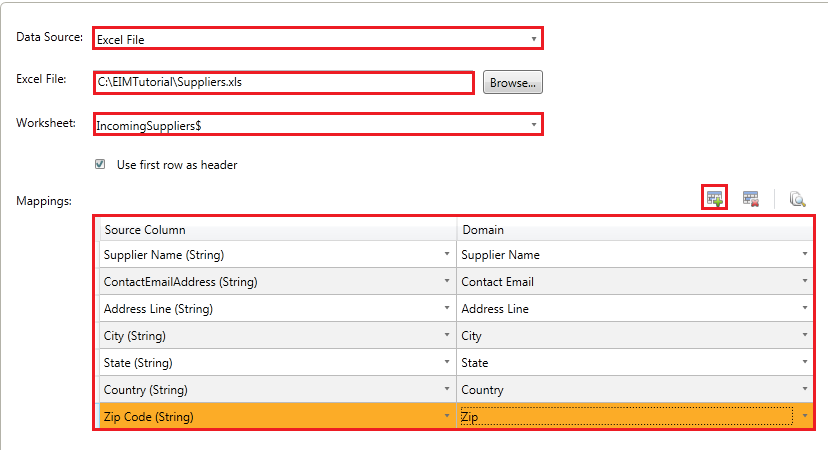


### Task 2: Mapping Excel Columns to DQS Domains

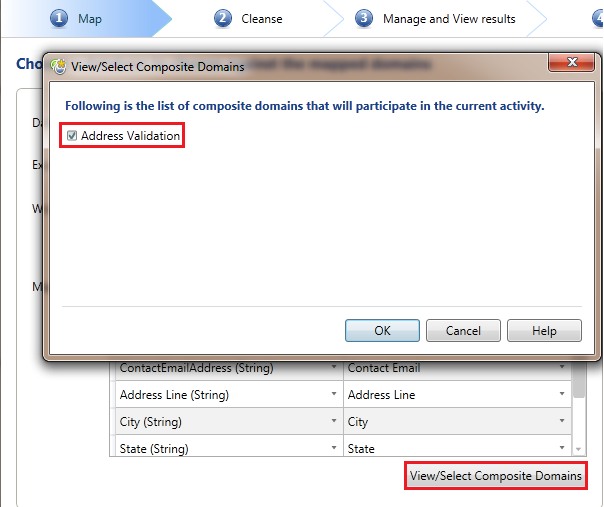
In this task, you will map columns in an Excel file to DQS domains in the **Suppliers** knowledge base.

1. In the **Map** page, select **Excel File** for **Data Source**.
2. Click **Browse**, select **Suppliers.xlsx**, and click **Open**.
3. Select **IncomingSuppliers$** for the **Worksheet**.
4. Map columns as shown in the following table and screenshot. When creating mappings for the **State** domain, click **Add a column mapping** toolbar button located just above the list. Note that you are not using **Supplier ID** column/domain for cleansing. You will use the **Supplier ID** domain later in the matching activity.

|  |  |
| --- | --- |
| **Excel column** | **DQS Domain** |
| Supplier Name | Supplier Name |
| ContactEmailAddress | Contact Email |
| Address Line | Address Line |
| City | City |
| State | State |
| Country (Click **+(Add a column mapping)** toolbar to add a new row) | Country |
| Zip Code | Zip |



1. As you have mapped all the individual domains within the **Address Validation** composite domain, the composite domain automatically participates in the cleansing process. Click **View/Select Composite Domains** button to see that the **Address Validation** composite domain is automatically selected, and then click **OK**.

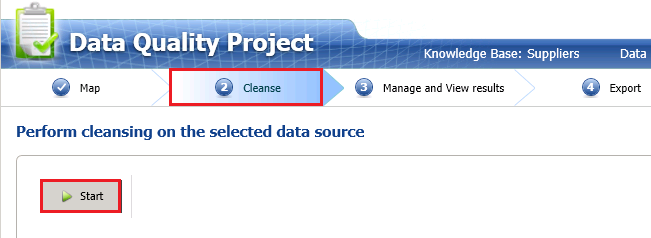


1. Click **Next** to switch to the **Cleanse** page.

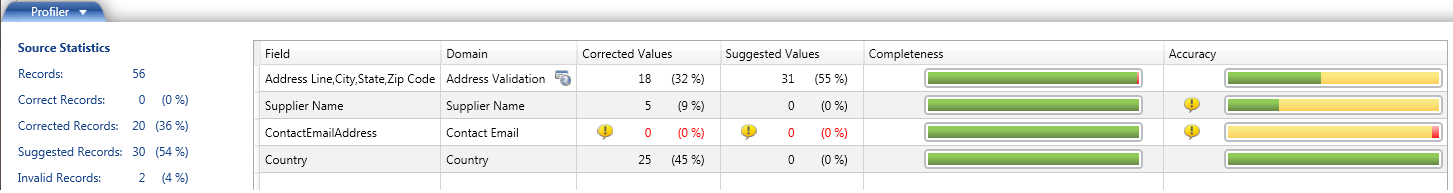
### Task 3: Cleansing Data against the Supplier Knowledge Base

In this task, you will run the computer-assisted cleansing process. DQS uses advanced algorithms and confidence levels based on the threshold values specified to analyze the data against the selected knowledge base, and then cleanse it. See [Cleansing Data Using DQS (Internal) Knowledge](http://msdn.microsoft.com/library/hh213061.aspx) for more details.

1. Click **Start** to start the computer-assisted cleansing process.



1. When the cleansing process is completed, review **statistics** in the **Profiler** tab. The Source Statistics provide the number of records processed, number of records that are found to be correct, number of records that are corrected by DQS, number of records that have changes suggested by DQS, and the number of records that are invalid. In the list box to the right, you can see the corrected values, suggested values, and the completeness (the extent to which the data is present) and accuracy (the extent to which the data can be used for intended purposes) of values for each domain involved in the cleansing process.

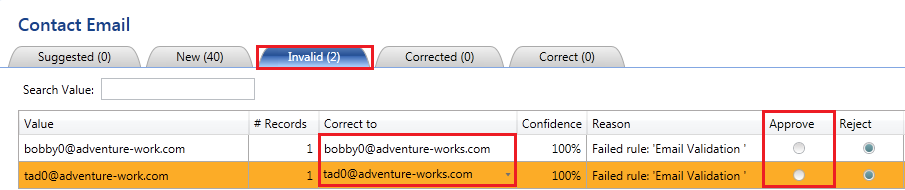


1. Click **Next** to switch to the **Manage and View Results** page.

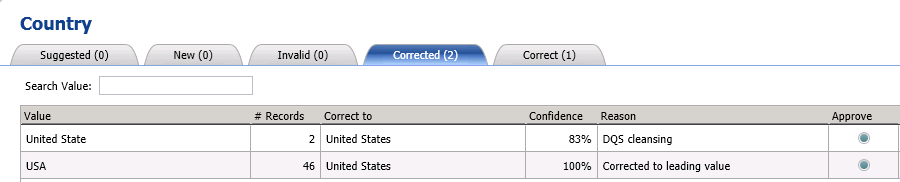
### Task 4: Managing and Viewing Results

In this task, you will review the results of computer-assisted cleansing and also perform interactive cleansing on the supplier data. See [Interactive Cleansing Stage](http://msdn.microsoft.com/library/hh213061.aspx#Interactive) for more details.

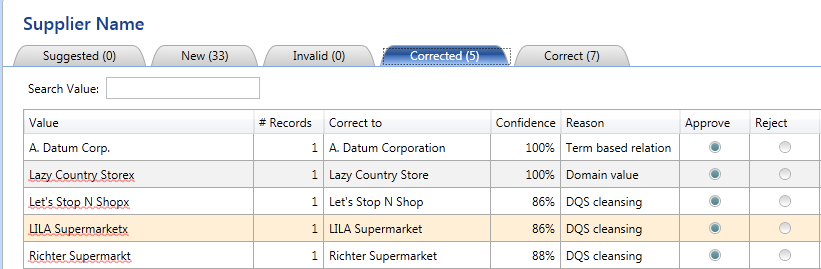
1. Select **Contact Email** domain from the list of domains.
2. Switch to the **Invalid** tab in the right pane. Notice that two emails address that were missing character ‘s’ at the end. These two emails were found to be invalid by the domain rule that requires all email addresses to end with **@adventure-works.com** (with ‘s’). DQS uses the domain rule while cleansing to determine whether an email is a valid one or not. This tab displays the domain values that were either marked as invalid in the KB or failed a domain rule. In this case, these values failed the domain rule (Email Validation).
3. In the **Correct To** column, type the right email address that end with **@adventure-works.com** (with ‘s’).



1. Click **Approve** for both the records to approve both the changes. When you approve, the records move to the **Corrected** tab. Instead of approving each item separately, you can approve all the changes at once using the **Approve all terms** toolbar button.
2. Switch to the **New** tab in the right pane. These are the values for which DQS does not have enough information in the KB yet to determine whether the values are correct. Therefore, it cannot make changes or suggest changes to the domain values.
3. Review the values to confirm that all the emails end with **@adventure-works.com** and click **Approve all terms** on the toolbar. The approved values from this tab move to the **Correct** tab.
4. Select the **Country** domain from the list of domains.
5. Switch to the **Corrected** tab in the right pane and notice that **United State** value is automatically corrected to the **United States** with ‘s’ at the end. This is not a rule you defined for the **Country** domain, but DQS is **83%** confident that the correct value is **United States**. The **Approve** button is automatically selected for all the **Corrected** items. You can override this and reject a change if needed.
6. Notice that **USA** is corrected to **United States** because they are synonyms and **United States** is the leading (preferred) value.

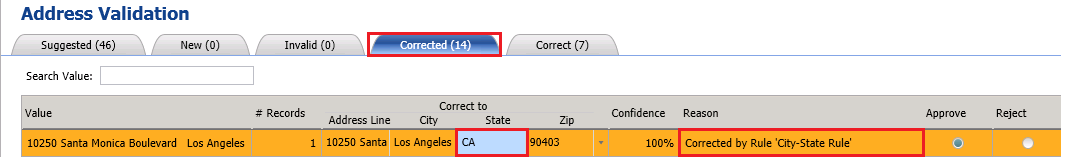


1. Notice that the **Approve** button is already selected for these corrected values. This is the default behavior for the corrected values. You can reject a change and when you do so, the value moves to the **Invalid** tab.
2. Select **Supplier Name** from the list of domains.
3. Switch to the **Corrected** tab in the right pane.



* 1. Notice that **A. Datum Corp.** is corrected to **A. Datum Corporation** and the **Reason** is set to **Term based relation**. **A. Datum Corporation** is a known domain value to DQS because it was discovered during the knowledge discovery process. Therefore, DQS is **100%** **confident** about this correction.
  2. Notice that that **Lazy Country Storex** is corrected to **Lazy Country Store**, **Confidence Level** is set to **100%**, and the **Reason** is set to **Domain Value**. During the knowledge discovery process, you set **Lazy Country Storex** as an error with **Lazy Country Store** as the **correction**, so DQS is **100% confident** about making this correction.
  3. DQS is not familiar with the other values in the list, but it found the corrections for these values using the **Spell Checker** and proposes the appropriate corrections. DQS is **not 100%** confident about these corrections, but the confidence level is above 80%, which is the threshold level for making corrections, so DQS proposes the corrections.

1. Notice that the **Approve** is automatically enabled for all the values. You can override the corrected value or reject the change as appropriate. By default the **Approve** button is selected for all the values on the **Corrected** tab.
2. Switch to the **New** tab.
3. Notice that **Corp.** is corrected to **Corporation**, **Co.** is corrected to **Company**, and **Inc.** is corrected to **Incorporated**. For example, **Consolidate Inc.** is corrected to **Consolidate Incorporated** and **Consolidated Co.** is corrected to **Consolidated Company**, and **Frabrikam Corp.** is corrected to **Fabrikam Corporation**. You can see that **term-based relation** is mentioned as the reason. These changes are proposed by using the term-based relations you defined during the domain management activity. You can change the **Correct To** values manually here.
4. Scroll the list to see **Hunxgry Coyote Store** with a red squiggly line. Right-click on it and click **Hungy Coyote Store** (with no ‘x’). The **Correct To** column should be automatically populated with **Hungry Coyote Store**. You can also manually type a value in the **Correct To** column.
5. Click **Approve all terms** from the toolbar. The domain values with the **Correct To** value specified move to the **Corrected** tab and the new values with no associated **Correct To** values move to the **Correct** tab.
6. Select the **Address Validation** composite domain from the domain list.
7. In the right pane, switch to the **Correct** tab. You should see the addresses that are found to be correct by the **Melissa Data – Address Check** DQS service on the **Azure Marketplace**.
8. Switch to the **Corrected** tab.
9. Notice that **State** for the record that has **City** as **Los Angeles** is set to **CA** now. Notice in the **Reason** field is that **Corrected by Rule ‘City-State Rule’**.



1. Notice that the **Approve** radio button is already selected for this item in the list. This is the default behavior for items on the **Corrected** tab.
2. Switch to the **Suggested** tab. Review the changes suggested by the **Melissa Data – Address Check** service.
3. **Click Approve all terms** on the toolbar button and click **OK** on the **Confirmation** message box.

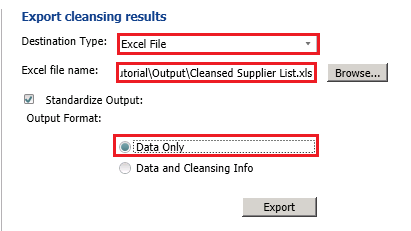


1. Click **Next** to switch to the **Export** page.

### Task 5: Exporting Cleansing Results to an Excel File

In this task, you will export results from the cleansing activity to an Excel file. See [Export Stage](http://msdn.microsoft.com/library/hh213061.aspx#Export) topic for more details.

1. In the right pane, select **Excel** for the **Destination Type**.
2. Click **Browse**, specify the output file name as **Cleansed Supplier List.xls**, and then click **Open**.
3. Select **Data Only** for the **Output** format to export just the cleansed data. The second option, **Data and Cleansing Info**, lets you export cleansing activity details along with the cleansed data. The **Standardize Format** option lets you apply any output formats you define on a domain to the values of that domain. You have not defined an output format on any domain in the tutorial.

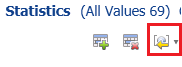


1. Click **Export** to export the data. Do not click **Finish** yet.
2. Click **Close** on the **Exporting** dialog box.
3. Click **Finish** to finish the activity. If you had forgotten to export results before clicking **Finish**, click **Open Data Quality Project** in the main page of **DQS Client**, select **Cleanse Supplier List** from the list of projects, and click **Next** at the bottom of the screen to get to the **Export** stage of cleansing process again. You can also switch to the **Manage and View Results** tab by clicking **Back** button.
4. Open the **Cleansed Supplier List.xls** and do the following:
   1. Ensure that there are no email address that end with adventure-work.com (without character ‘s’) by searching for adventure-work.com in the worksheet.
   2. See that there is no **USA** value in the **Country** column.
   3. Search for **Los Angeles** and see that the **State** is set to **CA**.
   4. Confirm that there are no terms **Co.**, **Corp.**, and **Inc.**
   5. **Important**: delete the **Address Validation** column from the spreadsheet and save the excel file. This additional column corresponds to the Address Validation composite domain.

### Task 6: Importing Values from the Cleanse Supplier List Project

In this task, you will import the data quality knowledge gathered during the cleansing process. See [Importing Cleansing Project Values into a Domain](http://msdn.microsoft.com/library/hh479581.aspx) topic for more details. You will also export the knowledge base into a DQS file before publishing the updated **Suppliers** KB.

1. In the main page of **DQS Client**, click **right-arrow** next to **Suppliers** under **Recent Knowledge Bases** and click **Domain Management**.
2. Click **Contact Email** in the list of domains, and switch to the **Domain Values** tab in the right pane.
3. Click **down arrow** next to the **Import Values** icon on the toolbar and click **Import Project Values**.



1. On the **Import Project Values** dialog box, select the **Cleanse Supplier List** project, and click **OK**.
2. Notice that all the emails are imported along with the two corrections you did during interactive cleansing. Scroll to see the two corrections.

|  |  |
| --- | --- |
| **Value** | **Correct To** |
| bobby0@adventure-work.com | bobby0@adventure-works.com |
| tad0@adventure-work.com | tad0@adventure-works.com |

1. Repeat the previous step of importing project values for the **Country** domain and notice that a new entry is added for correcting **United State** to **United States** (with ‘s’)

|  |  |
| --- | --- |
| **Value** | **Correct To** |
| United State | United States |

1. To see the old domain values, clear **Show Only New** checkbox.
2. Repeat the previous step of importing project values for the **Supplier Name** domain. By default, after importing, you will only see the new values. To see all the values, clear **Show Only New** check box. We have enriched the **Suppliers** KB with what we learned from the cleansing activity. The stronger the KB is, the better the cleansing results are. Note that it is not possible import values for a composite domain.
3. Click **Export Knowledge Base** icon on the toolbar and then click **Export Knowledge Base**.



1. Navigate to the Tutorial folder, type **Suppliers.dqs** for the **file name**, and click **Save**. You can use this DQS file to create a new knowledge base based on it.
2. Click **OK** to close the **Export Knowledge Base – Suppliers** message box.
3. Click **Finish** to finish the activity.
4. Click **Publish**.
5. Click **OK** on the message box.

## Lesson 3: Matching Data to Remove Duplicates from Supplier List

You prepare the knowledge base for performing matching activity by creating a matching policy in the knowledge base. There can be only one matching policy in a knowledge base. A matching policy consists of one or more matching rules. A rule identifies the domains that will be involved in the matching process, and specifies the weight that each domain value carries in the matching judgment. You specify in the rule whether domain values have to be an exact match or can just be similar, and to what degree of similarity. You also specify whether a domain match is a prerequisite for the matching process. You can test each rule separately and test the entire policy against sample data. The testing process displays records whose matching scores are greater than the **Min record score** threshold specified in the DQS configuration in a cluster (group). You can continue to tweak the rules in the policy until you are satisfied.

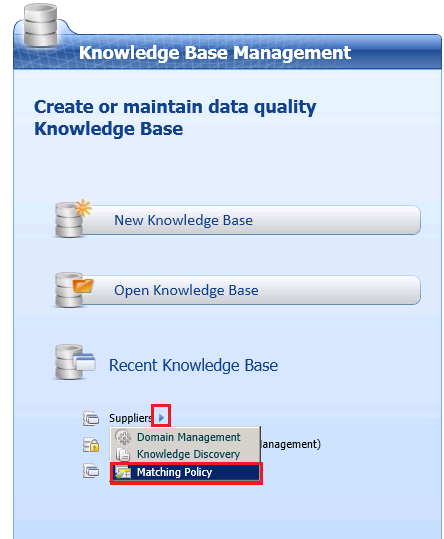
After defining the policy, you create a Data Quality Project to run the matching activity. The matching project applies the matching rules in the matching policy to the data source to be assessed. This process assesses the likelihood that any two rows are matches. When DQS performs the matching analysis, it creates clusters of records that DQS considers matches. DQS randomly identifies one of the records as a pivot record. You can verify and reject any record that is not an appropriate match for the cluster. See [Create a Matching Policy](http://msdn.microsoft.com/library/hh270290.aspx) topic for more details.

In this lesson, you will perform a matching activity to remove duplicates from the supplier list. First, you will create a matching policy with one rule to identify duplicates in the supplier list and publish the policy to the knowledge base. Next, you will create and run a data quality project for matching. Finally, you will export the results from the matching activity to an Excel file that you will use later in uploading data to Master Data Services (MDS).

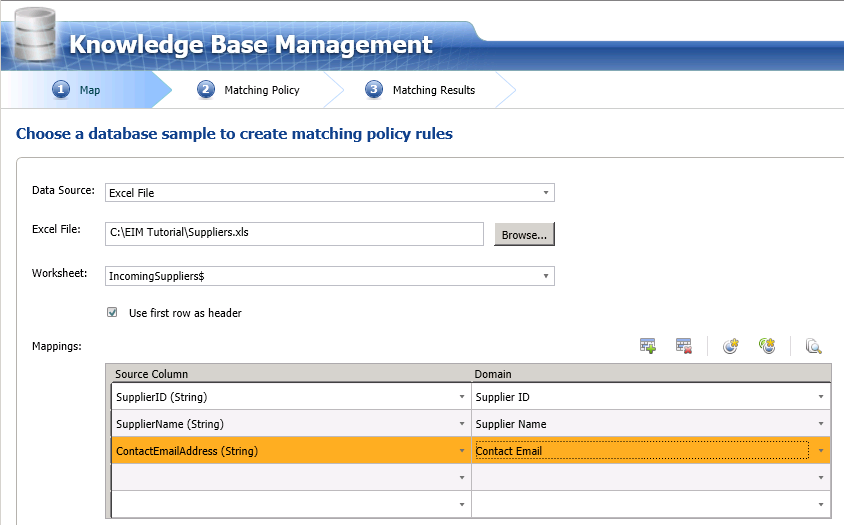
### Task 1: Defining a Matching Policy

In this task, you will create a matching policy with one rule in it. The rule will have one prerequisite: **Supplier ID**, which means that the Supplier IDs must match before using the other domains in the rule. The rule uses two other domains: **Supplier Name** with **Similarity** value set to **70%** and **Contact Email** with **Similarity** value set to **30%**.

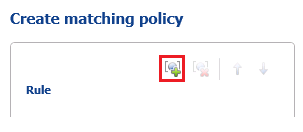
1. In the main page of **DQS Client**, click **right-arrow** next to **Suppliers** KB, and select **Matching Policy**.



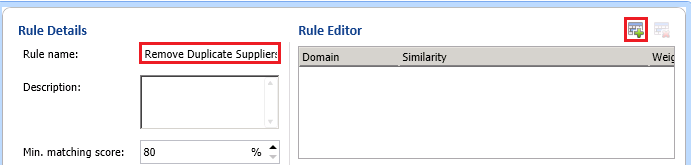
1. Select **Excel File** for **Data Source** in the **Map** page.
2. Click **Browse**, ensure filter is set to **Excel Workbook**, and select **Cleansed Supplier List.xls** file that you exported after performing the cleansing activity. Note that at the end of this activity, you will not be able to export results because this activity is primarily focused on defining a matching policy. You will create a Data Quality Project for the Matching activity and run it to remove duplicates from the supplier list using this matching policy in the next lesson.
3. Map **SupplierID** column to **Supplier ID** domain, **Supplier Name** column to **Supplier Name** domain, **ContactEmailAddress** column to **Contact Email** domain. You only need to map source columns to domains that you want to use in defining the matching policy. In this case, you are making the Supplier ID, Supplier Name, and Contact Email domains available for the matching policy activity.



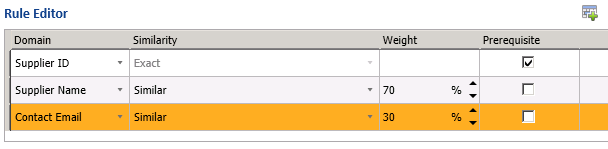
1. Click **Next** to move to the **Matching Policy** page where you will be defining a matching policy with one rule in it.
2. Click **Create a matching rule** button on the toolbar to create a rule in the policy.



1. In the **Rule Details** pane on the right, enter **Remove Duplicate Suppliers** for the **Rule name**.
2. Click **Add a new domain element** in the toolbar in the right pane.



1. Select **Supplier ID** for the **domain** and select the **Prerequisite** check box. Notice that **Similarity** is automatically set to **Exact**. By setting **Supplier ID** as the **Prerequisite**, you specify that the values for this field in the two records must return a 100% match, else the records are not considered a match and the other clauses in the rule are disregarded.

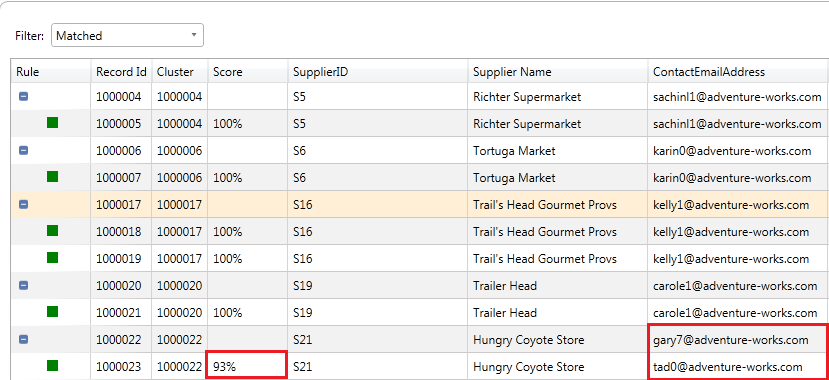


1. Click **Add a new domain element** from the toolbar again.
2. Select **Supplier Name** domain, select **Similar** for **Similarity**, and Type **70** for the **Weight**. Here, you are specifying that supplier names do not need to be identical but can be similar for the records to be considered as a match. The weight indicates the contribution of this field’s score to the overall matching score.
3. Repeat steps 10-11 to add **Contact Email** domain with **30** for the **Weight**.
4. Notice that the **min matching score** is set to **80%**, which is the value you see in the **General** tab of the **Configuration** page of **DQS Administration**. You can only increase this score above this threshold value here.
5. Notice that **Overlapping Clusters** option is selected. With this option, a record can show up in multiple clusters. If you change the setting to Non Overlapping Clusters, the clusters that have common records are combined into one single cluster.
6. The **Start** button on this page allows you to test each rule in the policy separately, whereas, the **Start** button in the next page allows you to test entire policy (all the rules in the policy).
7. Click **Next** to switch to the **Matching Results** page.

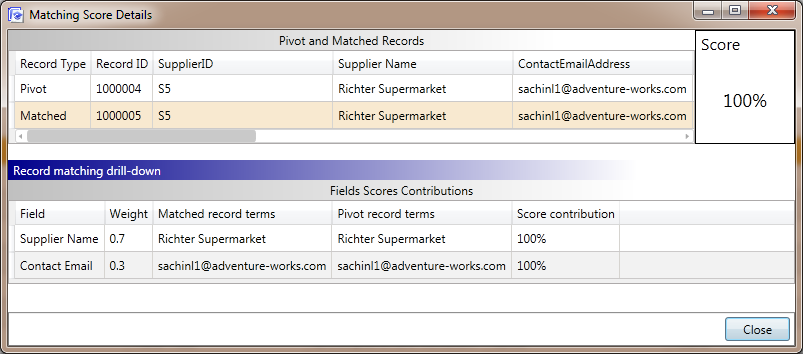
### Task 2: Testing and Publishing the Matching Policy

In this task, you will test and publish the **Remove Duplicate Suppliers** matching policy.

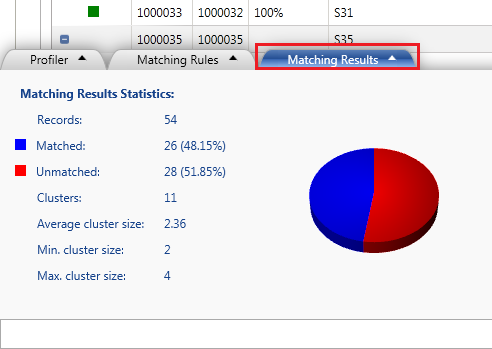
1. In the **Matching Results** page, click **Start** to test the entire policy. In our case, we have only rule in the policy, so the results from testing the rule and the policy should be the same.
2. Review all the matched records and their matching score in the list box. A record that has a **Green** icon associated with it is a duplicate of the pivot record that precedes it. Here are couple of examples:
   1. The record with **Record ID**: **1000005** is a match of the record with **Record Id**: **1000004** with **Score**: **100%** because both the records have the same values for **SupplierID (prerequisite)**, **Supplier Name**, and **ContactEmailAddress columns**. DQS randomly picks a record as the pivot record for a cluster.
   2. The record **1000023** is a match of the record **1000022** with the matching score: **93%** because the two records have the same values for **SupplierID (prerequisite)** and **Supplier Name** columns, but different values for the **ContactEmailAddress** column.
   3. Scroll to the bottom of the list to see two records with records IDs: **1000051** and **1000052**. Record **1000052** is considered a match with matching score **91%** because the two records have the same values for the **SupplierID** and **ContactEmailAddress** columns, but different values for the **Supplier Name** column.



1. Right-click on any matched record (with green icon) and click **View** **Details** to see more details about the matching such as contribution of each field score to the overall matching score.



1. Click **Close** to close the **Matching Score Details** dialog box.
2. Click **Matching Results** tab at the bottom of the page. This window gives you detail such as number of matched records, number of unmatched records, number of clusters with matched records, the average cluster size, minimum cluster size, and maximum cluster size. See [Create a Matching Policy](http://msdn.microsoft.com/library/hh270290.aspx) for more details. NOTE: you will not be able export results from this activity. You are just defining a matching policy using the sample data to test rules and the policy against the sample data.



1. Click **Finish** to finish creating the matching policy.

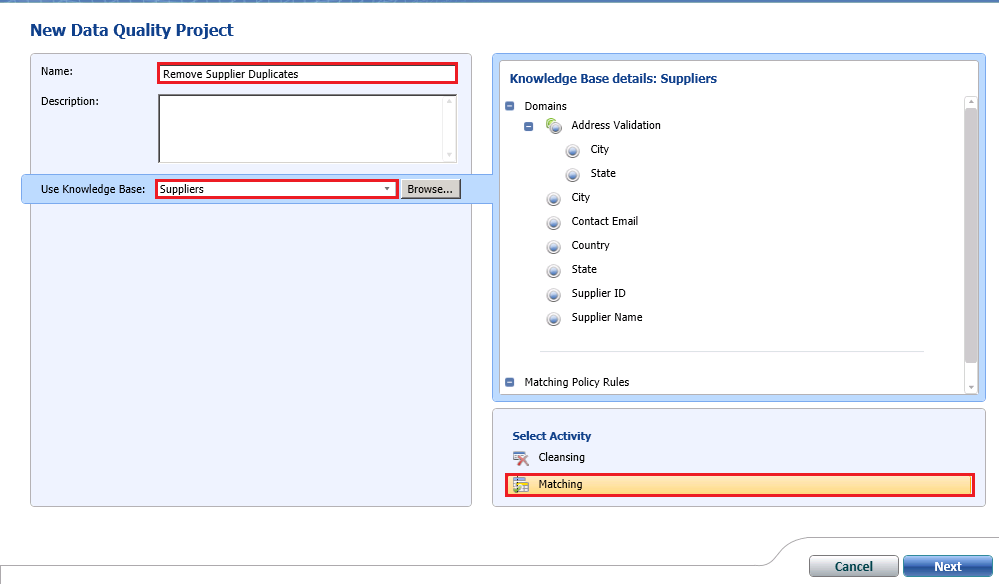
NOTE: You have defined the matching policy here; therefore you cannot export results to an output file. You basically used a sample input file, created rules, and tested the rules and policy against the sample data with the goal of defining the policy.

1. Click **Publish** on the SQL Server Data Quality Services dialog box and click **OK** on the message box. Now, the matching policy you defined is published into the **Suppliers** Knowledge Base. You can use the knowledge base to run the matching process against an input file to identify and remove duplicates.

### Task 3: Creating and Running a Data Quality Project for Matching

In this task, you will create a Data Quality Project for the matching activity and run the matching process on cleansed supplier data to remove any duplicates in the data.

1. On the main page of **DQS Client**, click **New Data Quality Project**.
2. Type **Remove Supplier Duplicates** from the **Name of the project**.
3. **Important:** Select **Suppliers** from the list of KBs for the **Use Knowledge Base** field. You have created a matching policy in this knowledge base in the previous lesson.
4. **Important:** Select **Matching** from the **list of activities** from the bottom-right pane.

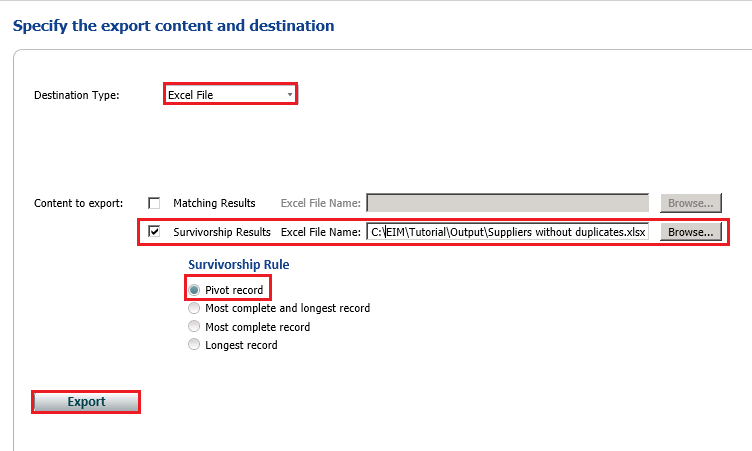


1. Click **Next**.
2. In the **Map** page, select **Excel File** for the **Data Source**.
3. Click **Browse** and select **Cleansed Supplier List.xls**, which is the output file from the cleansing activity.
4. Map **SupplierID** source column to the **Supplier ID** domain, **Supplier Name** column to **Supplier Name** domain, and **ContactEmailAddress** column to **Contact Email** domain.
5. Click **Next** to switch to the **Matching** page.
6. Click **Start** to start the matching process. You will see results similar to those from the previous task because you used the same input file for defining the matching policy.
7. Review all the matched records and their matching score in the list box. The results should be same as the ones you saw in the previous task. See the steps in the previous task to analyze the results from this matching activity.
8. Click **Next** to switch to the **Export** page.

### Task 4: Exporting the Results from Matching Activity to an Excel File

In this task, you will export the results from the matching activity to an Excel file.

1. In the **Export** page, select **Excel File** for the **Destination Type**.
2. Select **Survivorship Results** option. In the survivorship process, DQS determines a survivor record for each cluster based on the **Survivorship Rule** you selected.
3. Click **Browse** and navigate to the folder where you want to store the output file.
4. Type **Cleansed and Matched Suppliers.xls** for the name and click **Open**.
5. Confirm that **Pivot Record** is selected for the **Survivorship Rule**. When you select this option, the pivot record for each cluster is picked for the output from a cluster. The other options for the Survivorship Rule are:
   1. **Most complete record**: Survivor record is the one with the largest number of populated fields.
   2. **Longest record**: Survivor record is the one with the largest number of terms in source fields.
   3. **Most complete and longest record**: Survivor record is the one with the largest number of populated fields, and has the largest number of terms in each field.



1. Click **Export** to export the results to excel file.
2. Click **Close** to close the **Matching Export** dialog box.
3. Click **Finish** to finish the matching activity.
4. Open the **Cleansed and Matched Suppliers.xlsx** file and confirm that you do not see any duplicates (SupplierID).

Now, we have supplier data that has been cleansed and matched to remove duplicates.

## Lesson 4: Storing Supplier Data in MDS

Master Data Services (MDS) is the SQL Server solution for master data management. Master data management (MDM) describes the efforts made by an organization to discover and define non-transactional lists of data.

Models are the highest level of organization in Master Data Services and organize the structure of your master data. Your MDS implementation can have one or many models where each model groups similar kinds of data. In general, master data can be categorized in one of four ways: people, place, things, or concepts. For example, you can create a Product model to contain product-related data or Customer model to contain customer-related data. See [Models (Master Data Services)](http://msdn.microsoft.com/library/ee633746.aspx) for more details.

A model can contain one or more entities. Each entity has attributes (columns) and members (rows). Each row contains the master data. In this lesson, you will create a Suppliers model with two entities named Supplier and State. The Supplier entity will have the following attributes: Code, Name, Contact First Name, Contact Last Name, Contact Email Address, Address Line, City, State, Zip, and Country. See [Attributes (Master Data Services)](http://msdn.microsoft.com/library/ee633745.aspx) for more details about attributes in general. The Code and Name attributes correspond to the SupplierID and Supplier Name columns in the Cleansed and Matched Suppliers Excel file.

A domain based attribute is an attribute with values that are populated by members of another entity. Domain-based attributes prevent users from entering attribute values that are not valid. An attribute values can be selected only from the drop-down list that is populated by another entity. In this tutorial, the State attribute of the Supplier entity is a domain based attribute with values from the State entity. You can only change the value of the State attribute of the Supplier entity to one of the values in the State entity. See [Domain-Based Attributes](http://msdn.microsoft.com/library/ff487058.aspx) for more details.

A derived hierarchy in MDS is derived from the domain-based attribute relationship in the model. In this tutorial, you will create a derived hierarchy between the Supplier entity and the State entity. After you create the derived hierarchy, you will see a list of states in the Browser of Master Data Manager. When you click on a state in the list, you will see the suppliers in that state in the right pane. You will be creating a derived hierarchy later based on this relationship. See [Derived Hierarchies](http://msdn.microsoft.com/library/ee633747.aspx) for more details.

You built a knowledge base in DQS and used it to cleanse and match supplier data and stored the results in the Cleansed and Matched Supplier Data.xls file. In this lesson, you will upload the cleansed and matched data into MDS. Note that DQS only contains knowledge about the data (metadata) whereas MDS stores the data itself (master set). For example: DQS may have knowledge about several suppliers but MDS only maintains the suppliers that a company uses.

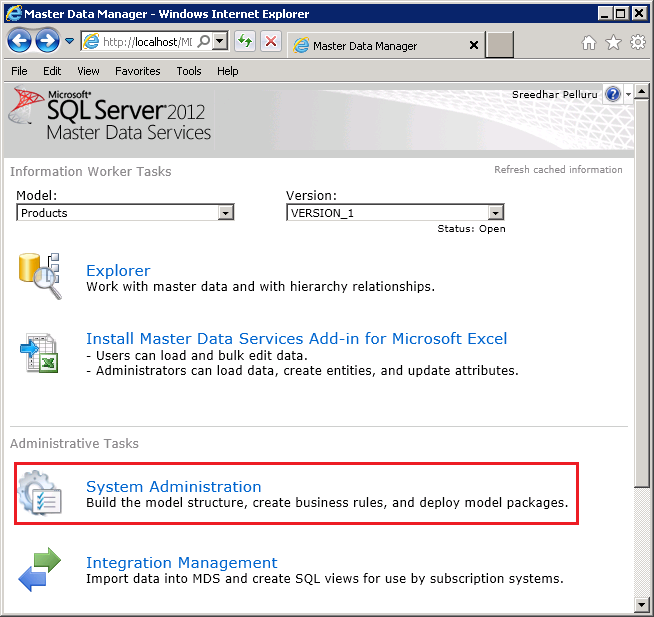
In this lesson, you will perform the following tasks:

1. Create the **Suppliers** model in **MDS** by using the **Master Data Manager Web Application**.
2. Open **Cleansed and Matched Supplier Data.xls** in Excel and use the **MDS Add-in for Excel** to create an entity named **Supplier** and upload the data to MDS.
3. Verify that the data is created in MDS by using the **Master Data Manager**.
4. Create a new entity named **State** and update the **State** attribute of **Supplier** entity to be a domain-based attribute depending on the **State** entity. You will do this all using the **MDS Add-in for Excel**.
5. Verify that the domain based attribute is created by using **Master Data** **Manager** and update the values for the **Name** attribute of the **State** entity.
6. View the updates you made using **Master Data Manager** in **Excel**.
7. Load values from the **State** entity into **Excel** and add a new value, and verify the addition by using **Master Data Manager**.
8. Create and use a derived hierarchy using the domain-based attribute relationship between the **Supplier** entity and the **State** entity (the State attribute of the Supplier entity is of type State entity) by using **Master Data Manager**.

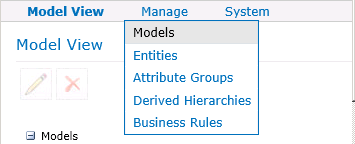
### Task 1: Creating Suppliers Model using Master Data Manager

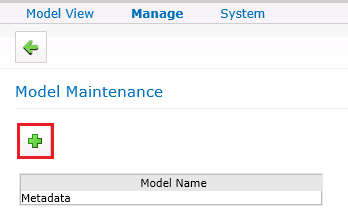
In this task, you will create a model named **Suppliers** in MDS using **Master Data Manager**.

1. Navigate to <http://localhost/mds> to launch **Master Data Manager**. Replace the URL if you have configured Web Application with a different name or on a different a Web site.

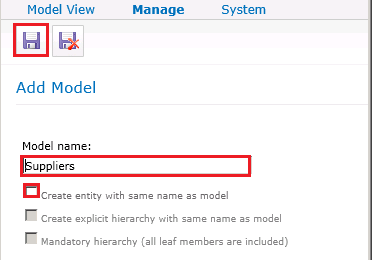


1. Click **System Administration** in the **Administrative Tasks** section.
2. If you do not see the **Add Model** page, hover mouse over **Manage** on the menu bar, click **Models** and then click **Add Model (+)** toolbar button to create a new model.





1. Enter **Suppliers** for **Model name**.
2. Clear **Create entity with same name as model** option. We will be creating an entity later using the **MDS Add-in for Excel**.

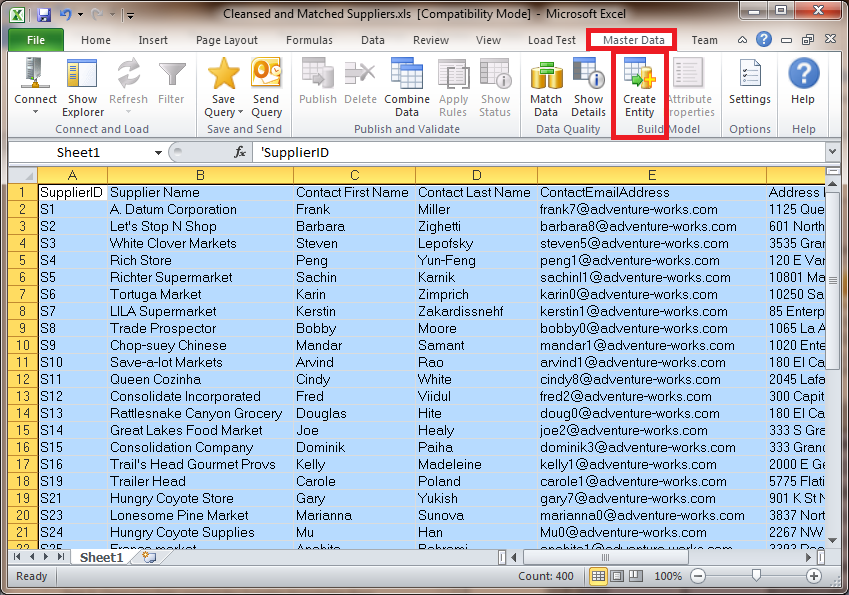


1. Click **Save Model** button on the toolbar.

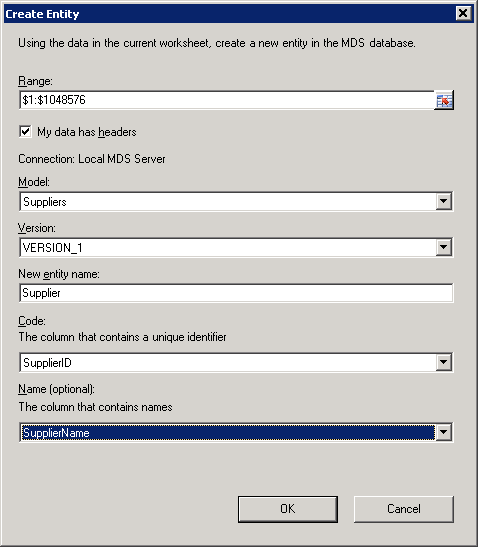
### Task 2: Uploading Supplier Data to MDS using MDS Add-in for Excel

In this task, you will publish the cleansed and supplier data to **MDS** using the **MDS Add-in for Excel**. You will create an entity named **Supplier** in the **Suppliers** model you created in the previous lesson. The entity will have an attribute for each column in the Excel file. The **Code** and **Name** attributes of the Supplier entity correspond to the **SupplierID** and **Supplier Name** columns in Excel.

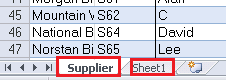
1. Open **Cleansed and Matched Suppliers.xls** in **EXCEL**.
2. Press **CTRL+A** to select entire data. It is **important** that you select the entire data in the spreadsheet.
3. Click **Master Data** on the menu bar.
4. Click **Create Entity** button on the ribbon.

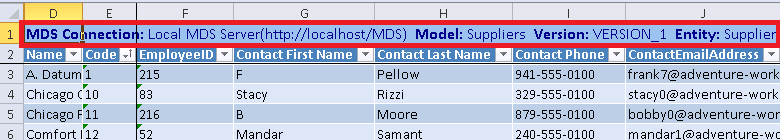


1. In the **Manage Connections** dialog box, if you do not see the connection to **local MDS server** under **Existing connections**, do the following:
   1. Select **Create a new connection**, and click **New** button.
   2. In the **Add New Connection** dialog box, type **Local MDS Server** for **Description** and **http://localhost/MDS** for **MDS server address**, and click **OK** to close the dialog box.
2. In the **Manage Connections** dialog box, select **Local MDS Server** (http://localhost/MDS), click **Test** to test the connection. Click **OK** on the message box.
3. Click **Connect** to connect to the MDS server.
4. In the **Create Entity** dialog box, select **Suppliers** for the **Model**.
5. Ensure that **VERSION\_1** is selected for **Version**.
6. Enter **Supplier** for **New** **entity name**.
7. Select **SupplierID** for **the column that contains a unique identifier** field (you can also generate a code automatically). You are essentially mapping the **SupplierID** column in **Excel** to the **Code** attribute of **Supplier** entity.
8. Select **Supplier Name** for **the** **column that contains names** field. You are essentially mapping the **Supplier Name** column in **Excel** to the **Name** attribute of the **Supplier** entity. The **Code** and **Name** attributes are mandatory attributes for an entity in MDS.



1. Click **OK** to create the entity on MDS, publish the master data to the entity, and close **Create Entity** dialog box.
2. Now, you should see a new sheet titled **Supplier**, which is the name of the entity, added to your Excel spreadsheet and at the top of the worksheet you should see that the worksheet is connected to the MDS server. Notice that the original worksheet (titled **Sheet1**) still exists.



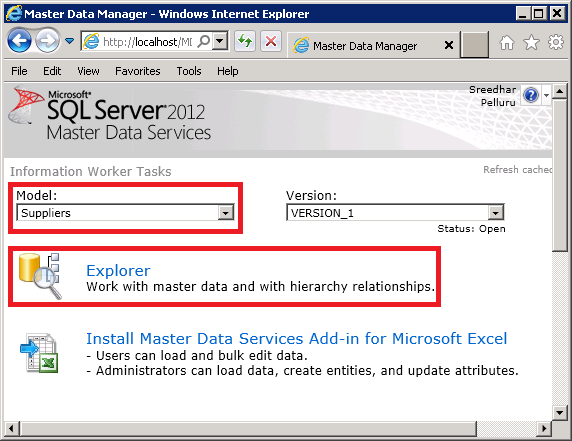


1. Keep **Excel** open.

### Task 3: Verifying the Data in Master Data Manager

In this task, you will verify that the **Supplier** entity is created on **MDS** using **Master Data Manager Web Application**.

1. If **Master Data Manager** is already open, click **SQL Server 2012 Master Data Services** at the top to navigate to the home page. Otherwise, navigate to <http://localhost/mds> to launch **Master Data Manager**.
2. Select **Suppliers** for **Model**, and click **Explorer**.

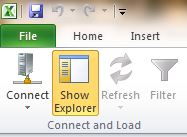


1. Review the data stored on MDS. If you do not see the data, confirm that you selected **Suppliers** for the **Model** on the home page before launching **Explorer**. You can add to or delete from the supplier list by using **Add Member** and **Delete Member** buttons on the toolbar.

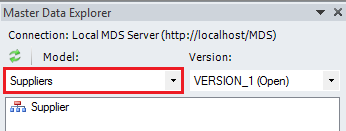
### Task 4 (Optional): Combining, Matching, and Publishing a New Set of Data

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. In the Master Data Services Add-in for Excel, you can combine data from two worksheets and then compare the data to identify and remove duplicates before publishing the data to MDS. The matching feature of the MDS Excel Add-in uses the DQS matching functionality to identify matches in the data. In this task, you will combine data from two worksheets into one and then perform matching to identify and remove duplicates before publishing to MDS. See [Data Quality Matching in the MDS Add-in for Excel](http://msdn.microsoft.com/library/hh548681.aspx) and [Combine Data](http://msdn.microsoft.com/library/hh548680.aspx) topics for more details.

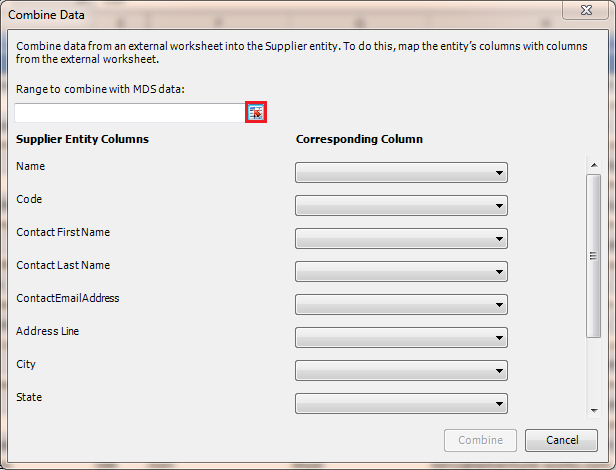
1. Launch new instance of **Excel**. Click **Start**, point to **Run**, type **Excel**, and click **OK**.
2. Switch to the **Master Data** tab by clicking **Master Data** on the menu bar.
3. Click **Connect** on the ribbon in the **Connect and Load** group to connect to the **MDS server**. You have configured this connection earlier in this lesson.



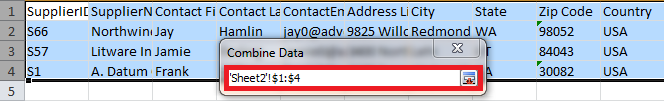
1. You should see the **Master Data Explorer** pane to the right. If you do not see the Master Data Explorer, click **Show Explorer** button on the ribbon.
2. In the **Master Data Explorer** Window, select **Suppliers** in the drop-down list for the **Model**. You should see that the model has one entity: **Supplier**.



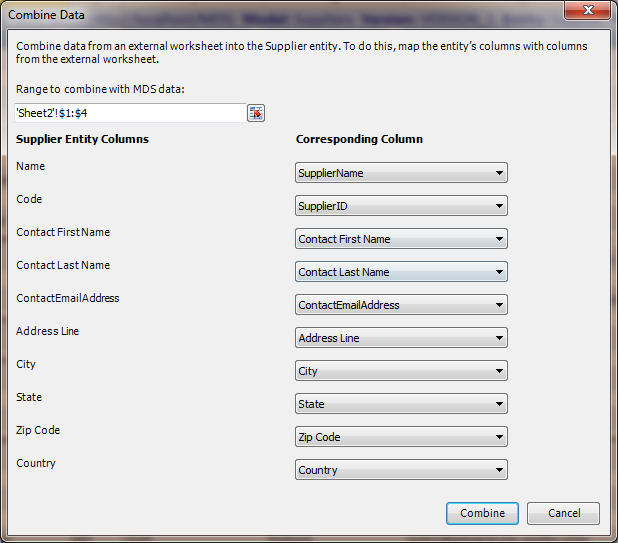
1. Double-click **Supplier** in the entity list to load the entity members into the Excel worksheet.
2. Click **Sheet2** at the bottom to switch to the **Sheet2** tab. If you do not see **Sheet2**, just create a new worksheet.
3. Open **Suppliers.xls** file (the original input file that is included in the tutorial files) and copy all (three) rows from the **CombineAndCleanse** worksheet to **Sheet2**.
4. Switch back to the **Supplier** sheet in the **Book 1 – Microsoft Excel** (not the **Cleansed and Matched Supplier List** Excel) that is connected to **MDS**.
5. Click **Combine Data** on the ribbon. You will see the **Combine Data** dialog box.
6. In the **Combine Data** dialog box, click the button next to **Range to combine with MDS data** text box as shown in the following image.



1. You should see the shrunken dialog box now. Now, click **Sheet2** to switch to the **Sheet2** tab that has the new supplier data with 4 rows (including one header row).
2. In the **Sheet2**, select **all rows including the header row** (even if they seem to be already selected). You should see the **Range to combine with MDS data** is automatically updated.



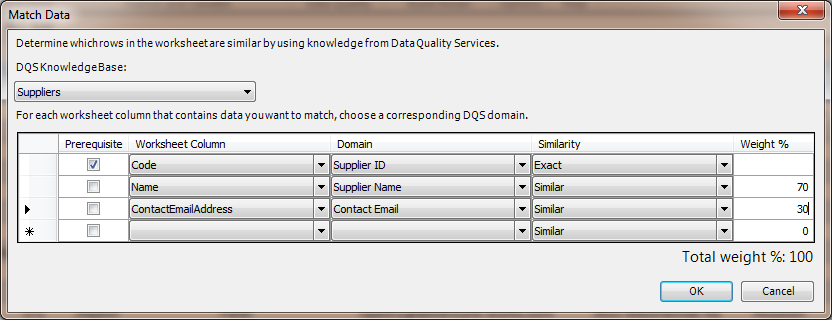
1. Switch back to the **Supplier** tab without closing the **Combine Data** dialog box.
2. Click the **button** next to the **text box**. You should see that the dialog box is expanded now. You should see that some columns of the **Supplier** MDS **entity** are mapped to **Excel** columns.



1. **Important:** Ensure that **Code** entity column is mapped to the **SupplierID** column in the worksheet and **Zip** **Code** entity column is mapped to the **Zip Code** column in the worksheet.
2. On the **Combine Data** dialog box, click **Combine**.
3. Confirm that three data rows are added to the bottom of the worksheet and they should be color coded.



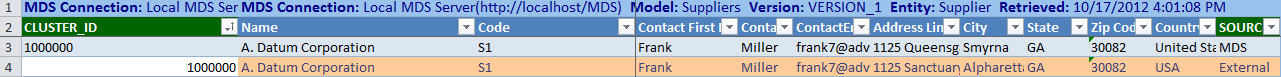
1. Click **Match Data** on the ribbon to identify duplicates. This feature uses the matching functionality of DQS.
2. In the **Match Data** dialog box, select **Suppliers** for **DQS** **Knowledge Base**.



1. Map worksheet columns to domains as shown in the following table.

|  |  |
| --- | --- |
| **Worksheet Column** | **Domain** |
| Code (you uploaded Supplier ID as the Code for the Supplier entity in MDS). | Supplier ID |
| Name (you uploaded Supplier Name as the Name for the Supplier entity to MDS) | Supplier Name |
| ContactEmailAddress | ContactEmail |

1. Select **Prerequisite** for the **Code** column mapping.
2. Enter **70%** as the **weight** for **Supplier Name** and **30%** as the **weight** for **Contact Email** as shown in the image.
3. Click **OK**.
4. The matching process should identify one duplicate for the supplier with **Code: S1**.

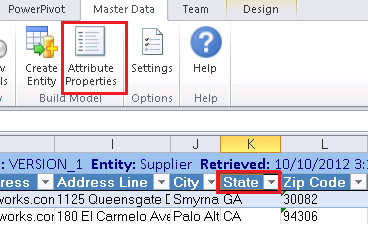


1. Select the **duplicate row (orange)**, right-click, and click **Delete** to delete the row.
2. Delete the **CLUSTER\_ID** column since you don’t need it anymore.
3. Click **Publish** to publish the updated record set (with two new records with **Codes** **S66** and **S57)** to MDS.
4. In the **Publish** **and Annotate** dialog box, add an **annotation**, and click **Publish**.
5. Switch to the **Master Data Manager** **Web application**.
6. On the home page, ensure that **Suppliers** is selected for the **Model**, and click **Explorer**. If you already have the **Explorer** open, refresh the internet browser.
7. **Sort** the list by **Code** and look for records with **S57** and **S66** as codes. You can also use the **Filter** button on the toolbar to search for a specific record in the list.
8. Now, close **Book1 – Microsoft Excel** window without saving the file.

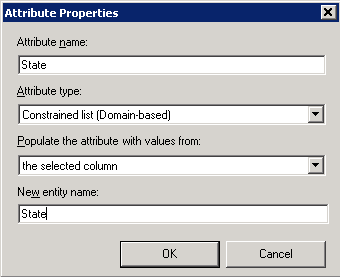
### Task 5: Creating a Domain-Based Attribute from Excel

In this task, you will convert the **State** attribute of the **Supplier** entity as a **domain-based attribute**. After you configure the State attribute to be a domain-based one and publish it to MDS, a new entity named **State** will be created on MDS server with all the values in the column and the **State** attribute of the **Supplier** entity will be populated with values from the **State** entity. Now, the **Suppliers** model should have two entities: **Supplier** and **State** where the **State** attribute of the **Supplier** entity is a domain-based attribute that depends on **State** entity.

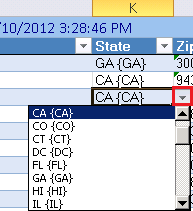
1. Switch to **Excel** window that has **Cleansed and Matched Suppliers.xlsx** open.
2. Click **Refresh** button on the ribbon to get the latest updates on MDS. You should see the two additional records if you have performed the optional **Task 4**.
3. Click column name **State** (Cell **I1**) in the **header row**.



1. Click **Attribute Properties** on the ribbon.
2. In the **Attribute Properties** dialog box, select **Constrained list (Domain-based)** for the **Attribute type**.
3. Type **State** for the **New entity name** and click **OK**.



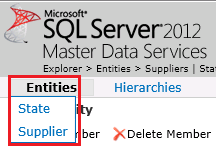
1. Now, in Excel, you will see **down arrow** when you click on any value in the **State** column. You can change the value using the drop-down list if you need.



### Task 6: Verify that the Domain-Based Attribute is Created using Master Data Manager

In this task, you will verify that the **State** entity is created in **MDS** and the **State** attribute of the **Supplier** entity is a domain-based attribute that depends on the **State** entity by using **Master Data Manager**.

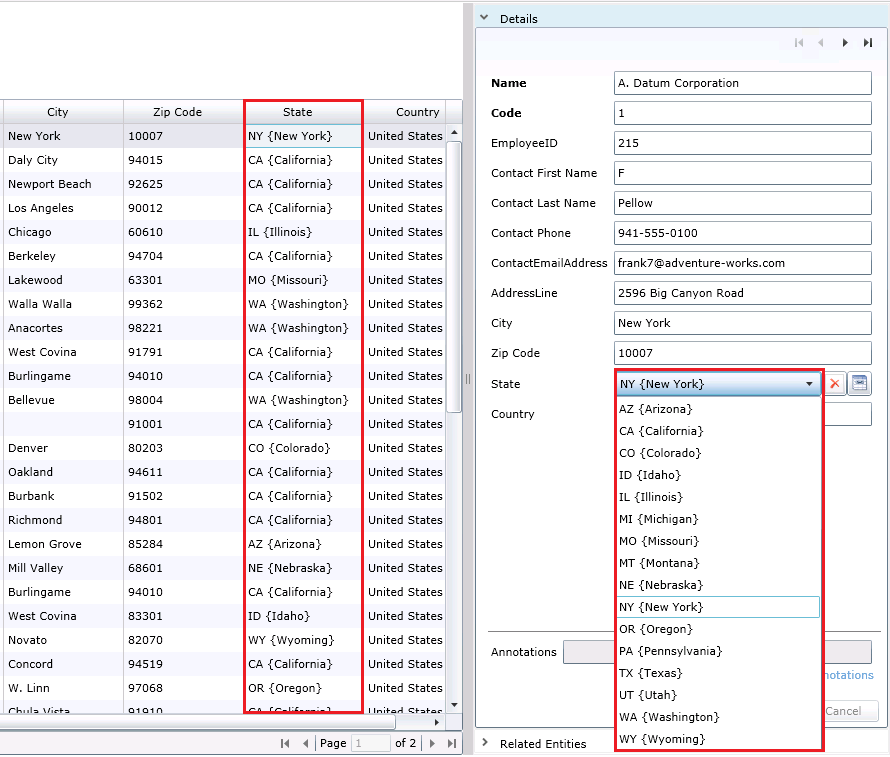
1. Switch to the **Master Data Manger** web application.
2. Click **SQL Server 2012 Master Data Services** at the top to get to the home page.
3. Ensure that **Suppliers** model is selected and click **Explorer**. You could just refresh the page if you already had **Explorer** open.
4. Hover your mouse over **Entities** in the menu bar and notice that now there are two entities: **Supplier** and **State**.



1. Click **State** if the entity is not open already.
2. Select **GA** from the list.
3. In the **Details** pane to the right, change the **Name** to **Georgia** in the **right pane**, and click **OK**.
4. Repeat the previous steps for other states.

|  |  |
| --- | --- |
| **Code** | **Name** |
| CA | California |
| CO | Colorado |
| IL | Illinois |
| DC | District of Columbia |
| FL | Florida |
| AL | Alabama |
| KY | Kentucky |
| MA | Massachusetts |
| AZ | Arizona |
| MI | Michigan |
| MN | Minnesota |
| NJ | New Jersey |
| NV | Nevada |
| NY | New York |
| OH | Ohio |
| OK | Oklahoma |
| OR | Oregon |
| PA | Pennsylvania |
| SC | South Carolina |
| KS | Kansas |
| TN | Tennessee |
| TX | Texas |
| UT | Utah |
| VA | Virginia |
| WA | Washington |
| WI | Wisconsin |
| HI | Hawaii |
| MD | Maryland |
| CT | Connecticut |

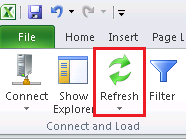
1. Select any of the above entries and click **View Transactions** from the Toolbar. You should see the transaction for the update you just made is in the list of transactions.
2. Hover the mouse over **Entities** menu and click **Supplier**.
3. Now, notice that a value for the **State** field can be changed in the **Details** pane using the drop-down list. You can also see that, in the list to the left and in the drop-down list in the **Details** pane, code is displayed first and then the name in curly braces. You can also change any other value in the **Details** pane.



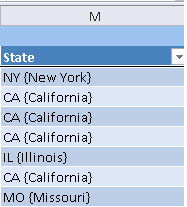
### Task 7: Viewing Updates Made using Master Data Manager in Excel

In this task, you will verify that you see the updates performed using Master Data Manager in Excel.

1. Now, switch to the excel window that has **Cleansed and Matched Suppliers** spreadsheet open.
2. Click **Refresh** button on the ribbon.



1. Notice that **names** show up (California, New York etc…) for the **State** field along with their **codes**.



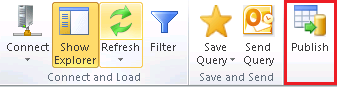
### Task 8: Adding a New Value for State Entity in Excel

In this task, you will add a new value for the **State** entity in Excel and publish the change to the MDS server.

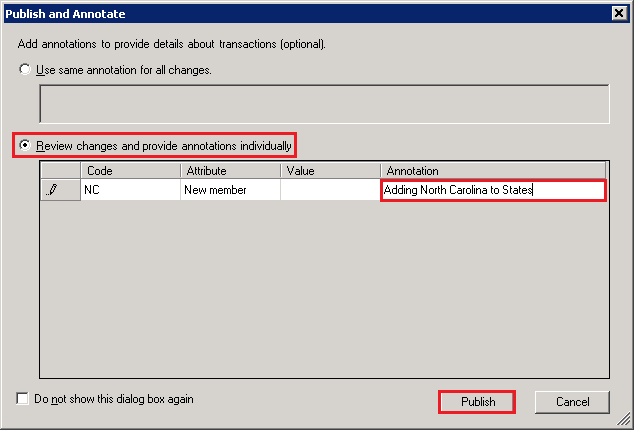
1. Create a **new work sheet** in Excel by clicking on a new tab at the bottom.



1. In **Excel**, click the **Master Data** tab on the menu, and then click **Show Explorer** on the ribbon.
2. In the **Master Data Explorer**, select **Suppliers** for **Model**. You should see two entities: **Supplier** and **State** in the entity list.
3. Double-click **State** in the list. All the members of the **State** entity from MDS should be displayed in the worksheet.
4. Now, add a new row at the end with the following values: **North Carolina** for **Name** and **NC** for **Code**. The color coding differentiates any new/updated records from the other records. 
5. Click **Publish** on the ribbon to publish the change to MDS.



1. On the **Publish and Annotate** dialog box, notice that the **Use same annotation for all changes** is selected. You can enter a single annotation for all the changes here.
2. Select **Review changes and provide annotations individually** option to provide annotation for each change (in this case, only one).

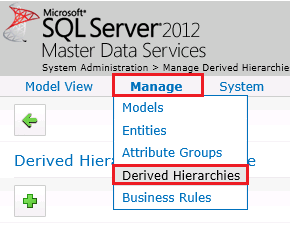


1. Click **Publish** to publish data to MDS.
2. Notice that **color coding** for the row with **North Carolina** as the **State** is same as other records now.
3. **Optiona**l: verify that the new member (NC) is added to the **State** entity by using the **Explorer** in **Master Data Manager**.
4. In Excel, right-click the **State** worksheet at the bottom, and click **Delete** to delete the worksheet. Deleting the worksheet does not delete any data from the MDS server.

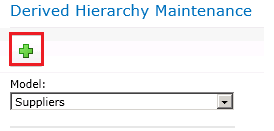
### Task 9: Creating a Derived Hierarchy using Master Data Manager

In this task, you will create a derived hierarchy by using Master Data Manager. This derived hierarchy is derived from the domain-based attribute relationships between the **Supplier** and **State** entities.

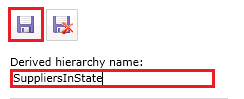
1. Switch to the main page of **Master Data Manager** by clicking **SQL Server 2012 Master Data Services** at the top of the page.
2. Click **System Administration** in the **Administrative Tasks** section.
3. Hover the mouse over **Manage** on the menu bar, and click **Derived Hierarchies**.



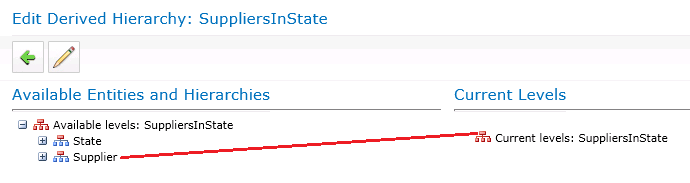
1. Click **Add Derived Hierarchy (+)** button on the toolbar.



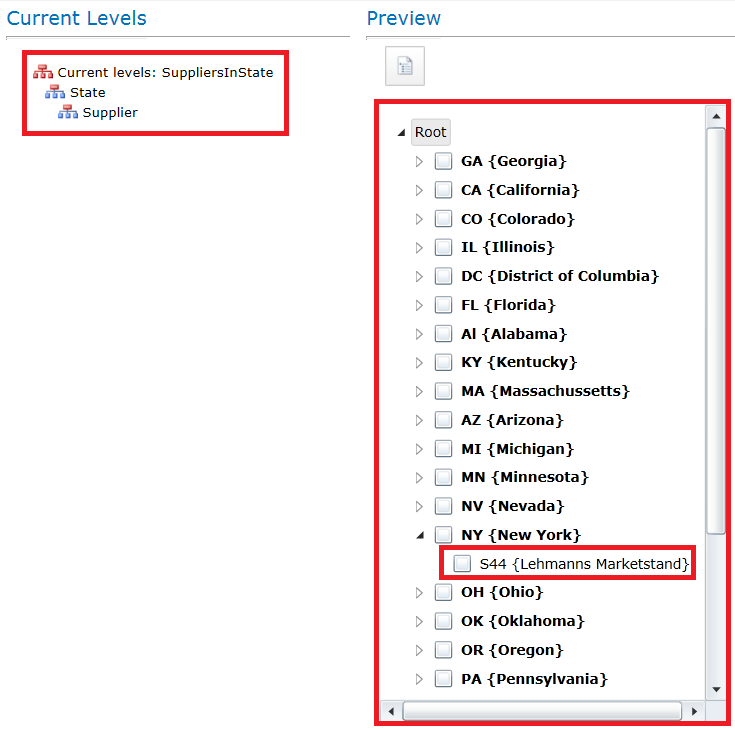
1. Type **SuppliersInState** for the **Derived hierarchy name**.
2. Click **Save** button on the toolbar to save.



1. Drag **Supplier** from **Available Levels: SuppliersInState** to **Current Levels: SuppliersInState**.



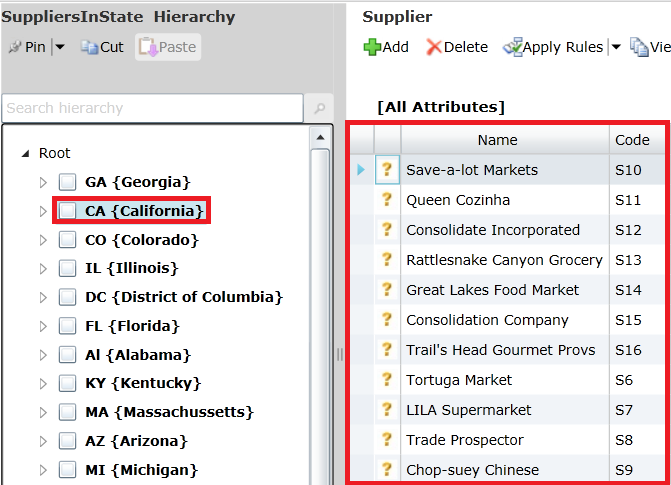
1. Drag State from **Available Levels**: **SuppliersInState** to **Current Levels**: **SuppliersInState**. The screen should have **Current Levels** as shown in the following picture.



1. In the **Preview** window, expand **NY { New York}** and you should see one supplier in that state as shown in the preceding image.
2. Switch to the main page of **Master Data Manager** by clicking **SQL Server 2012 Master Data Services** at the top of the page.
3. Click **Explorer**.
4. Hover the mouse over **Hierarchies** and click **Derived:SuppliersInState**.



1. Click on any **state** node in the **tree view** and you should see the suppliers in that state in the right pane.



## Lesson 5: Automating the Cleansing and Matching using SSIS

In Lesson 1, you built the Suppliers KB and used that KB to perform cleansing activity in Lesson 2 and the matching activity in Lesson 3 using the tool **DQS Client**. In a real world scenario, you may have to pull data from a source that is not supported by DQS or you want to automate the cleansing and matching process without having to use the **DQS Client** tool. SQL Server Integration Services (SSIS) has components that you can use to integrate data from various heterogeneous sources and a [**DQS Cleansing** **Transform**](http://msdn.microsoft.com/library/ee677619.aspx) component to invoke the cleansing functionality exposed by DQS. Currently, DQS does not expose matching functionality for SSIS to use, but you can use the [**Fuzzy Grouping Transform**](http://msdn.microsoft.comlibrary/ms141764.aspx) to identify duplicates in the data.

You can upload data to MDS by using the **Entity-based Staging feature**. When you create an entity in MDS, corresponding staging tables and stored procedures are automatically created. For example, when we created the Supplier entity, the **stg.supplier\_Leaf** table and the **stg.udp\_Supplier\_Leaf** stored procedure were automatically created. You use the staging tables and procedures to create, update, and delete entity members. In this lesson, you will be creating new entity members for the Supplier Entity. To load data into the MDS server, the SSIS package first loads the data into the staging table stg.supplier\_Leaf and then triggers the associated stored procedure stg.udp\_Supplier\_Leaf. See [Importing Data](http://msdn.microsoft.com/library/ee633726.aspx) for more details.

In this lesson, you will perform the following tasks:

1. Remove supplier data in MDS (if you have gone through Lessons 1-4). The SSIS package you create in this lesson uploads the data to MDS automatically. Earlier, you uploaded the cleansed and matched supplier data to MDS server manually using the DQS Client.
2. Create a subscription view on the Supplier entity to expose data in the entity to other applications. This creates a SQL view that you will verify using SQL Server Management Studio. You will not be consuming this view in this version of the tutorial.
3. Create and run an SSIS project using **SQL Server Data Tools**. The project will use **Data Cleansing** transform to submit a cleansing request to the DQS server. The matching functionality is not exposed by DQS yet, so you will use **Fuzzy Grouping** transform to identify duplicates.
4. Verify that the data is created in MDS by using Master Data Manger.
5. Review the results from DQS cleansing project created by the SSIS package and optionally perform interactive cleansing to further build the knowledge base.

### Task 1 (Prerequisite): Removing Supplier Data in MDS

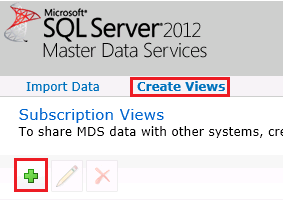
In this task, you will remove the supplier data stored in MDS. You had uploaded the data manually using **MDS Excel Add-in** in the previous lesson. The SSIS package you will be creating in this lesson will automatically load the data into MDS for you. Therefore, before testing the SSIS package, we need to remove the supplier data from MDS, remove the derived hierarchy, remove supplier and state entities, and create the supplier entity with no data.

1. Launch **Master Data Manager** by navigating to **http://localhost /MDS** or the Web site and application you specified when configuring MDS. If you kept the **Master Data Manager** open, click **SQL Server 2012 Master Data Services** at the top to switch to the **home page**.
2. Click **System Administration** in the **Administrative Tasks** section.
3. Hover the mouse over **Manage** on the menu and click **Derived Hierarchies**. We need to delete the derived hierarchy **SuppliersInState** before deleting the entities in the **Suppliers** model.
4. Select **SuppliersInState** from the **Derived Hierarchy** list and click **X (Delete)** button on the toolbar.
5. Click **OK** to confirm deletion.
6. Hover the mouse over **Manage** on the menu and click **Entities**.
7. Click **Supplier** and click **Delete (X)** button on toolbar to delete the entity. Click **OK** on message boxes.
8. Repeat the previous step to delete **State** entity.
9. Don’t close **Master Data Manager**.
10. Switch to the Excel window that has **Cleansed and Matched** **Suppliers.xls** file open. Switch to the **Sheet1** tab at the bottom.
11. Select only the **first row with headers**. Don’t select any other row. You want to just create the entities based on the Excel columns but don’t want to upload any data, therefore you select only the first row with the headers.
12. Click **Master Data** on the menu bar.
13. Click **Create Entity** from the ribbon.
14. In the **Manage Connections** dialog box, if you do not see the connection to **local MDS server** under **Existing connections**, do the following:
    1. Select **Create a new connection**, and click **New** button.
    2. In the **Add New Connection** dialog box, type **Local MDS Server** for **Description** and **http://localhost/MDS** for **MDS server address**, and click **OK** to close the dialog box.
15. In the **Manage Connections** dialog box, select **Local MDS Server** (http://localhost/MDS), click **Test** to test the connection. Click **OK** on the message box.
16. Click **Connect** to connect to the MDS server.
17. In the **Create Entity** dialog box, do the following:
    1. Confirm that **Range** is set to **$1:$1**.
    2. Select **Suppliers** for **Model**.
    3. Select **VERSION\_1** for **Version**.
    4. Type **Supplier** for **New entity name**.
    5. Select **SupplierID** for **Code**.
    6. Select **Supplier Name** for **Name**.
    7. Click **OK** to create the entity and close the dialog box.
18. Close **EXCEL** and **do not save** the file.
19. In **Master Data Manager**, refresh the internet browser and confirm that **Supplier** entity is displayed in the list.
20. Switch to the **home** **page** by clicking **SQL Server 2012 Master Data Services** at the top.
21. Confirm that **Suppliers** is selected for **Model** and **VERSION\_1** is selected for **Version**.
22. Click **Explorer**. Notice that the **Supplier** entity with all the attributes is created with **no values**.

### Task 2 (Optional): Creating a MDS Subscription View using Master Data Manager

In this task, you will create a subscription view to expose the **Supplier** entity in the **Suppliers** model to other applications. You will not be consuming this view in the current version of the tutorial.

1. Switch to the main page of **Master Data Manager** (<http://localhost/MDS>) by clicking **SQL Server 2012 Master Data Services** at the top.
2. Click **Integration Management.**
3. Click **Create Views** on the menu bar.



1. Click **+ (Plus)** icon on the toolbar to create a new subscription view.
2. In the **Create Subscription View** pane, type **Suppliers** for **Subscription view name**.
3. Select **Suppliers** for **Model**.
4. Select **VERSION\_1** for **Version**.
5. Select **Supplier** for **Entity**.
6. Select **Leaf members** for **Format**.

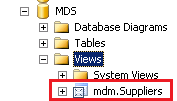


1. Click **Save** on the toolbar to save the subscription view. This actually creates a view in SQL Server named **Suppliers**. You can verify this using SQL Server Management Studio (SSMS).

### Task 3 (Optional): Reviewing the Subscription Views

In this task, you will confirm that the SQL views are created by using SQL Server Management Studio.

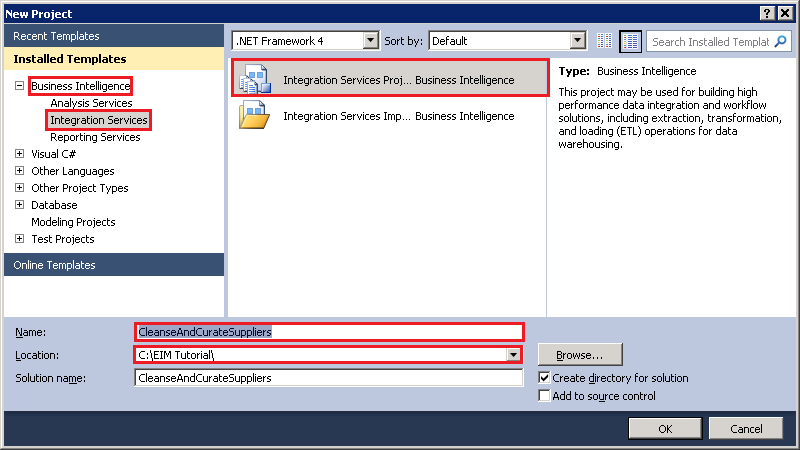
1. Launch **SQL Server Management Studio**. Click the **Start** button, click **All Programs**, click **Microsoft SQL Server 2012**, and then click **SQL Server Management Studio**.
2. In the **Connect to Server** window, set **Server Type** to **Database Engine**, type the **server name (**or select **(local)**, and select appropriate **authentication**, and click **Connect** to connect to the server.
3. In the **Object Explorer** pane, expand **Databases**, expand **MDS**, and then expand **Views**.
4. Confirm that you see the **mdm.Suppliers** view in the list.



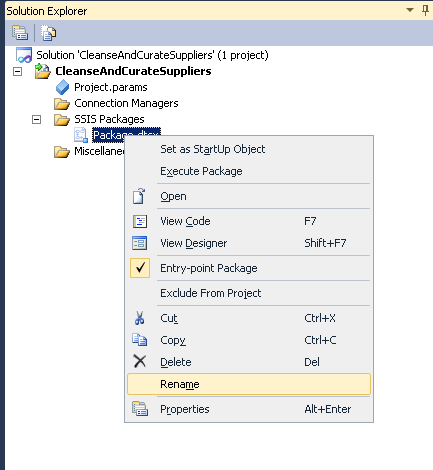
### Task 4: Creating an SSIS Project using SQL Server Data Tools

In this task, you will create an SSIS project using **SQL Server Data Tools** to automate cleansing and matching supplier data.

1. Launch **SQL Server Data Tools**. Click **Start**, point to **All Programs**, expand **Microsoft SQL Server 2012**, and click **SQL Server Data Tools**.
2. Click **File** on menu, point to **New**, and click **Project**.
3. Expand **Business Intelligence** in the **Installed Templates** pane, and select **Integration Services**.



1. Select **Integration Services Project** in the **list of project types**.
2. Type **CleanseAndCurateSuppliers** for **Name** and click **OK**.
3. In the **Solution Explorer** window, right-click **Package.dtsx** and select **Rename**. If you don’t see the **Solution Explorer** window, click **View** on the menu bar and click **Solution Explorer**.

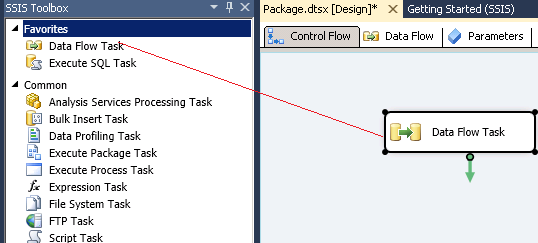


1. Type **CleanseAndCurate.dtsx** and press **ENTER**. Make sure that the **extension** remains **.dtsx**.

### Task 5: Adding Data Flow Task

In this task, you will add a Data Flow Task to the control flow of SSIS package.

1. Drag and drop **Data Flow Task** from **SSIS Toolbox** to the **Control Flow** tab in the SSIS Designer. If you do not see the **SSIS Toolbox**, click anywhere in the **Control Flow** tab, click **SSIS** on the menu bar, and click **SSIS Toolbox**.

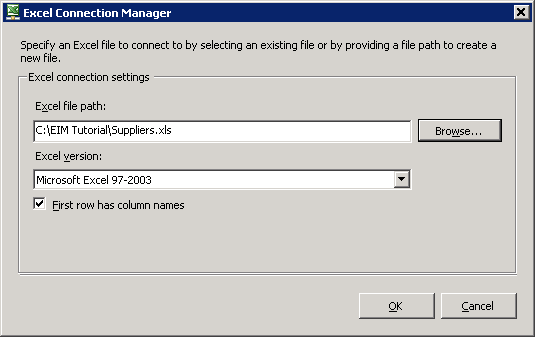


1. Right-click the **Data Flow Task** in the **Control Flow** tab and click **Rename**.
2. Type **Receive, Cleanse, Match, and Curate Supplier Data** and press **ENTER**.
3. Double-click on the **Data Flow Task** to switch to the **Data Flow** tab.

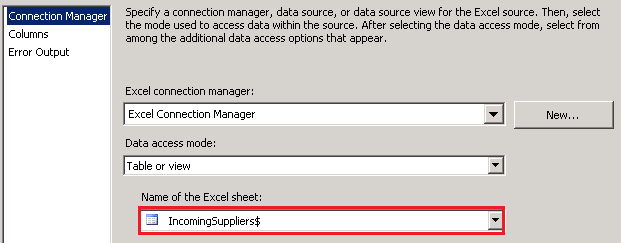
### Task 6: Adding Excel Source to the Data Flow

In this task, you will add an Excel Source to the data flow to read supplier data from the source Excel file. The Excel Source extracts data from worksheets or ranges in Microsoft Excel workbooks. See [Excel Source](http://msdn.microsoft.com/library/ms141683.aspx) topic for more details.

1. Drag-drop **Excel Source** from **Other Sources** in **SSIS Toolbox** to the **Data Flow** tab.
2. Right-click on **Excel Source** in the **Data Flow** tab, and click **Rename**.
3. Type **Read Supplier Data from Excel File** and press **ENTER**.
4. Double-click **Read Supplier Data from Excel File** to launch the **Excel Source Editor** dialog box.
5. In the **Excel Source Editor** dialog box, click **New** to create a new Excel connection.
6. In the **Excel Connection Manager** dialog box, click **Browse**, and then select the **Suppliers.xls** file in the **EIM Tutorial** folder. Confirm that **Microsoft Excel 97-2003** is selected in the **Excel Version** box and then click **OK**.



1. In the **Excel Source Editor** dialog box, select **IncomingSuppliers$** in the **Name of the Excel sheet** list box.

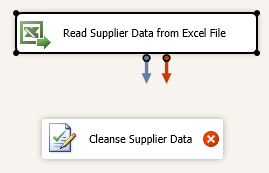


1. Click **Preview** to preview the data in Excel file.
2. Click **OK** to close the dialog box.
3. Drag-drop **DQS Cleansing** transform in **Other Transforms** on the **SSIS Toolbox** to the **Data Flow** tab under **Read Supplier Data from Excel File**. The DQS Cleansing transformation uses Data Quality Services (DQS) to correct data by applying approved rules in the knowledge base. This transform, at runtime, creates a DQS cleansing project on the DQS server. See [DQS Cleansing Transformation](http://msdn.microsoft.com/library/ee677619.aspx) topic for more details.

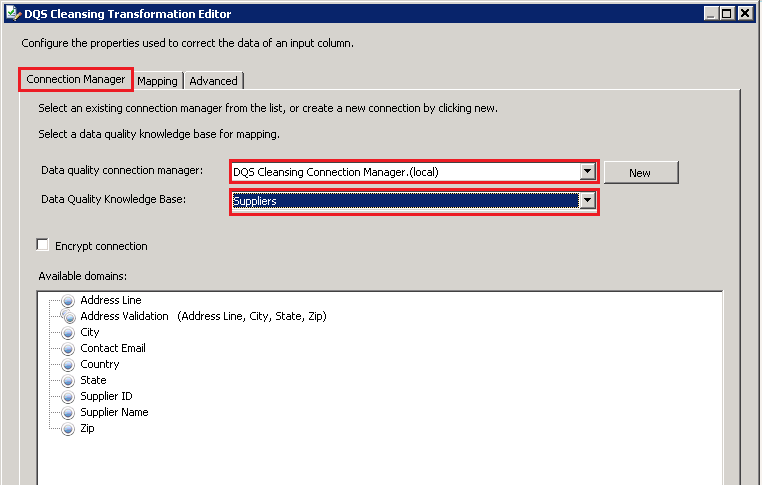
### Task 7: Adding DQS Cleansing Transform to the Data Flow

In this task, you will add DQS Cleansing Transform to the data flow to cleanse the input supplier data by using DQS. See [**DQS Cleansing** **Transform**](http://msdn.microsoft.com/library/ee677619.aspx)for more details about the transform.

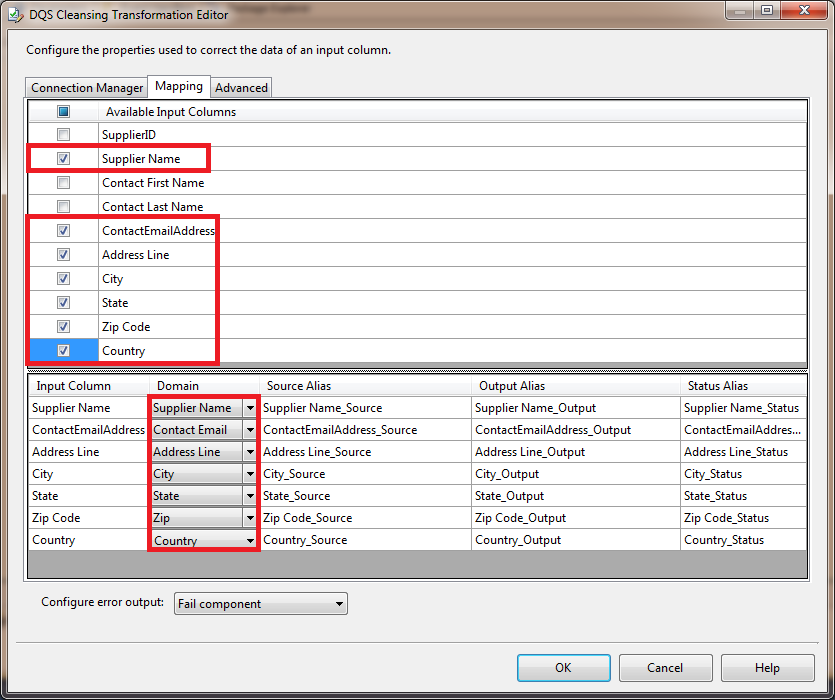
1. Right-click **DQS Cleansing** in the **Data Flow** tab, and click **Rename**.Type **Cleanse Supplier Data**, and press **ENTER**.
2. Select **Read Supplier Data from Excel File**; drag the blue connector to **Cleanse Supplier Data**. The components are now connected.



1. Double-click **Cleanse Supplier Data**.
2. In the **DQS Cleansing Transformation Editor**, click **New** next to the **Data Quality Connection Manager drop-down list**.
3. In the **DQS Cleansing Connection Manager** dialog box, type **(local)** or **period** (.) to connect to the local server. This lesson assumes that you have DQS installed on a local server.
4. Click **Test Connection** to test the connection to DQS server.
5. Click **OK** to close the dialog box.
6. Select **Suppliers** for the **Data Quality Knowledge Base**.



1. Switch to the **Mapping** tab at the top.
2. From **Available Input Columns**, select **Supplier Name**, **ContactEmailAddress**, **Address Line**, **City**, **State**, **Country, and Zip Code** by clicking the check boxes.



1. In the bottom pane, map these columns using drop-down lists in the **Domain** column:

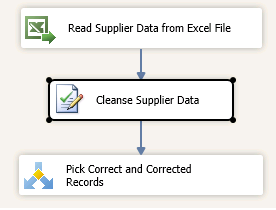
|  |  |
| --- | --- |
| **Column** | **Domain** |
| Supplier Name | Supplier Name |
| ContactEmailAddress | Contact Email |
| Address Line | Address Line |
| City | City |
| State | State |
| Country | Country |
| Zip Code | Zip |

1. Click **OK** to close the **DQS Cleansing Transformation Editor** dialog box.

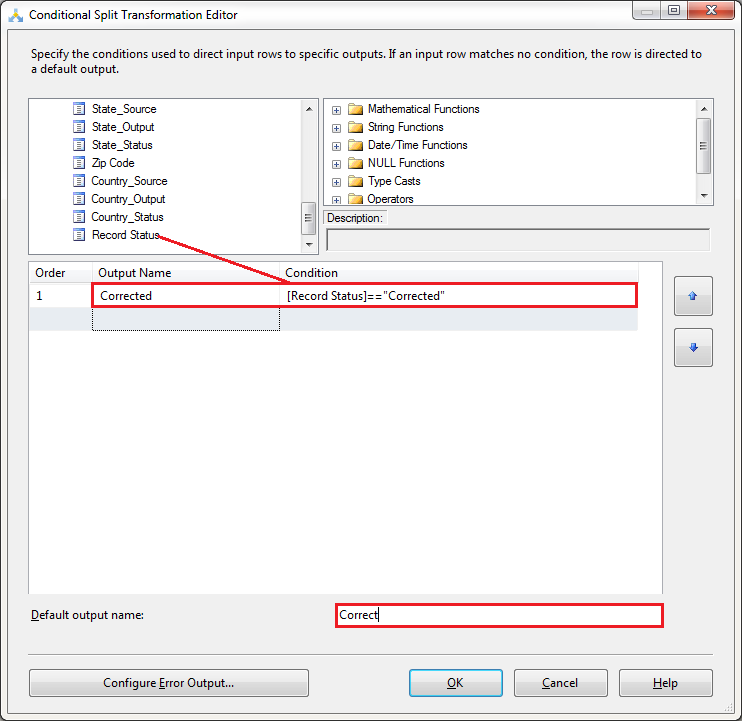
### Task 8: Adding Conditional Split Transform to Split Cleansing Output

In this transform, you will add a Conditional Split Transform to the data flow. The Conditional Split transformation can route rows to different outputs depending on the content of the data. For the purpose of this tutorial, you will use the **Record Status** output column from the DQS Cleansing transform. You will upload only correct or corrected records to MDS server in this tutorial. Therefore you will check if the **Record Status** is **Correct** or **Corrected**, and combine the records before uploading the records to MDS.

1. Drag-drop **Conditional Split Transform** from **Common** section in the **SSIS Toolbox** to the **Data Flow** tab below **Cleanse Supplier Data**.
2. Right-click **Conditional Split**, and click **Rename**. Type **Pick Correct and Corrected Records** and press **ENTER**.
3. Connect **Cleanse Supplier Data** and **Pick Correct and Corrected Records** using the blue connector.



1. Double-click **Pick Correct and Corrected Records** in the **Data Flow** tab.
2. Change the **Default Output Name** at the bottom of the screen to **Correct**.
3. Expand **Columns** in the **top-left pane**.

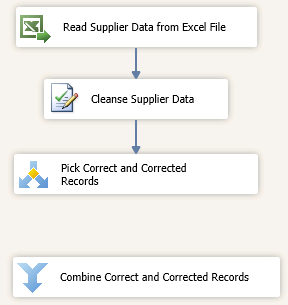


1. Drag-drop **Record Status** to the **Condition** column.
2. Type **==“Corrected”** next to **[Record Status]** for the **Condition** column.
3. Click **Case 1** in the **Output Name Column**, and change the name to **Corrected**.
4. Click **OK** to close the **Conditional Split Transformation** Editor dialog box.

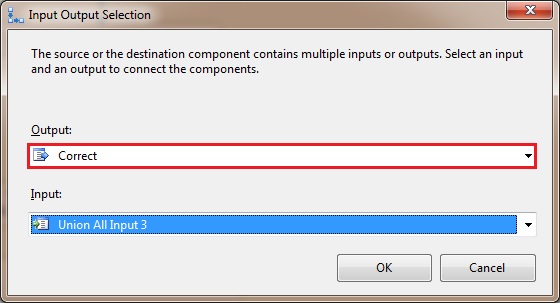
### Task 9: Adding Union All Transform to Combine Correct and Corrected Records

In this task, you will add the Union All Transform to the data flow. The Union All transformation combines multiple inputs into one output. In our scenario, it will combine both Correct and Corrected records into one stream.

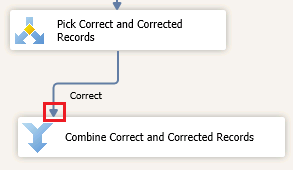
1. Drag-drop **Union All** Transform from **Common** section of the **SSIS Toolbox** to the **Data Flow** tab and place it below **Pick Correct and Corrected Records**.
2. Right-click **Union All** Transform in the **Data Flow** tab, and click **Rename**. Type **Combine Correct and Corrected Records**, and press **ENTER**.



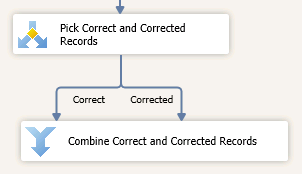
1. Connect **Pick Correct and Corrected Records** to **Combine Correct and Corrected Records** in the **Data Flow** tabusing the blue connector. You should see the **Input Output Selection** dialog box.
2. In the **Input Output** dialog box, select **Correct** for **Output** and click **OK**.



1. Move the connector titled **Correct** to the left by dragging and dropping the dot at the end of the connector to left.



1. If you select **Pick Correct and Corrected Records** transform, you should see another blue connector. Drag that blue connector to **Combine Correct and Corrected Records**.

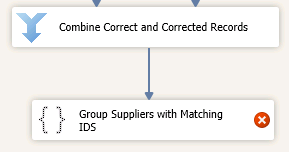


1. This **connector** should be titled **Corrected**. Since we have only two conditions **Correct** and **Corrected**, and one condition was already used, the **Input Output Selection** dialog box is not displayed this time. If the connectors overlap, move one to left and the other one to right by dragging the connector to left or right.

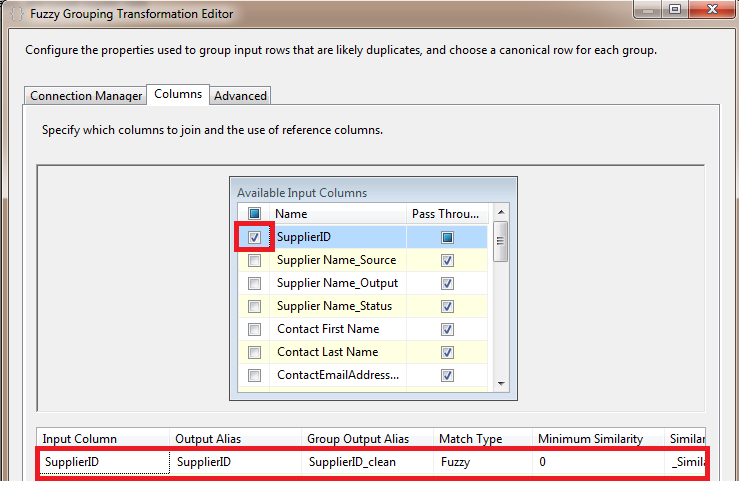
### Task 10: Adding Fuzzy Group Transform to Identify Duplicates

In this task, you will add a Fuzzy Group Transform to the data flow. The Fuzzy Group transformation can help identify duplicates in the source data. See [Fuzzy Grouping Transformation](http://msdn.microsoft.com/library/ms141764.aspx) for more details.

1. Drag-drop **Fuzzy Group** transform in **Other Transforms** on the **SSIS Toolbox** to the **Data Flow** tab below **Combine Correct and Corrected Records**.
2. Right-click **Fuzzy Group** Transform in the **Data Flow** tab, and click **Rename**. Type **Group Suppliers with matching IDs** and press **ENTER**.
3. Connect **Combine Correct and Corrected Records** to **Group Suppliers with matching IDs** using the blue connector.



1. Double-click **Group Suppliers with matching IDs**.
2. In the **Fuzzy Group Transformation Editor**, click **New** next to **OLE DB Connection Manager drop-down list** to launch **Configure OLE DB Connection Manager** dialog box.
3. In the dialog box, click **New** to launch **Connection Manager** dialog box.
4. Type **(local)** or **period** (.) for the Server name.
5. Select **MDS** for **Select or enter a database name** field. We will be using MDS database as the temporary storage for the **Fuzzy Group Transform**. The **Fuzzy Grouping** transformation requires a connection to an instance of SQL Server to create the temporary SQL Server tables that the transformation algorithm requires to do its work. You can create a new database or use another existing database for this purpose.
6. Click **Test Connection** to test the connection and click **OK** on the message box.
7. In the **Connection Manager** dialog box, click **OK**.
8. Select **(local).MDS (**or **localhost.MDS)** from the **list of Data Connections** and click **OK**.
9. In the **Fuzzy Grouping Transformation** Editor, confirm that **(local).MDS** or **localhost.MDS** is selected for the **OLE DB Connection Manager**.
10. Switch to the **Columns** tab.
11. Select (check box) **SupplierID\_Output** from the list of **Available Input Columns**. To configure the transformation, you must select the input columns to use when identifying duplicates. To keep it simple, you will only use the SupplierID in this step.

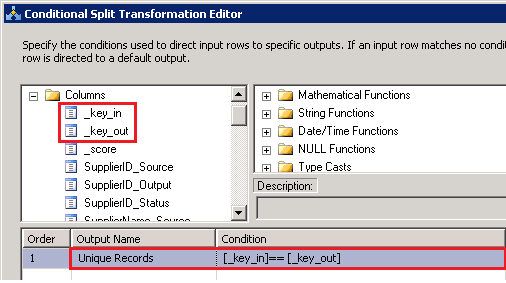


1. Click **OK** to close the **Fuzzy Group Transformation Editor**.

### Task 11: Adding Conditional Split Transform to Filter Duplicates

In this task, you will add the Conditional Split Transform to the data flow. This transform will help you filter duplicates from the incoming record set. The Fuzzy Group transform groups the records that it finds to be matches and picks one of the record as a pivot record. All the records in a group have the same \_key\_out value. The pivot record in the group has \_key\_in same as the \_key\_out value. The other records in the group have different values for \_key\_in and \_key\_out. Therefore, when you filter using the condition \_key\_in==\_key\_out, you only get the pivot row in the group.

1. Drag-drop **Conditional Split** Transform from **Common** section in the **SSIS Toolbox** to the **Data Flow** tab.
2. Right-click **Conditional Split** Transform in the **Data Flow** tab, and click **Rename**. Type **Filter Duplicates** and press **ENTER**.
3. Connect **Group Suppliers with Matching IDs** to **Filter Duplicates**.
4. Double-click **Filter Duplicates** to launch the **Conditional Split Transform Editor** dialog box.
5. Expand **Columns** in the top-left pane.
6. Drag-drop **\_key\_in** to the **Condition** column.
7. Type **==** (equals to) next to **\_key\_in** and drag-drop **\_key\_out**.
8. Click **Case 1** in the **Output Name** column, type **Unique Records**, and press **ENTER**.

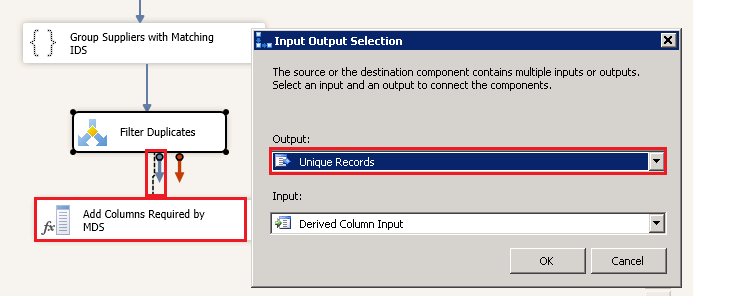


1. Click **OK** to close the **Conditional Split Transformation Editor** dialog box.

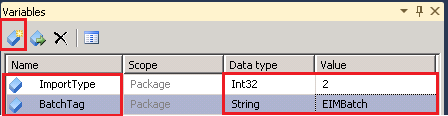
### Task 12: Adding Derived Column Transform to Add Columns Required by MDS

In this task, you will add the Derive Column Transform to the data flow. You will add two derived columns, **ImportType** and **BatchTag**, to the records passed to this transform. You need to add these columns before uploading the data to staging tables in MDS. These two are required columns for the staging tables in MDS. See [Leaf Member Staging Tables](http://msdn.microsoft.com/library/ee633854.aspx) for more details.

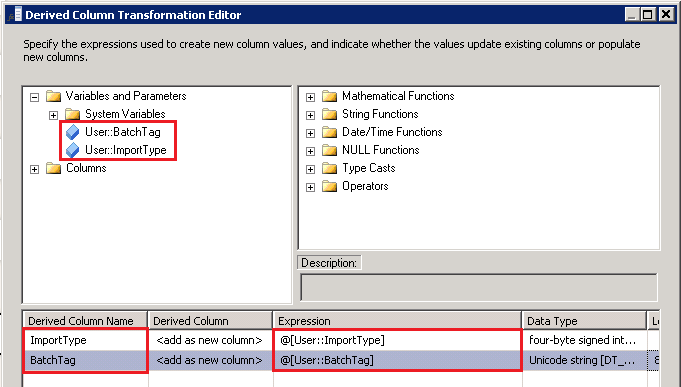
1. Drag-drop **Derived Column transform** from **Common** section in the **SSIS Toolbox** to the **Data Flow** tab.
2. Right-click **Derived Column** Transform in the **Data Flow** tab, and click **Rename**. Type **Add Columns Required by MDS** and press **ENTER**.
3. Connect **Filter Duplicates** to **Add Columns Required by MDS** using the blue connector. This will launch the **Input Output Selection** dialog box.
4. In the **Input Output Selection** dialog box, select **Unique Records**, and click **OK**.



1. Click **SSIS** on the menu bar and click **Variables**.
2. In the **Variables** window, click **Add Variable** button on the toolbar.



1. Type **ImportType** for the **Name** and **2** for the **value**. You specify the value as 2 because you want to add new members to an entity in MDS. For details about this parameter, see [Leaf Member Staging Table](http://msdn.microsoft.com/library/ee633854.aspx).
2. Click **Add Variable** toolbar button again.
3. Type **BatchTag** for the **Name**, select **String** for the **Data type**, and **EIMBatch** for the Value. **BatchTag** is just a unique name for the batch you will be submitting to MDS.
4. In the **Data Flow** tab, double-click **Add Columns Required by MDS**.
5. In the **Derived Column Transformation** **Editor** dialog box, in the **list box in the bottom pane**, type **ImportType** for the **Derived Column Name**.
6. Expand **Variables and Parameters** in the top-left pane, drag-drop **User::ImportType** to the **Expression** column.

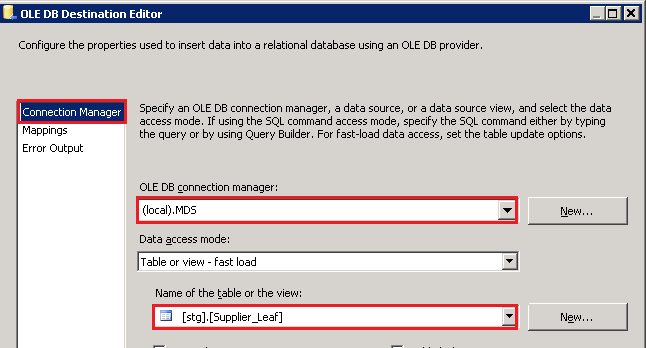


1. Type **BatchTag** in the next row for the **Derived Column Name**.
2. Drag-drop **User::BatchTag** from **Variables and Parameters** to the **Expression** column.
3. Click **OK** to close the **Derived Column Transformation** dialog box.

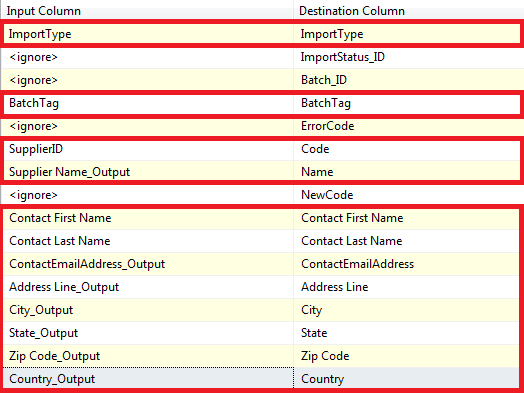
### Task 13: Adding OLE DB Destination to Write Data to MDS Staging Table

Now that you have added **ImportType** and **BatchTag** values to all records, you are ready to send them over to MDS for staging. In this task, you will use the OLE DB Destination to write the data into **stg.supplier\_Leaf** staging table.

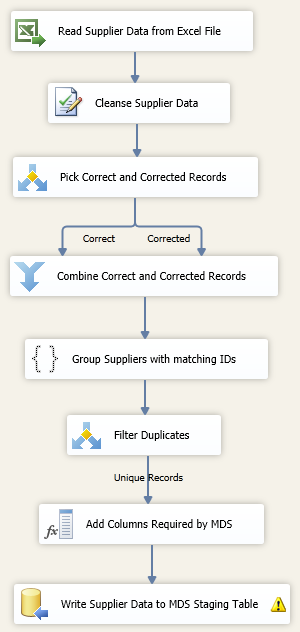
1. Drag **OLE DB Destination** from **Other Destinations** section in the **SSIS Toolbox** to the **Data Flow** tab and drop it below **Add Columns Required by MDS**.
2. Right-click **OLE DB Destination** in the **Data Flow** tab, and click **Rename**. Type **Write Supplier Data to MDS Staging Table** and press **ENTER**.
3. Connect the **Add Columns Required by MDS** to **Write Supplier Data to MDS Staging Table** using the blue connector.
4. Double-click **Write Supplier Data to MDS Staging Table** in the **Data Flow** tab.
5. In the **OLE DB Destination Editor** dialog box, make sure that **(local).MDS** (or **localhost.MDS**) is selected for the **OLE DB Connection Manager** field.
6. Select **stg.Supplier\_Leaf** table from the list of **Name of the table or the view**.



1. Switch to the **Mappings** page by clicking **Mapping** in the menu on left.
2. Map **input** and **destination** columns as shown in the following table.



1. Confirm that you are using **\_Output** columns for Input Columns, not the **\_Status** or **\_Source** columns. **\_Output** columns contain the output values from DQS Cleansing.
2. Click **OK** to close the **OLE DB Destination Editor** dialog box.
3. The data flow should like the following image.

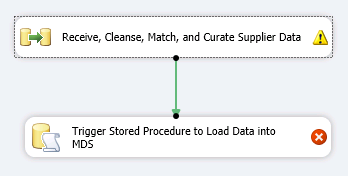


### Task 14: Adding Execute SQL Task to Control Flow to Run the Stored Procedure for MDS

After loading data into the staging tables of MDS, you need to run a stored procedure associated with that table to load the data from staging into the appropriate tables in the MDS database. This stored procedure has two required parameters that you need to pass: LogFlag and VersionName. LogFlag specifies whether transactions are logged during the staging process and VersionName represents the version of the model. See [Staged Stored Procedure](http://msdn.microsoft.com/library/hh231028.aspx) topic for more details.

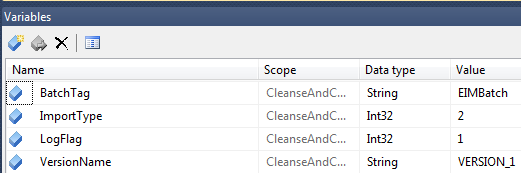
In this task, you will add the Execute SQL Task to the control flow to invoke the stored procedure to load the staged data into appropriate MDS tables.

1. Now, switch to the **Control Flow** tab.
2. Drag-drop **Execute SQL Task** from the **SSIS Toolbox** to the **Control Flow** tab.
3. Right-click **Execute SQL Task** in the **Control Flow**tab, and click **Rename**. Type **Trigger Stored Procedure to Load Data into MDS** and press ENTER.
4. Connect **Receive, Cleanse, Match, and Curate Supplier Data** to **Trigger Stored Procedure to Load Data** using the green connector.

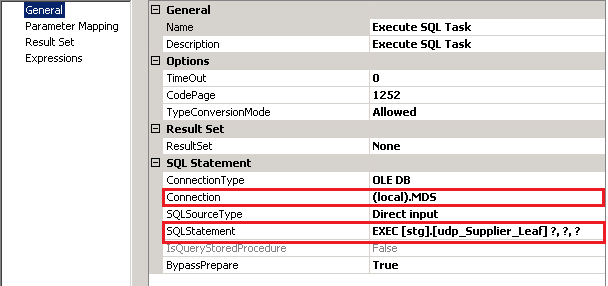


1. Using the **Variables** window, add two new variables with the following settings. If you do not see the **Variables** window, click **SSIS** on the menu bar and click **Variables**.

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Value** |
| LogFlag | Int32 | 1 |
| VersionName | String | VERSION\_1 |

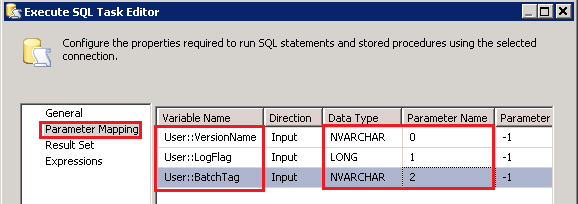


1. Double-click **Trigger Stored Procedure to Load Data into MDS**.
2. In the Execute **SQL Task Editor** dialog box, select **(local).MDS** (or **localhost.MDS**) for **Connection**.
3. Type **EXEC [stg].[udp\_Supplier\_Leaf] ?, ?, ?** for **SQL Statement**. You can verify the name using SQL Server Management Studio.



1. Click **Parameter Mapping** from the menu on left.
2. In the **Parameter Mapping** page, click **Add** to add a new mapping. Maximize the window and resize columns so that you can see values in drop-down lists properly.
3. Select **User::VersionName** from the drop-down list for the **Variable Name**.
4. Select **NVARCHAR** for Data Type.
5. Type **0** (zero) for **Parameter Name**.
6. Repeat the previous four steps to add two more variables.

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type (important)** | **Parameter Name** |
| User::LogFlag | LONG | 1 |
| User::BatchTag | NVARCHAR | 2 |

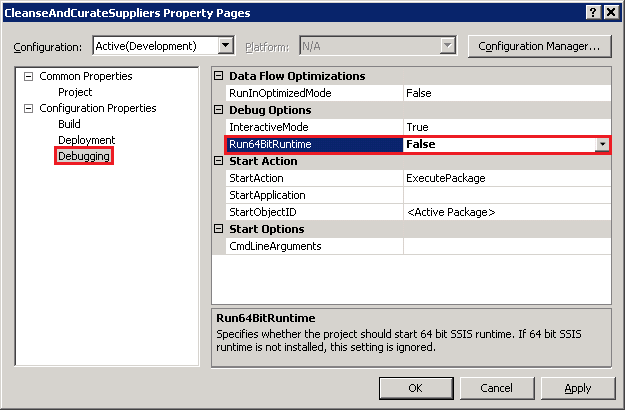


1. Click **OK** to close the **Execute SQL Editor** dialog box.

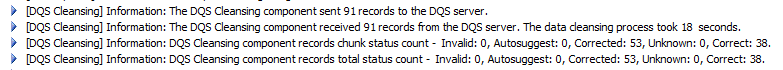
### Task 15: Building and Running the SSIS Project

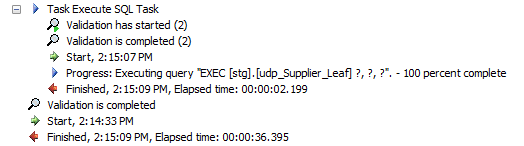
In this task, you will build and run the SSIS project. If you have the 64-bit version of Excel 2010 installed on your computer, you need to set the value of **Run64BitRuntime** to **False** for the Excel source to work. This is a known issue.

1. In the **Solution Explorer** window, Click **Project** on the menu, and click **CleanseAndCurateSuppliers** **Properties**.
2. In the **Properties** dialog box, expand **Configuration Properties** on left, and click **Debugging**.
3. Set **Run64BitRuntime** to **False**.



1. Click **OK** to close the **Properties** dialog box.
2. Click **Build** on menu bar and click **Build CleanseAndCurateSuppliers**. Make sure that there are no build errors.
3. Click **Debug** on the menu bar and click **Start Debugging**.
4. Review messages in the **Progress** window and verify that package executed and ended successfully.



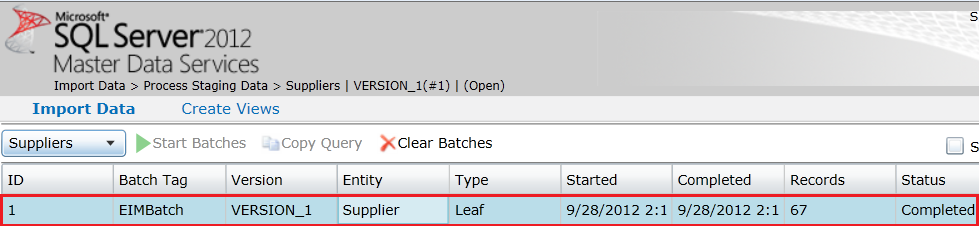


1. Click **Debug** on menu bar and click **Stop Debugging** to stop the debugging session. If the package fails, you may want to enable data viewers and see how the data flows between components.

### Task 16: Verifying with Master Data Manager

In this task, you will check the status of the batch job submitted by the SSIS package and verify that the data was uploaded to MDS server using Master Data Manager.

1. Launch **Master Data Manager** (<http://localhost/MDS>). If it is already open, click **Microsoft SQL Server Master Data Services** at the top to switch to the **home page**.
2. Click **Integration Management**.
3. Notice that there is a batch with named **EIMBatch** that you submitted in the list. Click **Import Data** on the menu bar if you do not see the following screen.

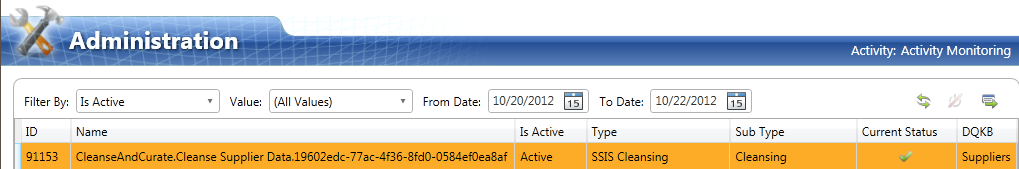


1. Switch back to the home page by click **SQL Server 2012** **Master Data Services** at the top.
2. Make sure that **Suppliers** is selected for **Model** and **VERSION\_1** is selected for **Version**, and click **Explorer**.
3. You can see the data SSIS package imported into MDS. The data should be cleansed and have no duplicates **Code** values (Note: **SupplierID** column in Excel corresponds to **Code** attribute of Supplier entity in MDS).

### Task 17: Reviewing DQS Cleansing Project Created by the SSIS package

In this project, you will open the DQS project created by the SSIS package in DQS Client, review the results from the cleansing process, and optionally perform interactive cleansing and export the results.

1. Launch **Data Quality Client**.
2. Click **Activity Monitoring** in the **Administration** pane.
3. Sort the list based on **Activity Start Time** to see the latest record.
4. Notice that you see a name of the project in the following format: **CleanseAndCurate.Cleanse Supplier Data.GUID**.



1. Notice that the value in the **Is Active** field is **Active**.
2. Click **Profiler** tab in the bottom pane to see profiler statistics for the Cleansing activity that the SSIS package performed.
3. Click **Close** to close the **Administration** screen.
4. In the main page of **DQS Client**, click **Open Data Quality Project** in the **Data Quality Projects** pane.
5. In the list of projects, select the project created by SSIS DQS Cleansing component. The name of the project should be in format: **CleanseAndCurate.Cleanse Supplier Data.GUID (in red color)**. You may need to sort the list based on **Date Created** column and look for the latest record.
6. Click **Next**.
7. The **Manage and View Results** page should be familiar to you from the interactive cleansing you did earlier in this tutorial.
8. Review the cleansing results. You can also perform interactive cleansing and export results to an Excel file or to a database in the next page.
9. Click **Next**. In this **Export** page, you can export results to an excel file, CSV file, or to a SQL database.
10. Click **Finish** to finish the activity.
11. In the main page of **DQS Client**, click **Activity Monitoring** in the **Administration** pane.
12. Notice that the value of **IsActive** field for the project is **Ended** now.

## Conclusion

In this tutorial, you have learned how to use SQL Server Integration Services (SSIS), Master Data Services (MDS), and Data Quality Services (DQS) together to implement a sample Enterprise Information Management (EIM) solution. First, you used the Data Quality Client tool to create a DQS knowledge base with the knowledge about suppliers, cleansed the input supplier data in an excel file against the knowledge base, and then matched the supplier data using a matching policy in the knowledge base to identify and remove duplicates in the data. Next, by using the MDS Add-in for Excel, you stored the cleansed and matched supplier list in MDS. Finally, you automated the whole process of receiving input data, cleansing and matching the data, and storing the master data in MDS by creating an SSIS solution.

**For more information:**

[Enterprise Information Management with SQL Server 2012](http://download.microsoft.com/download/B/F/8/BF81264C-413F-42FC-8FBF-171D97B3C8A7/SQL_Server_2012_Enterprise_Information_Management_Whitepaper.pdf) (Whitepaper)

[Enterprise Information Management (EIM): Bringing together SSIS, DQS, and MDS](http://go.microsoft.com/fwlink/?LinkId=258672) (Video)

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

* Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
* Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

[Send feedback](mailto:spelluru@microsoft.com?subject=Tutorial%20Feedback:%20Enterprise%20Information%20Management%20using%20SSIS,%20MDS,%20and%20DQS%20Together)