

# **Semarchy Convergence for MDM**

## **Getting Started Guide**

**2.2, Revision 1**

# Getting Started Guide

## Overview

Welcome to **Semarchy Convergence for MDM!**

This Getting Started tutorial provides a step-by-step introduction to **Semarchy Convergence for MDM** (Master Data Management).

**Tip:** If you want to learn about MDM or discover Semarchy Convergence, please visit the Semarchy Demo Center.

**Note:** The **Semarchy Convergence Documentation Library**, including the development, administration and installation guides is available online at the following URL: <http://semarchy.com/download/>

## Contents

1. Getting Started Guide
  1. Overview
  2. Contents
  3. Audience
  4. Document Conventions
  5. Other Semarchy Resources
  6. Obtaining Help
  7. Feedback
2. Overview
  1. What is Convergence for MDM?
  2. Tutorial Overview
3. Installing Semarchy
  1. Requirements
  2. Installing the Semarchy Convergence for MDM Demo
    1. Configure the Database Schemas
    2. Install and Start the Convergence for MDM Server
    3. Install the Repository
    4. Create the Demo Model
4. Designing the MDM Hub
  1. What is a Model?
  2. Creating Customized Data Types
    1. User Defined Types
    2. Complex Types
  3. Designing Entities
    1. Creating Entities
      1. To create the Customer entity:
      2. To add attributes to the Customer entity:
      3. To create a display name for the Customer entity:
    2. Creating References
    3. Review the Diagram

- 4. Defining Attribute Groups
- 5. Defining Constraints
- 6. Understanding the Certification Process
- 7. Enriching Source Data
  - 1. Creating a SemQL Enricher
  - 2. Creating a Plug-in Enricher
    - 1. Enriching Postal Addresses
    - 2. Enriching Phone Numbers
- 8. De-duplicating Data
  - 1. Matching Records
  - 2. Consolidating Records
- 4. Validating the Model
- 5. Working with Integration Jobs
- 5. Deploying the MDM Hub
  - 1. Creating a Data Location
  - 2. Deploying a Model Edition
  - 3. Creating a Root Data Edition
- 6. Running the MDM Hub
  - 1. Publish Data to the MDM Hub
  - 2. Viewing the Log
- 7. Creating an Application
  - 1. Creating Table Views and Form Views
    - 1. Table Views
    - 2. Form Views
  - 2. Creating Business Objects and Business Object Views
    - 1. Creating Business Objects
    - 2. Creating Business Object Views
  - 3. Creating Human Workflows
    - 1. Creating a Data Entry Workflow
    - 2. Creating a Duplicate Management Workflow
- 8. Using the Application
  - 1. Connecting to the Application
  - 2. The Demo Application
  - 3. Navigating the Master Data
    - 1. Browsing
    - 2. Quick Search
    - 3. Advanced Search
    - 4. Export
    - 5. Browsing References
  - 4. Authoring Master Data
    - 1. Creating and Modifying Records
    - 2. Reviewing the Changes
  - 5. Data Stewardship
    - 1. Using the Lineage
    - 2. Managing Duplicates
      - 1. Un-Merging Duplicates
      - 2. Merging Duplicates

- 3. [Using the Home Dashboard](#)
- 4. [Detecting Rejects](#)
- 9. [Summary](#)
  - 1. [Summary](#)
  - 2. [Going Further with Convergence for MDM](#)
  - 3. [Learn More](#)

## Audience

This document is intended for users interested in learning how to use the **Semarchy Convergence for MDM** for their Enterprise Master Data Management Initiatives.

## Document Conventions

This guide uses the following formatting conventions:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or a product specific term or concept.
<i>italic</i>	Italic type indicates special emphasis or placeholder variable that you need to provide.
<code>monospace</code>	Monospace type indicates code example, text or commands that you enter.

## Other Semarchy Resources

In addition to the product manuals, Semarchy provides other resources available on its web site: <http://www.semarchy.com>.

## Obtaining Help

There are many ways to access Semarchy Technical Support. You can call or email our global Technical Support Center ([support@semarchy.com](mailto:support@semarchy.com) ). For more information, see <http://www.semarchy.com/>.

## Feedback

We welcome your comments and suggestions on the quality and usefulness of this documentation. If you find any error or have any suggestion for improvement, please mail [support@semarchy.com](mailto:support@semarchy.com) and indicate the title of the documentation along with the chapter, section, and page number, if available. Please let us know if you want a reply.

## Overview

This Getting Started tutorial provides a step-by-step introduction to Convergence for MDM (Master Data Management).

### What is Convergence for MDM?

Semarchy Convergence for MDM is designed to support any kind of Enterprise Master Data Management initiative. It brings an extreme flexibility for defining and implementing master data models and releasing them to production. Convergence for MDM can be used as the target deployment point for all master data of your enterprise or in conjunction with existing data hubs to contribute to data transparency and quality with federated governance processes. Its powerful and intuitive environment covers all common use cases for setting up a successful master data governance strategy.

For more information, see:

- Why Do I need MDM?
- What is Evolutionary MDM?
- Semarchy Convergence for MDM Product Tour

### Tutorial Overview

In this tutorial, you will:

- **Design** a master data hub containing employees, customers and cost centers information. You will also configure the **Integration Rules** to load, enrich and consolidate data coming from several distinct source systems and produce certified golden records.
- **Deploy** the MDM Hub.
- **Integrate** data from various source systems into the hub, and review the consolidated data.
- **Design an Application** for business users and data stewards to access the MDM hub data.
- Use the application to **Browse** the master data and modify this data through **Human Workflows**.

# Installing Semarchy

## Requirements

Before starting this tutorial, make sure that the following requirements are met:

1. Oracle Database version 10.2 or above is installed and configured.
2. A Java Runtime Environment (JRE) or Development Kit (JDK) 1.6 (Update 24 or above) is installed and the JAVA\_HOME or JRE\_HOME environment variable is configured to point to this installation of Java. Set JAVA\_HOME to your JDK installation directory (e.g., "c:\Progra~1\java\jdk1.6.0") or set JRE\_HOME to the JRE base directory (e.g., "c:\Progra~1\java\jre1.6.0").

## Installing the Semarchy Convergence for MDM Demo

### Configure the Database Schemas

Convergence for MDM uses three schemas for the demonstration environment:

1. SEMARCHY\_DEMO\_REPOSITORY contains the Convergence for MDM Repository.
2. SEMARCHY\_DEMO\_MDM is the schema into which you will deploy your MDM Hub.
3. SEMARCHY\_DEMO\_SOURCE contains sample source data used to load the MDM Hub.

To configure the database schemas:

1. Connect with a system account to the Oracle Database.
2. Run the following script to create the Convergence for MDM demo schemas:

```
CREATE USER SEMARCHY_DEMO_REPOSITORY IDENTIFIED BY  
SEMARCHY_DEMO_REPOSITORY DEFAULT TABLESPACE USERS  
TEMPORARY TABLESPACE TEMP;
```

```
CREATE USER SEMARCHY_DEMO_SOURCE IDENTIFIED BY  
SEMARCHY_DEMO_SOURCE DEFAULT TABLESPACE USERS  
TEMPORARY TABLESPACE TEMP;
```

```
CREATE USER SEMARCHY_DEMO_MDM IDENTIFIED BY  
SEMARCHY_DEMO_MDM DEFAULT TABLESPACE USERS  
TEMPORARY TABLESPACE TEMP;
```

```
GRANT CONNECT,RESOURCE TO SEMARCHY_DEMO_REPOSITORY,  
SEMARCHY_DEMO_SOURCE, SEMARCHY_DEMO_MDM;
```

If you have already created these schemas and wish to delete them beforehand, use the following script:

```
DROP USER SEMARCHY_DEMO_REPOSITORY CASCADE;  
DROP USER SEMARCHY_DEMO_SOURCE CASCADE;
```

```
DROP USER SEMARCHY_DEMO_MDM CASCADE;
```

## Install and Start the Convergence for MDM Server

Convergence for MDM runs as a web application in a Java Application Server. A simple Apache Tomcat Server, pre-configured with Convergence for MDM is used for this tutorial.

In the following section, the `semarchy_with_tomcat.zip` file refers to the *Semarchy Convergence for MDM – Full Setup with Apache Tomcat* file that you can download to install Semarchy Convergence for MDM. The name of this file varies as it includes the Convergence for MDM version and build number.

1. Download the Semarchy Convergence for MDM distribution. Make sure to download the version that includes the pre-configured Tomcat Server. It is named `semarchy_with_tomcat.zip`.
2. Install and start the Apache Tomcat Server:
  - On a Windows Platform:
    1. Uncompress the `semarchy_with_tomcat.zip` file in your machine. This will create a `\semarchy` folder.
    2. Open Windows Explorer, and go to the `\semarchy\bin`.
    3. Run `startup.bat`.
  - On a UNIX/Linux Platform:
    1. Uncompress the `semarchy_with_tomcat.zip` file in your home folder. This will create a `$HOME/semarchy` folder.
    2. In a Shell window, run `$HOME/semarchy/bin/startup.sh`.

**Warning:** The Apache Tomcat server starts by default on the port 8088. If this port is already used by another application and you wish to start the server on a different port, edit the `/conf/server.xml` file with a text editor, and change the port value in the following line: `<Connector port="8088" protocol="HTTP/1.1"`

**Warning** The Convergence for MDM server is preconfigured with three JDBC datasources pointing to the three schemas you created (called `SEMARCHY_DEMO_REPOSITORY`, `SEMARCHY_DEMO_SOURCE` and `SEMARCHY_DEMO_MDM`), stored in an Oracle XE instance started on the local machine and listening on port 1521 (that is: `localhost:xe:1521`). If the Oracle instance that you are using is configured differently: First, stop the Convergence for MDM server, edit the `/conf/catalina/localhost/semarchy.xml` to change the configuration of the datasources and then restart the Convergence for MDM server.

To shut down the Convergence for MDM server, run the `shutdown.bat` (Windows) or `shutdown.sh` (UNIX/Linux) script in the `/bin` folder.

## Install the Repository

Convergence for MDM holds all its information in a repository stored in a database schema. The first task when connecting Convergence for MDM is to create this repository structure in the database schema previously created.

1. Open your web browser and connect to the following URL: `http://localhost:8088/semarchy/` (update the port number if you changed it during the installation process)

2. In the login prompt, enter the following:
  - **User:** *semadmin*
  - **Password:** *semadmin*
3. The Convergence for MDM Workbench opens with the license agreement. Review the End-User License Agreement.
4. Check the **I have read and accept Semarchy's End-User License Agreement** box and then click **Next**.
5. In the **License Key File** page, select a valid license key by clicking the **Upload License Key file...** button and then click **Next**.
6. In the **Repository Creation** wizard, select **Design** for the type of repository and then click **Finish**.
7. Click **OK** when the *Repository Successfully Created* message appears.

The repository has been created and Convergence for MDM is now up and running.

## Create the Demo Model

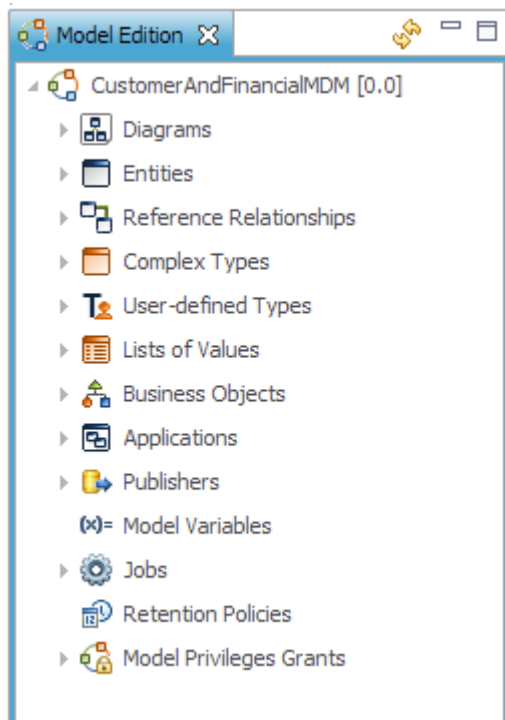
The Tutorial Environment contains a *Customer and Financial Hub* model. This model simulates an MDM project in progress. During this tutorial, you will finish the design of this model, deploy the MDM Hub and load it from sample data sources.

To seed the *Customer and Financial Hub* model:

1. In the Convergence for MDM menu, Select **Help > Getting Started > Create Demo Model...**
2. In the **Getting Started Setup** wizard, Select the **Partial Setup** option and then click **Next**.
3. The second wizard screen allows you to rename the model:
  - If you are running this tutorial for the first time, leave the fields unchanged.
  - If you have already seeded a demonstration model in this repository and want to seed the model a second time with a different name, edit the Demo Model Name (for example, enter *CustomerAndFinancialMDM2*). Note that you will have to use this new name for the rest of the tutorial every time you are requested to enter *CustomerAndFinancialMDM* as a value.
4. Click **Finish**.
5. Click **OK** in the confirmation window.

The demonstration model is seeded, and the **Model Edition** view opens.





Congratulations! You have successfully installed and configured Semarchy Convergence for MDM. You can now proceed and work with the *Customer and Financial Hub* model.

# Designing the MDM Hub

In this chapter, you will design a master data hub containing employees, customers, contacts and cost centers information. You will also configure the integration rules to augment, validate and consolidate data coming from several distinct source systems and produce certified golden records.

## What is a Model?

A Model contains the description of the master data.

A Model in Convergence for MDM is not simply a *physical* data model but a *logical* model. It includes the logical Entity-Relation definition of the objects stored in the hub — the *entities* with their *attributes*. It also includes the various types, constraints and rules that apply to these entities.

The model also contains the definition of the *integration jobs* that run into the platform to create golden data from raw information pushed by *publishers* (third party applications).

In the first part of the tutorial, we will explore the *Customer and Financial Hub* model.

## Creating Customized Data Types

Convergence for MDM includes built-in data types for the attributes of the model's entities, such as String, Number, etc.

You can declare customized types that will be reused across the model, including:

- *User-Defined Types* are a restriction of a built-in data type.
- *Complex Types* are composite types made of attributes.
- *List of Values* or *LOV* are list of Code/Label pairs.

## User Defined Types

1. Expand the **User-defined Types** node in the Model Edition.
2. Double-click the **GenericNameType** node. The editor for this user-defined type opens.
  - In the **Details** group, you can see that this user-defined type is based on the *String* built-in type, but with a fixed Length of 80 characters.
  - Click the **Used in** group (in the left banner of the editor) to see the attributes in the model using this user-defined type. For example, the Employee's First Name and Last Name use this type. If the User-Defined type is modified (for example, if we extend the length to 120 characters), every attribute in the model using this type will automatically benefit from this change.
3. Close this **GenericNameType** editor by clicking the Close (a cross) icon on the editor tab.

## Complex Types

A *Complex Type* is a user-created type that contains several attributes.

In this example, we will create a complex type called *SimpleAddressType* to represent a postal address. This type will have the following definition attributes: Address, PostalCode, City and Country. We will reuse the *GenericNameType* user-defined type for some of these definition attributes.

**Tip: Auto fill** is a feature in Convergence for MDM that automatically generates field values such as Labels for objects when you provide their name in *CamelCase*. CamelCase consists in having all words joined without spaces, with each word's initial letter capitalized (Examples: *FirstName*, *ZipCode*, *SubjectArea*). CamelCase is a recommended naming convention for objects in Convergence for MDM. For this tutorial, we will use this naming convention to benefit from the Auto fill feature.

To create the *SimpleAddressType* complex type:

1. Right-click the **Complex Types** node and select **Add Complex Type**
2. In the **Create New Complex Type** wizard, enter the following values:
  - **Name:** *SimpleAddressType*. Note that as the **Auto Fill** box is checked, the **Label** is automatically filled in.
3. Click **Finish** to close the wizard. The **Complex Type: SimpleAddressType** editor opens.
4. In the **Description** field, enter the following description: *Type that can be used to represent a simple address.*
5. Select the **Definition Attributes** group (in the left banner of the editor).
6. Click the **Add Definition Attribute**



button in the **Definition Attributes** table to add a new attribute to the complex type.

7. In the **Create New Definition Attribute** dialog, set the following properties.
  - **Name:** *Address*
  - **Type:** *GenericNameType [User-defined Type]*
8. Click **Finish** to close the dialog.
9. Click the **Add Definition Attribute**



button.

10. In the **Create New Definition Attribute** dialog, set the following properties.
  - **Name:** *PostalCode*
  - **Type:** *String [Built-in Type]*
  - **Length:** *20*
11. Click **Finish** to close the dialog.
12. Click the **Add Definition Attribute**



button.

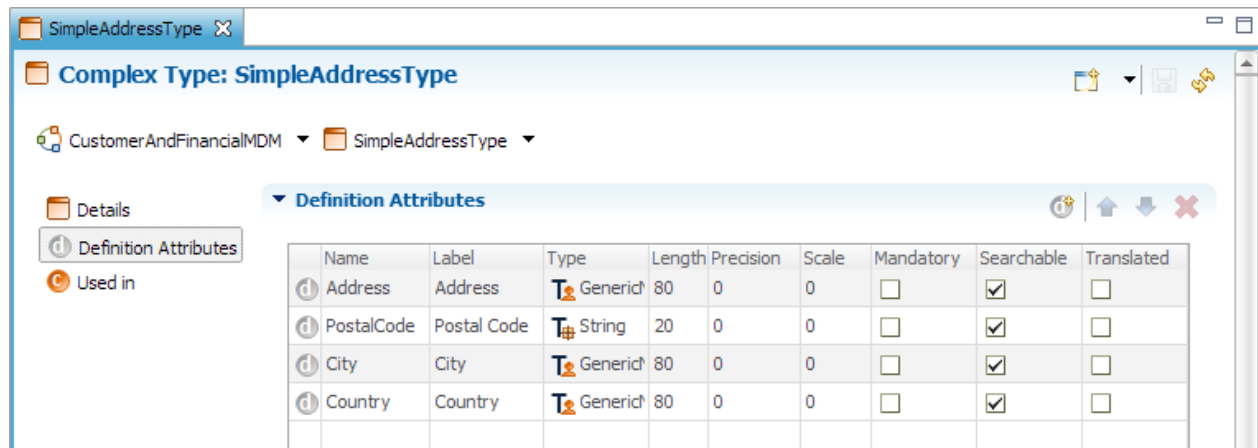
13. In the **Create New Definition Attribute** dialog, set the following properties.
  - **Name:** *City*
  - **Type:** *GenericNameType [User-defined Type]*
14. Click **Finish** to close the dialog.
15. Click the **Add Definition Attribute**



button.

16. In the **Create New Definition Attribute** dialog, set the following properties.
  - **Name:** *Country*

- **Type:** *GenericNameType* [User-defined Type]
- Click **Finish** to close the dialog.
  - Select **File > Save** in the menu. You can alternately press the **Save** button in the toolbar or **CTRL+S** to save the editor.
  - Close the **Complex Type: SimpleAddressType** editor.



Convergence for MDM also models how artifacts are displayed. The *Display Name* for *SimpleAddressType* will define how an address stored in this type is displayed in a compact format in the user interface.

To create a display name for the *SimpleAddressType* type:

- Expand the **Complex Types** node in the Model Edition.
- Right-Click the **SimpleAddressType** node in the tree view and then select **Define Display Name**. The **Create New Display Name** wizard appears.
- In first wizard screen, select the content of the **Separator** field and replace it with a space.
- Click **Next**
- In the **Display Name Attributes** page, click the **Add All >>** button to add all available attributes to the **Selected Attributes**. Use the **Move Up** and **Move Down** buttons to arrange them in the following order: *Address*, *Postal Code*, *City* and *Country*.
- Click **Finish** to close the wizard.
- Press **CTRL+S** to save the **Display Name: SimpleAddressType** editor.
- Close the editor.

With this display name, an address stored in a *SimpleAddressType* will display as all the selected attributes, separated by a space.

## Designing Entities

### Creating Entities

An entity represents an object in the MDM hub; For example Customers, Contacts or Parties are entities. Usually, MDM entities are clearly defined and the definition agreed at corporate level.

In this example, we will create a *Customer* **Entity**, and use the customized data types for its attributes. The *Customer* entity represents for our organization the *companies we are doing business with*. Customer data is scattered and duplicated in various applications. This entity will be designed to de-duplicate (fuzzy matching) and consolidated this data into single golden customer records.

#### To create the Customer entity:

1. In the **Model Edition** view, expand the



**Diagrams** node.

2. Double-click the



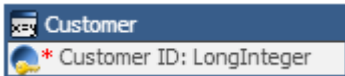
**MainDiagram** node. The **MainDiagram** editor opens. This diagram displays the entities of the model in a graphical way, and allows creating or modifying these entities. This model already contains the *Contact*, *Employee* and *CostCenter* entities.

3. In the **Palette** (at the right of the editor), click to select the



**Add Entity** tool.

4. Click in the diagram. The **Create New Entity** wizard opens.
5. In the **Create New Entity** wizard first screen, make sure that the **Auto Fill** box is checked and then enter the following:
  - **Name:** *Customer*
  - **Plural Label:** *Customers*
  - Select *Fuzzy Matching* for the **Matching Behavior**.
6. Click **Next**
7. In the **Primary Key Attribute** screen, make sure that the **ID Generation** is set to *Sequence* starting with a value of 1.
8. Click **Finish** to close the wizard. The new **Customer** entity is added to the diagram.



9. Press **CTRL+S** to save the editor.
10. Double-click the title of the **Customer** entity in the diagram. The **Entity: Customer** editor opens.
11. In this editor, enter the following value in the **Description** Field: *Entity that represents all customers we are doing business with. Customers come from various source applications such as the CRM and Marketing applications.*
12. Press **CTRL+S** to save the editor and leave it open.

#### To add attributes to the Customer entity:

Now that the *Customer* entity is created, we will add **Attributes** (fields) to this entity, such as the *CustomerName*. Each attribute will be of a certain type (built-in, user-defined, complex, etc.). We will also define whether these attributes are mandatory or not.

1. In the **Entity: Customer** editor, select the



**Attributes** section (in the left banner of the editor). The list of attributes is displayed. It already contains the primary key field *CustomerID* defined when we created the entity.

2. In the **Attributes** table, select the



**Add Simple Attribute** button.

3. In the **Create New Simple Attribute** dialog, make sure that the **Auto Fill** box is checked and then enter the following:

- **Name:** *CustomerName*
- **Type:** *GenericNameType [User-defined Type]*
- Check the **Mandatory** box.
- **Mandatory Validation Scope:** *Pre and Post Consolidation*

4. Click **Finish**. The *CustomerName* attribute is created. It is a mandatory attribute for this entity.

5. In the **Attributes** table, select the



**Add Simple Attribute** button.

6. In the **Create New Simple Attribute** dialog, make sure that the **Auto Fill** box is checked and then enter the following:

- **Name:** *TotalRevenue*
- **Type:** *Integer [Built-in Type]*
- **Mandatory:** *No*

7. Click **Finish**.

8. In the **Attributes** table, select the



**Add Complex Attribute** button.

9. In the **Create New Complex Attribute** dialog, make sure that the **Auto Fill** box is checked and then enter the following:

- **Name:** *InputAddress*
- **Physical Prefix:** *INP*
- **Complex Type:** *SimpleAddressType [Complex Type]*

10. Click **Finish**. The complex attribute *InputAddress* is now created.

11. In the **Attributes** table, select the



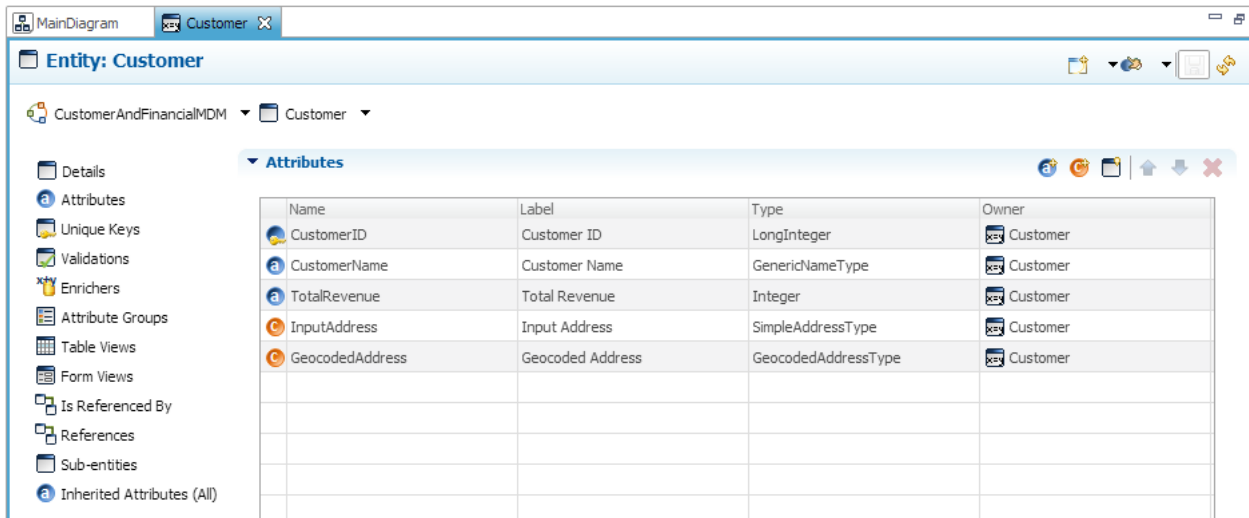
**Add Complex Attribute** button.

12. In the **Create New Complex Attribute** dialog, make sure that the **Auto Fill** box is checked and then enter the following:

- **Name:** *GeocodedAddress*
- **Physical Prefix:** *GEO*
- **Complex Type:** *GeocodedAddressType [Complex Type]*

13. Click **Finish**.

14. Press **CTRL+S** to save the **Entity: Customer** editor and leave it open.



**Note:** The **Physical Table Name**, **Physical Column Name** and **Physical Prefix**, define the name of the physical database objects created for the entities and attributes. As this tutorial includes loading scripts targeting predefined database objects, it is important to use the correct physical prefix and names when designing the model, in order to deploy physical database objects with the expected names.

#### To create a display name for the Customer entity:

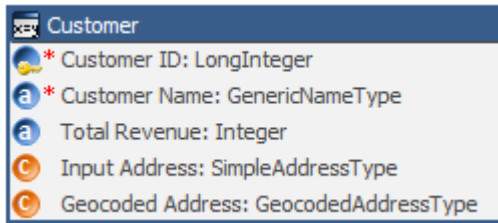
We will now define the **Display name** for the entity, which is the way to display it in a compact way (for example, in a tree view). For customers, we only want to show the *CustomerName*.

1. Select the **Customer** node in the **Outline** view (right of the screen), right-click and then select **Define Display Name**. The **Create New Display Name** wizard appears.
2. In first wizard screen, select the content of the **Separator** field, and replace it with a space.
3. Click **Next**
4. In the **Display Name Attributes** screen, select the *CustomerName* attribute and click the **Add >>** button to add it to the **Selected Attributes**.
5. Click **Finish** to close the wizard.
6. Press **CTRL+S** to save the **Entity: Customer** editor and leave it open.

You can now return to the **MainDiagram** editor by selecting it.



The diagram now displays the Customer entity with its various attributes as shown below.



**Congratulations!** You have successfully created your first entity!


**Tip:** Creating attributes can also be done directly in the diagram by right-clicking on any of the entities and selecting the appropriate actions

## Creating References

Entities are related using **References Relationships**. We will now create the following relations on the newly created *Customer* entity:

- *CustomerHasAccountManager*, representing the fact that a *Customer* is managed by one *Employee* (his Account Manager).
- *ContactBelongsToCustomer*, representing the fact that a *Contact* (a person) is directly attached to a *Customer* (a company).

To create the *CustomerHasAccountManager* relation:

1. In the **Palette** of the **MainDiagram** editor, click to select the  **Add Reference** tool.
2. Click and drag from **Customer** to **Employee**. The **Create New Reference Relationships** wizard opens.
3. In the **Create New Reference Relationships** wizard screen, make sure that the **Auto Fill** box is checked and then enter the following:
  - **Name:** *CustomerHasAccountManager*
  - **Validation Scope:** *Pre and Post Consolidation*
  - In the **Referencing** group, check that **Referencing Entity** is set to *Customer [Entity]*.
  - In the **Referencing** group, set the following values:
    - **Referencing Role Name:** *Customers*
    - **Referencing Role Label:** *Managed Customers*
  - In the **Referenced** group, check that **Referenced Entity** is set to *Employee [Entity]*.
  - In the **Referenced** group, set the following values:
    - **Referenced Role Name:** *AccountManager*
    - Select the **Mandatory (One to Many)** option.
4. Review the content of the wizard.



Create New Reference Relationship

**Reference Relationship**

A reference is a relationship from an entity to another entity. It translates into a single ER foreign key.

Auto Fill: ☒

Name: CustomerHasAccountManager

Physical Name: CUSTOMER\_HAS\_ACCOUNT\_MANA

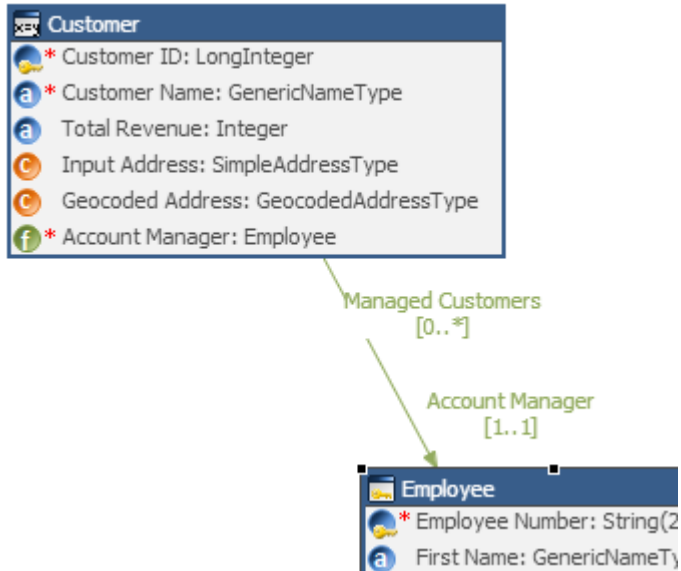
Label: Customer Has Account Manager

Validation Scope: Pre and post-consolidation

Referencing [0..*]	Referenced [0..1]
Referencing Entity: Customer [Entity]	Referenced Entity: Employee [Entity]
Referencing Role Name: Customers	Referenced Role Name: AccountManager
Referencing Role Label: Managed Customers	Referenced Role Label: Account Manager
Referencing Navigable: <input checked="" type="checkbox"/>	Referenced Navigable: <input checked="" type="checkbox"/>
	Mandatory (One To Many): <input checked="" type="checkbox"/>
	Physical Name: ACCOUNT_MANAGER

Finish Cancel


- Click **Finish** to close the wizard. The reference appears now in the diagram as an arrow linking the *Customer* entity to the *Employee* entity.



6. Press **CTRL+S** to save the editor.

This reference has a *Pre and Post Consolidation Validation Scope*: On the source data, but also after consolidating the candidate customer golden records, we will check that customers reference valid account managers. As the reference is marked as **Mandatory**, we will not allow null values. Customer records that do not respect these rules will be rejected as errors.

To create the *ContactBelongsToCustomer* relation:

1. In the **Palette** of the **MainDiagram** editor, click to select the  **Add Reference** tool.
2. Click and drag from **Contact** to **Customer**. The **Create New Reference Relationships** wizard opens.
3. In the **Create New Reference Relationships** wizard screen, make sure that the **Auto Fill** box is checked and then enter the following:
  - **Name**: *ContactBelongsToCustomer*
  - **Validation Scope**: *Pre and post-consolidation*
  - In the **Referencing** group, check that **Referencing Entity** is set to *Contact [Entity]*.
  - In the **Referencing** group, set the following values:
    - **Referencing Role Name**: *Contacts*
    - **Referencing Role Label**: *Contacts*
  - In the **Referenced** group, check that **Referenced Entity** is set to *Customer [Entity]*.
  - In the **Referenced** group, select the **Mandatory (One to Many)** option.
4. Review the content of the wizard.

**Create New Reference Relationship**

**Reference Relationship**

A reference is a relationship from an entity to another entity. It translates into a single ER foreign key.

Auto Fill: ☒

Name:

Physical Name:

Label:

Validation Scope:

**Referencing [0..\*]**

Referencing Entity:

Referencing Role Name:

Referencing Role Label:

Referencing Navigable: ☒

**Referenced [0..1]**

Referenced Entity:

Referenced Role Name:

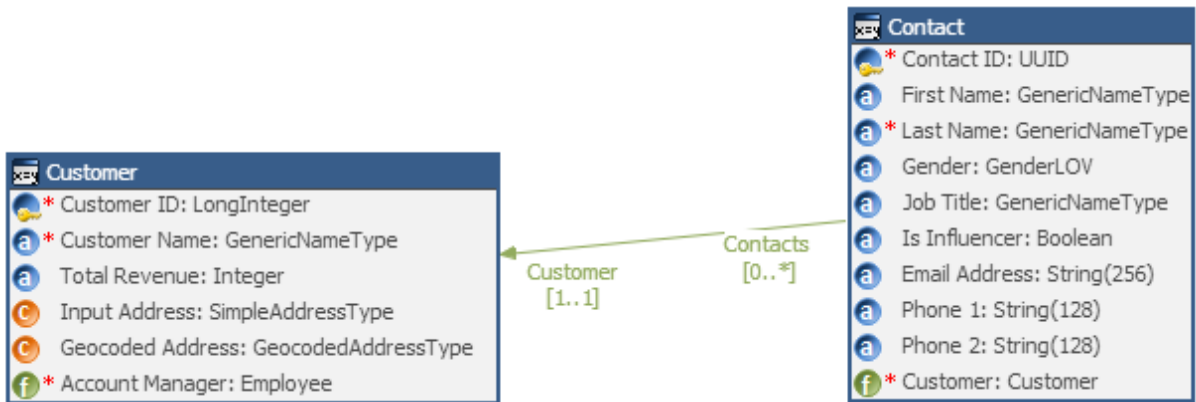
Referenced Role Label:

Referenced Navigable: ☒

Mandatory (One To Many): ☒

Physical Name:

- Click **Finish** to close the wizard. The reference appears now in the diagram as an arrow linking the *Contact* entity to the *Customer* entity.

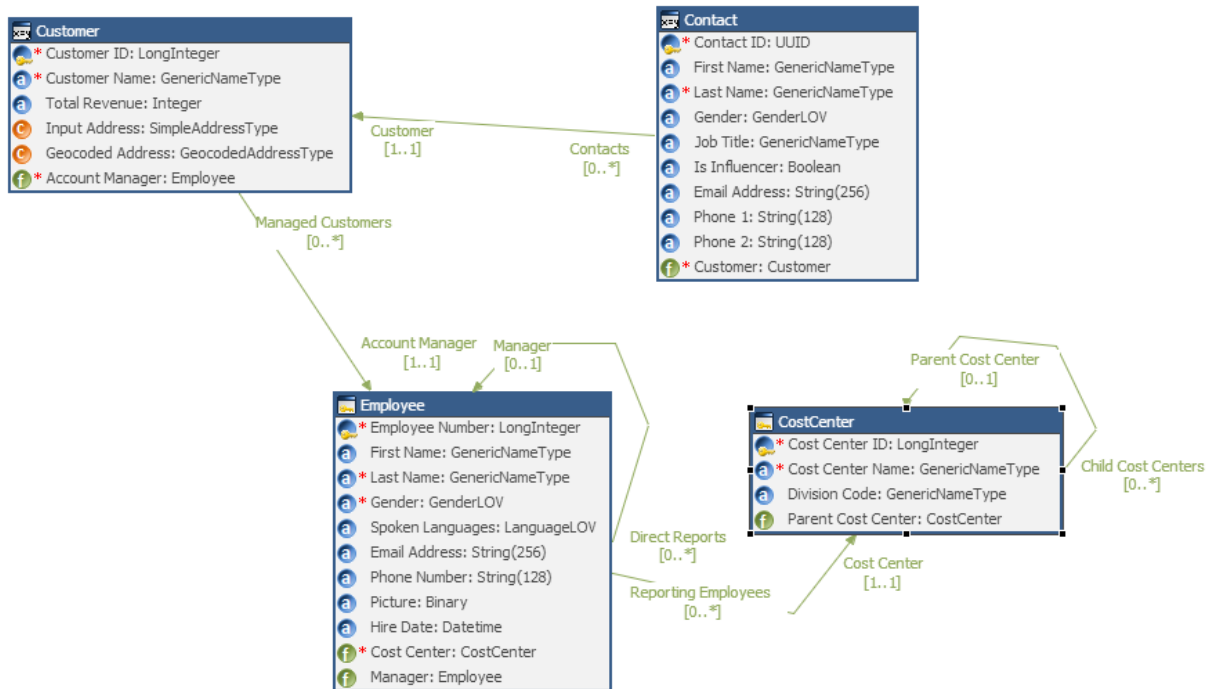


6. Press **CTRL+S** to save the editor.

This reference is marked as **Mandatory** and its **Validation Scope** is *Pre and post-consolidation*: We will check that contacts are always attached to valid customers in the source records as well as in the candidate golden records.

## Review the Diagram

You can rearrange the diagram elements as shown below, and close the diagram editor.



**Congratulations!** You have now related the *Customer* entity to the rest of the logical model.

## Defining Attribute Groups

**Attribute Groups** define how the entity's attributes are displayed and grouped into sections in the default data forms. We will design such a default view now. For business users and data stewards who want to browse and edit data in a more friendly way, we will create later a dedicated **Application**.

To create an attribute group:

1. Return to the *Customer* entity editor.
2. In the **Outline** view, expand the **Customer** Node.

3. Right-click the **Attribute Group** node and then select **Add Attribute Group**
4. In the **Create New Attribute Group** wizard, enter the following:
  - **Name:** *CustomerDetail*
5. Click **Next**
6. In the **Group Attributes** screen, click **Add All >>** button to add all attributes to the **Group Attributes**.
7. Using the **Move Up/Move Down** buttons, arrange these attributes in the following order: *CustomerID*, *CustomerName*, *TotalRevenue*, *AccountManager*, *InputAddress* and *GeocodedAddress*.
8. Click **Finish** to create the Attribute Group. You return to the **Entity: Customer** editor.
9. Press **CTRL+S** to save this editor and leave it open.

## Defining Constraints

While creating attributes and references, we have introduced data quality rules (mandatory attributes and references, references that must point to a valid record). We will now create a **Validation** to enforce a certain level of completeness on the customer's postal address.

Constraints add validation points on entities, which can run while refining information from a variety of source into golden data.

Constraints may be mandatory attributes, unique keys, validations, list of values, etc. Reference relationships are also constraints as they may define referential integrity between entities.

Constraints are checked at various points of the integration process:

- A **Pre-Consolidation** validation validates the records before they are matched, de-duplicated and consolidated. Only valid records for these validations participate to the matching/de-duplication and consolidation phase.
- A **Post-Consolidation** validation validates the records after they are de-duplicated and consolidated, and before they become golden.

To add a validation:

1. In the **Outline** view, right-click the **Validations** node and then select **Add SemQL Validation**
2. In the **Create New SemQL Validation** wizard, enter the following:
  - **Name:** *ValidateAddressCompleteness*
  - **Label** is auto filled with: *Validate Address Completeness*
  - **Description:** *Source address should contain at least one address line and either a postal code or a city.*
  - **Condition:** `InputAddress.Address is not null and ( not ( InputAddress.PostalCode is null and InputAddress.City is null ) )`
  - **Validation Scope:** *Pre-Consolidation Only*
3. Click **Finish** to create the validation. You return to the **Entity: Customer** editor.
4. Press **CTRL+S** to save this editor and leave it open.

Note that the condition, as well as the transformation expressions in Convergence for MDM are defined using the SemQL Language, which is executed within the Oracle Database engine. Convergence for MDM leverages the transformation capabilities and processing power of the database hosting the MDM Hub, providing the best performances for building golden records from very large source data volumes.

## Understanding the Certification Process

Source data is published to the MDM Hub from enterprise applications (for example via an ETL tool), or entered by users through human workflows.

This information is frequently incomplete, inconsistent, not standardized, and contains data that violates some of the constraints or restrictions defined for the entities. There is also duplicate data that must be matched, de-duplicated and consolidated into the Golden Records.

The certification process is a key element of master data management. It uses the various rules defined in the model to refine the imperfect source data into Golden Data.

## Enriching Source Data

The process that certifies source data into golden data uses the data quality rules previously defined in the tutorial. It must also include **Enrichers** to augment and standardize the source data.


In this example, we will create several *Enrichers*:

- A **SemQL Enricher** that uses the SemQL language and the Oracle Database to enrich data submitted by the publishers for the Customer entity.
- A **Plug-in Enricher** that enriches and standardizes the employees' phone numbers.
- A **Plug-in Enricher** that uses the Google Maps API for enriching the customer address with geographical information.

Both plug-in enrichers illustrate the Convergence for MDM extensibility framework.

## Creating a SemQL Enricher

To create a SemQL enricher:

1. In the **Outline** view, right-click the **Enrichers** node and then select **Add SemQL Enricher**
2. In the **Create New SemQL Enricher** wizard, enter the following:
  - **Name:** *StandardizeCustomerData*
3. Click **Next**
4. In the **Enricher Expressions** screen, double-click the *Customer Name* line in the **Available Attributes** list. It is added to the **Used Attributes** list.
5. Repeat the previous operation to add *InputAddress.Address*, *InputAddress.City* and *InputAddress.Country* to the **Used Attributes** list.
6. Click **Finish**. The **SemQL Enricher: StandardizeCustomerData** editor appears. This editor contains the list of attributes to enrich in the **Enricher Expressions** table, but there is no expression to enrich these attributes.
7. Define the expression for *CustomerName*:
  1. In the **Enricher Expression** table, select the **Expression** column for the *CustomerName* attribute and then click the 
  2. In the **Expression** field, enter: `Upper( CustomerName )`
  3. Click **OK** to close the SemQL editor.
8. Repeat these operations to create the following Enricher Expressions:

- `InputAddress.Address: Upper( InputAddress.Address )`
- `InputAddress.City: Regexp_replace( Upper( InputAddress.City ), '[^A-Z ]+', '' )`
- `InputAddress.Country: Upper( InputAddress.Country )`

9. Press **CTRL+S** to save this editor and leave it open.

**SemQL Enricher: StandardizeCustomerData**

CustomerAndFinancialMDM ▾ Customer ▾ StandardizeCustomerData ▾

**Details**

**Name and Description**

Name:

Label:

Description:

Active: ☒

Position:

**Condition**

Filter:

**Enricher Expressions**

	Attribute Name	Expression
f(x)	CustomerName	Upper( CustomerName )
f(x)	InputAddress.Address	Upper( InputAddress.Address )
f(x)	InputAddress.City	Regexp_replace( Upper( InputAddress.City ), '[^A-Z ]+', '' )
f(x)	InputAddress.Country	Upper( InputAddress.Country )

## Creating a Plug-in Enricher

Plug-in Enrichers process information provided to them as *Inputs* and deliver *Outputs*. When using such an enricher, you need to map on the plug-in input and output the attributes of the entity to be enriched.

For example, the *Google Maps Enricher* that is used in this example takes as an input a basic address information (address line, city name, country and postal code) and outputs a fully geocoded address (street name/number, longitude/latitude, etc.) .



## Enriching Postal Addresses

### Note on the Postal Address Enricher

This enricher uses the Google Maps Service and requires an Internet connection. Creating this enricher is an optional step. If you do not have access to the Internet, you can skip this step and go to the Enriching Phone Numbers step. Per Google's terms of service, there is a limitation to

the number of requests you can perform on this service. Please refer to the Google Site for more information.

To create a plug-in enricher for addresses:

1. In the **Outline** view, right-click the **Enrichers** node and then select **Add Plug-in Enricher**
2. In the **Create New Plug-in Enricher** wizard, enter the following:
  - **Name:** *GeocodeInputAddressUsingGoogle*
  - **Label** is auto filled with: *Geocode Input Address Using Google*
  - **Plug-in ID:** *Google Maps Enricher – com.semarchy.integration...*
3. Click **Finish** to create the Plug-in Enricher. The **Plug-in Enricher: GeocodeInputAddressUsingGoogle** editor appears.
4. In the **Description** field, enter: *Illustrates the use of the Google Maps Geocoder Plug-in.*
5. Deselect the **Active** box if you are not connected to the Internet.
6. Now, you must provide values for the plugin input parameters. To map the *Address Line* to the plug-in input:
  1. In the **Plug-in Inputs** table, select the **Expression** column for the *Address Line* input and then click the  **Edit Expression** button in the cell. The **SemQL Editor** appears.
  2. In the **Expression**, enter *InputAddress.Address*. You can alternately double-click this attribute in the **Attributes** list available on the left side of the editor to add it to the expression.
  3. Click **OK** to close the SemQL Editor.
7. Repeat the same operation to map the other Plug-in inputs with the following attributes:
  - City Name: *InputAddress.City*
  - Country: *InputAddress.Country*
  - PostalCode: *InputAddress.PostalCode*
8. Map the plug-in outputs:
  1. Scroll down in the Editor, and in the **Plug-in Outputs** table, click the  **Define Plug-in Outputs** button. The **Define Output Bindings** dialog appears.
  2. In this dialog, select all the attributes starting with *GeocodedAddress* line in the **Available Attributes** list and then click the **Add >>** button. All these attributes are added to the **Used Attributes** list.
  3. Click **Finish** to close the dialog. The attributes from the entity that were selected now appear in the Plug-in Outputs table. They must be mapped to the output of the plug-in.
9. Define the bindings for the *GeocodedAddress.Region* attribute.
  1. In the **Plug-in Outputs** table, click the **Output Name** column on the *GeocodedAddress.Region* plug-in output line. This column is a drop-down list from which you can now select *Administrative Level 1 (Long)*.
10. Repeat the same operation to map the other attributes of the *GeocodedAddress* complex type attribute of the *Customer* entity on the plug-in outputs according to the list below:
  - *GeocodedAddress.BoxEast*: *East bound longitude*
  - *GeocodedAddress.BoxNorth*: *North bound latitude*
  - *GeocodedAddress.BoxSouth*: *South bound latitude*
  - *GeocodedAddress.BoxWest*: *West bound longitude*



- GeocodedAddress.CoordLatitude: *Latitude*
- GeocodedAddress.CoordLongitude: *Longitude*
- GeocodedAddress.Country: *Country (Long)*
- GeocodedAddress.CountryCode: *Country (Short)*
- GeocodedAddress.Locality: *Locality (Long)*
- GeocodedAddress.PostalCode: *Postal Code(Long)*
- GeocodedAddress.Quality: *Quality*
- GeocodedAddress.Street: *Route (Long)*
- GeocodedAddress.StreetNum: *Street Number (Short)*

11. Press **CTRL+S** to save this editor and leave it open.

MainDiagram
Customer

**Plug-in Enricher: GeocodeInputAddressUsingGoogle**

CustomerAndFinancialMDM
 Customer
 GeocodeInputAddressUsingGoogle

Details

**Name and Description**

Name:

Label:

Description:

Plug-in ID:

Plug-in Provider: Semarchy

Plug-in Description:

Plug-in Legal Notice:

Active: ☐

Position: 1

**Condition**

**Plug-in Params**

**Plug-in Inputs**

Input Name	Expression
Address Line	InputAddress.Address
City Name	InputAddress.City
Country	InputAddress.Country
Postal Code	InputAddress.PostalCode

**Plug-in Outputs**


Attribute Name	Output Name
GeocodedAddress.BoxEast	East bound longitude
GeocodedAddress.BoxNorth	North bound latitude
GeocodedAddress.BoxSouth	South bound latitude
GeocodedAddress.BoxWest	West bound longitude
GeocodedAddress.CoordLatitude	Latitude
GeocodedAddress.CoordLongitude	Longitude
GeocodedAddress.Country	Country (Long)
GeocodedAddress.CountryCode	Country (Short)

## Enriching Phone Numbers

### Note on the Phone Number Enricher

This enricher uses a built-in library that parses and standardizes international and national phone numbers and optionally infer geographical information from the phone numbers. For this example, we will standardize the phone numbers from the US employees in the national format '(ddd) ddd-dddd'.

To create a plug-in enricher for employee phone numbers:

1. In the **Model Edition** view, expand the **Entities > Employee** node.
2. Select the **Enrichers** node, right-click and then select **Add Plug-in Enricher**.
3. In the **Create New Plug-in Enricher** wizard, enter the following:
  - **Name:** *StandardizeEmployeePhoneNumber*
  - **Label** is auto filled with: *Standardize Employee Phone Number*
  - **Plug-in ID:** *Convergence Phone Enricher – com.semarchy.engine...*
4. Click **Finish** to create the Plug-in Enricher. The **Plug-in Enricher: StandardizeEmployeePhoneNumber** editor appears.
5. In the **Description** field, enter *Standardizes the Employee phone number to the US national format using the built-in plugin.*
6. Now, you must provide values for the plugin input parameters. To map the *PhoneNumber* to the plug-in input:
  1. In the **Plug-in Inputs** table, select the **Expression** column for the *Input Phone Number* input and then click the   
**Edit Expression** button in the cell. The **SemQL Editor** appears.
  2. In the **Expression**, enter *PhoneNumber*. You can alternately double-click this attribute in the **Attributes** list available on the left side of the editor to add it to the expression.
  3. Click **OK** to close the SemQL Editor.
7. Repeat the previous step to perform the following mappings:
  - *Country Code:* 'US'. This input indicates to the enricher that the input phone numbers are from the US.
  - *Enriched Phone Format:* 'NATIONAL'. This input indicates to the enricher that the output phone numbers should be formatted in national format: '(ddd) ddd-dddd'.
8. Map the plug-in outputs:
  1. Scroll down in the Editor, and in the **Plug-in Outputs** table, click the   
**Define Plug-in Outputs** button. The **Define Output Bindings** dialog appears.
  2. In this dialog, select the *PhoneNumber* attribute in the **Available Attributes** list and then click the **Add >>** button. This attribute is added to the **Used Attributes** list.
  3. Click **Finish** to close the dialog. The *PhoneNumber* attribute now appears in the Plug-in Outputs table. It must be mapped to the output of the plug-in.
  4. In the **Plug-in Outputs** table, click the **Output Name** column on the *PhoneNumber* plug-in output line. This column is a drop-down list from which you can now select *Enriched Phone Number*.
9. Press **CTRL+S** to save this editor and then close it.

## De-duplicating Data

Now that we have defined the rules to augment/standardize the source data as well as the rules to check their quality, we will define how to handle duplicate customer records. This operation takes place in two phases: First, the duplicate records are detected in a phase called **Matching**. Afterwards, information from all these duplicates is assembled in a single master record. This second phase is called **Consolidation**.

### Matching Records

Record matching consists in identifying two or more records that are duplicates. For example, an employee with a typo in his name in one of the applications may cause a duplicate in the golden data. Detecting this employee as a duplicate is key to obtaining real unique records in the MDM Hub. This matching process requires a preliminary *binning* mechanism to accelerate the matching process. Binning consists in creating smaller sub sets of records. Matching will take place on these subsets to avoid matching every record against the whole data set.

To create a matcher:

1. In the **Outline** view, right-click the **Customer > Matcher** node and then select **Define SemQL Matcher**
2. In the **Create New SemQL Matcher** wizard, enter the following:
  - **Description:** *Find customer duplicates using name and address*
3. Click **Finish**. The **SemQL Matcher – Customer** editor appears.
4. In the **Binning Expressions** group, click the



**Add Binning Expression** button. The SemQL Editor opens.

5. In the SemQL Editor, enter the following expression: `InputAddress.Country`
6. Click **OK** to close the SemQL Editor. We have now divided the sub sets of possible matches by considering only customers belonging to the same country as possible matching candidates.
7. In the **Matching Expression** group, click the

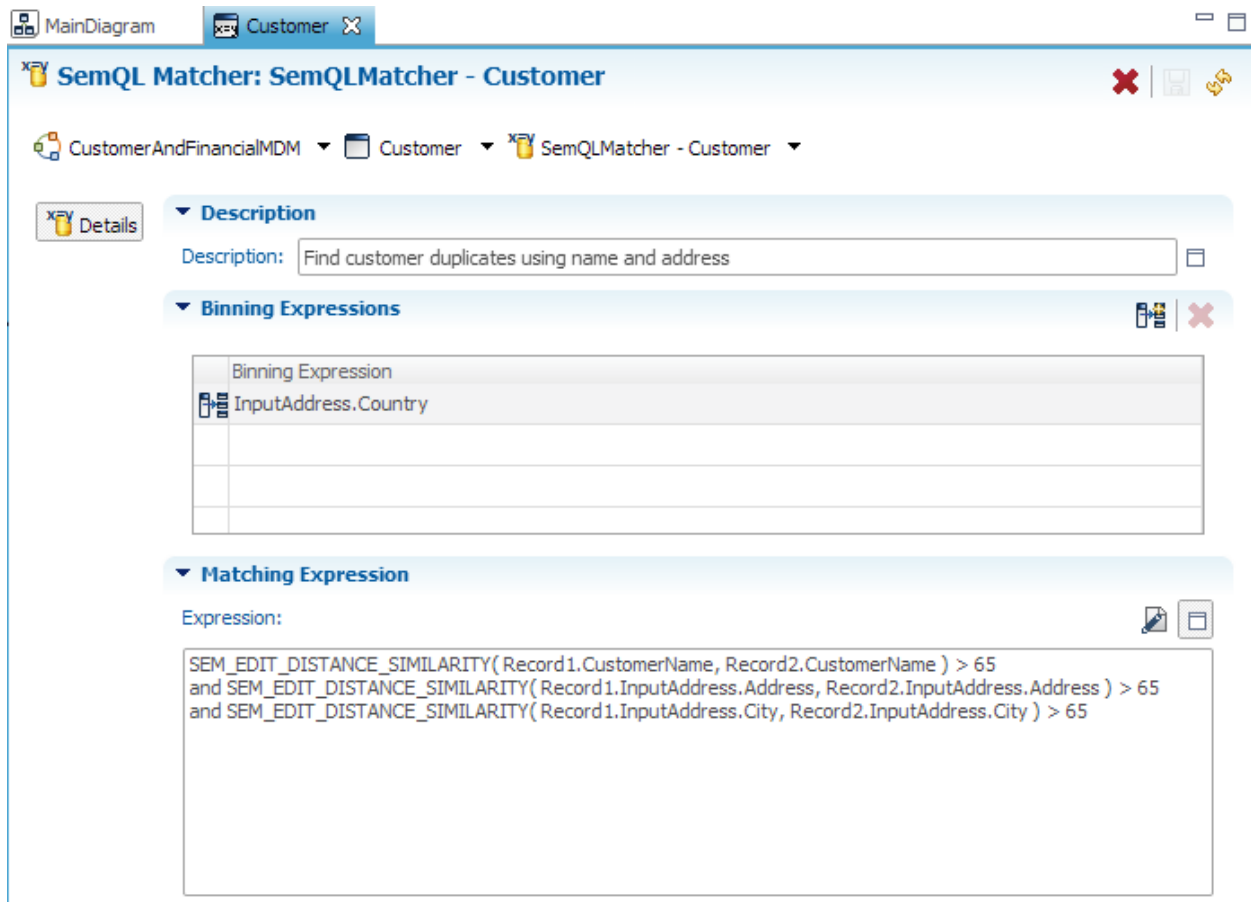


**Edit Matching Expression** button. The SemQL Editor opens.

8. In the SemQL Editor, enter the following expression:

```
SEM_EDIT_DISTANCE_SIMILARITY( Record1.CustomerName, Record2.CustomerName ) >
65
and SEM_EDIT_DISTANCE_SIMILARITY( Record1.InputAddress.Address,
Record2.InputAddress.Address ) > 65
and SEM_EDIT_DISTANCE_SIMILARITY( Record1.InputAddress.City,
Record2.InputAddress.City ) > 65
```

1. Click **OK** to close the SemQL Editor.
2. Press **CTRL+S** to save this editor and leave it open.



You have created a matching rule that expresses that two customer records would be considered similar if they match all these rules:




1. They belong to the same country (as expressed in the binning expression)
2. They have a strong similarity on their name (greater than 65%) using the Levenshtein distance algorithm:  
`SEM_EDIT_DISTANCE_SIMILARITY( Record1.CustomerName, Record2.CustomerName ) > 65`
3. They have a strong similarity on their address line and city (greater than 65%) using the Levenshtein distance algorithm:  
`SEM_EDIT_DISTANCE_SIMILARITY( Record1.InputAddress.Address, Record2.InputAddress.Address ) > 65 and SEM_EDIT_DISTANCE_SIMILARITY( Record1.InputAddress.City, Record2.InputAddress.City ) > 65`

You can of course use other complex rules and matching techniques such as Jaro-Winkler distance, Name normalization or Soundex.

## Consolidating Records

When a set of records is identified as a duplicates group, a single **golden** record is created out of it. The *consolidation* process consists in taking the best data from each record to create this golden record. For example, to build a golden customer record, we may take the address and phone number from the shipping application, and the financial numbers from the billing system. The resulting golden record would be a composite of information from several sources.

To create the *Customer* consolidator:

1. Expand **Entities > Customer** in the Model Edition.
2. Right-click the **Consolidator** node and then select **Define Consolidator**
3. In the **Create New Consolidator** wizard, enter the following:
  - **Consolidator Type:** *Field Level Consolidation*. With this type, the consolidation strategy can be different for each attribute.
  - **Description:** *Consolidate customer golden records – Field-level consolidation*.
4. Click **Finish** to close the wizard. The **Consolidator – Customer** editor appears.
5. Define the consolidation strategy for *CustomerName* and *AccountManager*.
  1. In the **Field Level Consolidators** table, press and hold down the control key (CTRL) and then click the *CustomerName* and *AccountManager* attributes to select them.
  2. Click the  **Field Level Consolidation Strategy** button. **Define**
  3. In the **Define Selected Field-Level Consolidator** wizard, select **Most Frequent Value** and then click **Finish**.
6. Define the consolidation strategy for the geographical attributes.
  1. Select all the other attributes except *TotalRevenue*, that is all attributes starting with *GeocodedAddress* and *InputAddress*.
  2. Click the **Define Field Level Consolidation Strategy** button.
  3. In the  **Selected Field-Level Consolidator** wizard, select **Preferred Source** and then click **Next**. **Define**
  4. In the **Publisher Ranking** table, click the  **Add Publisher Ranking** button.
  5. Double-click the *DataEntry* in the **Available Publishers** list to add to the **Publishers** list.
  6. Repeat this operation to add *CRM*, *Marketing*, *Finance* and then *HumanResources* in this order.
  7. Click **Finish** and then **Finish** again to return to the **Consolidator – Customer** editor.
7. Define the consolidation strategy for *TotalRevenue*.
  1. Double-click the *TotalRevenue* line. The **Define Selected Field-Level Consolidator** wizard appears.
  2. Select **Largest Value** and then click **Finish**.
8. In the **Additional Order By** field, enter *SourceID asc*
9. Press **CTRL+S** to save this editor and leave it open.

**Consolidator: Consolidator - Customer**

CustomerAndFinancialMDM > Customer > Consolidator - Customer

**Description**

Consolidator Type: Field Level Consolidation

Description: Consolidate customer golden records – Field-level consolidation.

Additional Order By : SourceID asc

**Field Level Consolidators**

Attribute Name	Field Level Consolidation Strategy
AccountManager	Most frequent value
CustomerName	Most frequent value
GeocodedAddress.E	Preferred Source (DE,CRM,MKT,FIN,HR)
GeocodedAddress.E	Preferred Source (DE,CRM,MKT,FIN,HR)
GeocodedAddress.E	Preferred Source (DE,CRM,MKT,FIN,HR)
GeocodedAddress.E	Preferred Source (DE,CRM,MKT,FIN,HR)
GeocodedAddress.C	Preferred Source (DE,CRM,MKT,FIN,HR)
GeocodedAddress.C	Preferred Source (DE,CRM,MKT,FIN,HR)
GeocodedAddress.C	Preferred Source (DE,CRM,MKT,FIN,HR)
GeocodedAddress.C	Preferred Source (DE,CRM,MKT,FIN,HR)
GeocodedAddress.L	Preferred Source (DE,CRM,MKT,FIN,HR)
GeocodedAddress.F	Preferred Source (DE,CRM,MKT,FIN,HR)

You have created a consolidation rule that arbitrates how golden data is created out of the source data when duplicates are detected. In this example, *CustomerName* and *AccountManager* will take the most frequent value from the duplicated records; *Address* information will preferably come from the Data Entry application, then – in order of priority – from the CRM, Marketing, Financial and HR applications. Finally, the *TotalRevenue* will be the largest value from all the duplicates.

As a reference was created on the *Contact* entity to the newly created *Customer*, the *Contact* entity now includes a new field called *Customer*.

We must check that this field is consolidated as expected.

Update the *Contact* Consolidator:

1. Expand the **Entities > Contact > Consolidator** Node.
2. Double-click the **Consolidator – Contact** node. The **Consolidator – Contact** editor appears.

3. In the **Field Level Consolidators** group, click the



**Refresh** button to refresh the attributes list and then double-click the *Customer* attribute.

4. In the **Define Selected Field-Level Consolidator** wizard, select **Preferred Source** and then click **Next**.
5. In the **Publisher Ranking** table, click the **Add Publisher Ranking** button.
6. Double-click the *DataEntry* in the **Available Publishers** list to add to the **Publishers** list.
7. Repeat this operation to add *CRM*, *Marketing*, *Finance* and then *HumanResources* in this order.
8. Click **Finish** and then **Finish** again to return to the **Consolidator – Contact** editor.
9. Press **CTRL+S** to save this editor and then close it.

**Congratulations!** You have defined all the data quality, enrichment and match/merge rules for your *Customer* entity. You have also modified existing entities: A new phone enricher for the *Employees*, and an additional attribute consolidated for the *Contacts*.

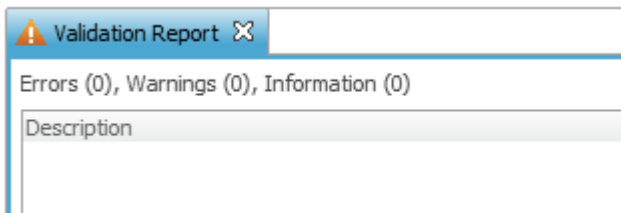
## Validating the Model

Convergence for MDM detects errors or missing elements in the model. For example, entities with no consolidation strategies will be detected.

Validating a model guides you in the process of designing a model. You can perform regular validations to assess how complete the model really is, and you must validate the model before deploying it.

To validate the model:

1. Close all open editors by right-clicking on any of them and selecting **Close All**.
2. Select the model node **CustomerAndFinancialMDM [0.0]** at the root of the tree.
3. Right-click and select **Validate**.
4. The validation process starts. At the end of the process, the list of issues is displayed in the **Validation Report** view.




**Note:** At this stage in the tutorial, no error or warning should show up in the validation report.

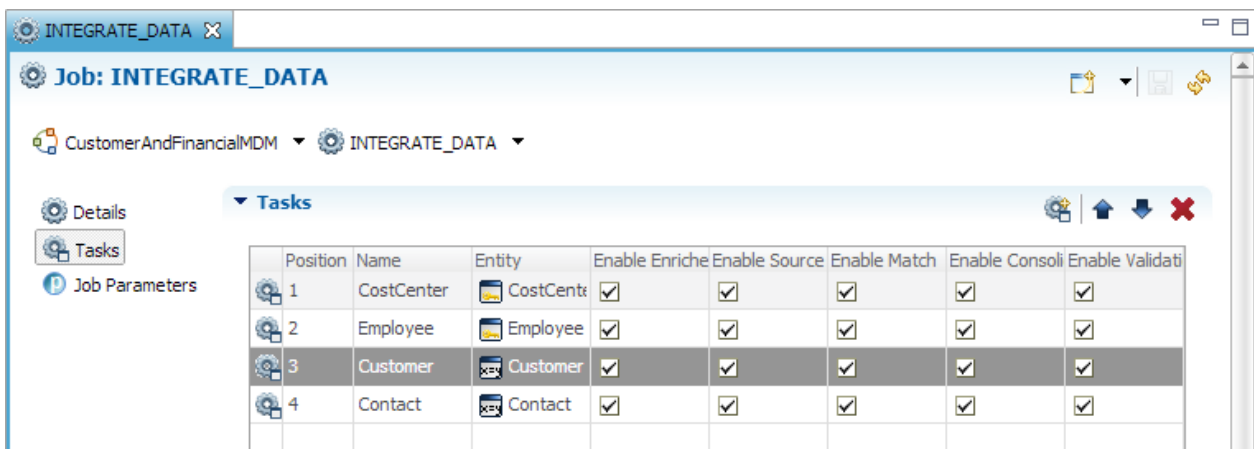
## Working with Integration Jobs


When data is loaded in the hub by an ETL process, or when a user modifies data in the hub as part of a human workflow, Convergence for MDM runs an *Integration Job* to enrich, validate, match/de-duplicate and consolidate this staged data and create golden records. This job uses the enricher, constraints, validations, matchers and consolidators defined in the model.



Integration jobs are already created in the model to certify the master records for the *CostCenter*, *Employee* and *Contacts* entities. As the *Customer* entity was added in the tutorial, it must be added to some of these jobs as a new task.

1. Expand **CustomerAndFinancialMDM [0.0] > Jobs > INTEGRATE\_DATA** in the **Model Edition** view.
2. Double-click the **Tasks** node. The **Job: INTEGRATE\_DATA** editor opens.
3. Click the  **Add Task** button. A **Create New Task** wizard opens.
4. In the **Entity** drop-down box, select *Customer [Entity]*
5. Make sure that all check boxes are checked.
6. Click **Finish** to close the wizard. The *Customer* task is added at the end of the task list.
7. Select this task in the list, and use the **move up** button to move it before the *Contact* task.
8. Press **CTRL+S** to save this editor and then close it.



1. Expand **CustomerAndFinancialMDM [0.0] > Jobs > DE\_CUSTOMER\_CONTACT** in the **Model Edition** view.
2. Double-click the **Tasks** node. The **Job: DE\_CUSTOMER\_CONTACT** editor opens.
3. Click the  **Add Task** button. A **Create New Task** wizard opens.
4. In the **Entity** drop-down box, select *Customer [Entity]*
5. Make sure that all check boxes are checked.
6. Click **Finish** to close the wizard. The *Customer* task is added at the end of the task list.
7. Select this task in the list, and use the **move up** button to move it before the *Contact* task.
8. Press **CTRL+S** to save this editor and then close it.

Ordering tasks in jobs is important as validations on certain entities require that other entities are loaded previously. In this example, *Contact* has a reference to *Customer* that mandates loading customers golden records before contacts.

Congratulations! You have successfully finished your first model! You can now proceed and deploy it to production!

# Deploying the MDM Hub

In this chapter, you will deploy your hub to a data location and make it ready for execution.

## Creating a Data Location

A *Data Location* is an Oracle schema into which several *Model Editions* and *Data Editions* will be deployed.

A *Model Edition* is a Convergence for MDM model deployed at a given time. An MDM Hub model evolves over time, for example to include new entities or functional areas. Model Editions reflect this evolution in the **structure** of the MDM Hub.

Similarly, a *Data Edition* reflects the evolution of the data stored in the hub over time. You can perform snapshots of the master data at a given time to perform for example a what-if analysis. Data Editions reflect the evolution in the **content** of the MDM Hub.

To create a new data location:

1. In the menu, select **File > New > New Data Location**.
2. In the **Create New Data Location** wizard, enter the following values:
  - **Name:** *CustomerAndFinancialMDM*
  - **JNDI Datasource Name:** *java:comp/env/jdbc/SEMARCHY\_DEMO\_MDM*
  - **Description:** *MDM Hub Data Location*
  - **Location Type:** *Development Location*
  - **Root Model:** *CustomerAndFinancialMDM [Root Model]*
3. Click **Finish** to close the wizard. The **Data Editions** view opens.

**Create New Data Location**

**Data Location**  
Data location bound to a JEE datasource to deploy model editions.

Auto Fill: ☒

Name: CustomerAndFinancialMDM

Label: Customer And Financial MDM

JNDI Data Source Name: java:comp/env/jdbc/SEMARCHY\_DEMO\_MDM

Description: MDM Hub Data Location

Location Type: Development Location

Root Model: CustomerAndFinancialMDM [Root Model]

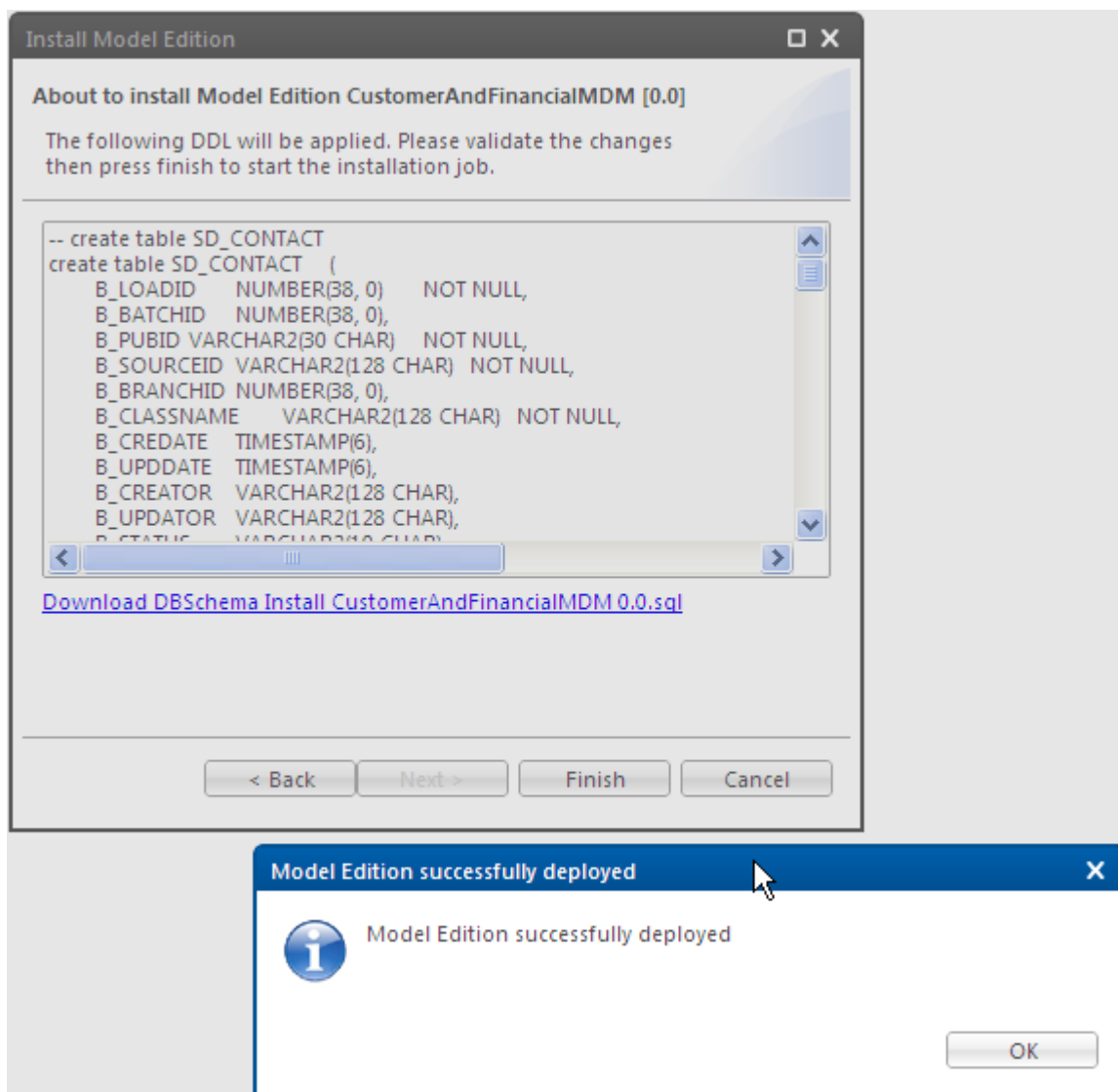
Finish Cancel

## Deploying a Model Edition

Now that the data location is created for the *CustomerAndFinancialMDM* model, it is possible to deploy the first edition of this model.

To install the model edition:

1. In the **Data Editions** view, select the *CustomerAndFinancialMDM* data location, right-click and select **Install Model Edition**. The **Install Model Edition** wizard opens.
2. In **Model Edition**, select *CustomerAndFinancialMDM [0.0] [Model Edition]*
3. Click **Next**. Convergence for MDM generates the SQL code to create the physical database objects corresponding to this model edition. The code is displayed in the wizard and you can download this code to optimize it manually if you wish.
4. Click **Finish** to run this code and deploy the model edition in the data location.
5. Once this operation is complete a **Model Edition Successfully Deployed** dialog appears. Click **OK** to close this dialog.



In the future, when we perform changes to this model's structure, we will be able to create new editions of this model (For example [0.1], [0.2], etc.) and will be able to deploy these new editions in the same data location. These new deployments will generate patching scripts to bring the physical database objects to the level of the new model edition.

## Creating a Root Data Edition

To create a data edition:

1. In the **Data Editions** view, select the *CustomerAndFinancialMDM* data location, right-click and select **Create Root Data Edition**. The **Create New Root Data Edition** wizard opens.

2. Click **Next**.
3. In the second wizard screen, for **Deployed Model Edition**, select *CustomerAndFinancialMDM 0.0 [Deployed Model Edition]*.
4. Click **Finish** to create the first data edition.

This data edition is open. It means that we can perform loads and changes to this edition. Later on, we will be able to close this edition and open a new one. Closing an edition freezes its content, creating a snapshot of the MDM Hub content.

Congratulations! You have successfully deployed your model! You can now proceed and start publishing data!

## Running the MDM Hub

In this chapter, you will publish data from various sources in the MDM hub and review the execution of the integration job.

### Publish Data to the MDM Hub

**Note:** For this tutorial, we use a demonstration application that accesses sample data stored in the SEMARCHY\_DEMO\_SOURCE schema. The tutorial also includes demonstration ETL scripts to load the MDM staging tables from the sample data.

**Tip: Semarchy Convergence for Data Integration** is a high performance data integration platform which can be used in conjunction with Convergence for MDM to publish source data into the MDM hub and consume golden data from the hub. If you want to discover Convergence for DI, a tutorial available on [www.semarchy.com/download](http://www.semarchy.com/download) can drive you through the configuration of Semarchy Convergence for MDM and DI, and guide you for designing a data integration project integrating the MDM hub with a variety of information sources, including XML, file and databases.

To load sample data into the MDM Hub:

1. In the menu, select **Help > Getting Started > Open Demo Application....** If you run the demo application for the first time, click **OK** to install the sample data when prompted. The **Demo Applications** editor opens.
  - The demonstration application contains several editors, each corresponding to an application publishing to the MDM hub. These applications appear as separate tabs at the bottom of the editor.
  - You can edit or delete sample records from these tabs.
  - The **Reset Sample Data** button allows you to restore the sample data to its original state.
  - The **Publish Data to MDM...** button allows you to run the ETL script loading the sample data into the MDM staging table.
2. Click the **Publish Data to MDM...** button in the editor toolbar.
3. In the **Publish Data to MDM** dialog, for **Data Edition** select *CustomerAndFinancialMDM [0.0]*.
4. Click **Finish** to start the loading process.
5. A **Data Submit Succeeded** dialog appears. Click **OK** to close it.

Within 10 seconds, the published data will be detected by the Semarchy Convergence for MDM and integrated by the *INTEGRATE DATA* job.

### Viewing the Log

The execution of the integration job can be monitored in the log.

To review the job log:

1. In the **Data Editions** view, expand **CustomerAndFinancialMDM > Deployed Model Editions > CustomerAndFinancialMDM 0.0 > Integration Jobs > INTEGRATE\_DATA**.
2. Double-click the **Latest Logs** node under the **INTEGRATE\_DATA** job. The list of executed jobs opens.

3. In the **Latest Logs** table, double-click the latest job log (according to the **Start Date**). The **Job Log** editor opens.
4. You can review in this editor the statistics for the job and number of processed rows.
5. Scroll down in the editor to see the list of tasks for this job.

**Job Log: INTEGRATE\_DATA**

INTEGRATE\_DATA ▾

Details

Job Log Parameters

**Details**

Name: INTEGRATE\_DATA

Job Definition: INTEGRATE\_DATA

Status: Done

Start Date: Sep 25, 2013 11:46:40 AM CEST

End Date: Sep 25, 2013 11:46:59 AM CEST

Duration: 0 00:00:19 126.000000

Message:

Row Count: 3288

Insert count: 24

Merge count: 1544

Update count: 925

Error count: 3

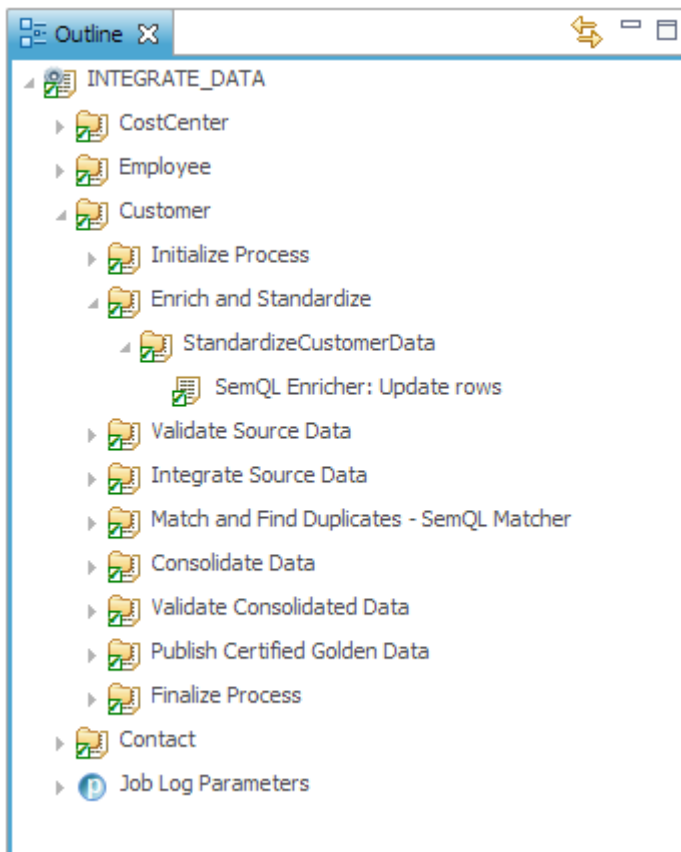
Delete count: 0

**Tasks**

Name	Status	Start Date	End Date	Duration	Message	Row Count	Insert count	Merge count	Update count	Error count
CostCent	Done	Wed Sep	Wed Sep	0 00:00:00		110		81	0	0
Employee	Done	Wed Sep	Wed Sep	0 00:00:00		641		318	214	1
Customer	Done	Wed Sep	Wed Sep	0 00:00:00		696	21	339	96	0
Contact	Done	Wed Sep	Wed Sep	0 00:00:00		1841	3	806	615	2

6. In the list of **Tasks**, double-click the *Customer* task. The **Task Group Log: Customer** editor opens.
7. Scroll down to see the **Task Log** table. These tasks represent the successive process the submitted *Customer* has gone through to build golden records.
8. You can also use the **Outline** view to browse a job log. In the **Outline**, expand the **INTEGRATE\_DATA > Customer > Enrich and Standardize > StandardizeCustomerData** node. It corresponds to your SemQL enricher.





9. Close all editors by select **File > Close All** in the menu.

Congratulations! You have successfully published and certified your first data batch into the hub! So far, we have modeled the data, the rules, deployed the hub and loaded data into it. It is now time to design a friendly interface for the business users and data stewards.

# Creating an Application

**Applications** are designed to provide generated user interfaces from which:

- Business users can browse the master data using dedicated views.
- Data stewards and business users can run human workflows for master data management operations, such as data entry and duplicate management.

An application already exists in the model. This application must be modified to support the new *Customer* entity.

## Creating Table Views and Form Views

We will start with the creation of the **Views** and **Business Objects** required to display the content of the hub.

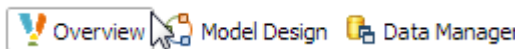
Forms Views and Table Views define the layout of the entities when they are displayed in the application in lists or forms.

### Table Views

First, a new view must be created to define how the new *Customer* entity is displayed in a table layout.

To create a table view for the *Customer* entity:

1. In the top-right toolbar, select the **Overview** perspective.



2. In the **Overview** screen, select the **CustomerAndFinancialMDM [0.0]** link. The **Model Design** perspective opens.



3. In the **Model Edition** view, expand the **Entities > Customer** node.
4. Right-click the **Table Views** node and then select **Add Table View**. The **Create New Table View** wizard opens.
5. In the **Create New Table View** wizard, enter the following values:
  - **Name:** *DefaultTableView*
6. Click **Finish** to close the wizard. The **Table View** editor opens.
7. In the **Table View** editor, in the **Attributes** table, click the



**Add Table Attributes** button. The **Create New Table Attribute** wizard opens.

8. In the **Attributes** list, select the following attributes: *CustomerID*, *Gold\_CustomerID*, *CustomerName*, *FDN\_AccountManager*, *TotalRevenue*, *InputAddress*, *GeocodedAddress*, *CreationDate*, *Creator*, *UpdateDate* and *Updater*. Press and hold the **CTRL** key to perform a multiple selection.
9. Click the **Add >>** button. The selected attributes are added to the **Displayed Attributes** list.
10. Click **Finish**. The attributes are added in the table view **Attributes** list.

**Table View: DefaultTableView**

CustomerAndFinancialMDM ▾ Customer ▾ DefaultTableView ▾

**Details**

**Name and Definition**

Name: DefaultTableView

Default Table View: ☒

Description:


**Attributes**

Name	SemQL Expression	Use Custom Label	Label
CustomerID	CustomerID	<input type="checkbox"/>	
Gold_CustomerID	Gold_CustomerID	<input type="checkbox"/>	
CustomerName	CustomerName	<input type="checkbox"/>	
FDN_AccountManager	FDN_AccountManager	<input type="checkbox"/>	
TotalRevenue	TotalRevenue	<input type="checkbox"/>	
InputAddress	InputAddress	<input type="checkbox"/>	
GeocodedAddress	GeocodedAddress	<input type="checkbox"/>	

11. Press **CTRL+S** to save the table view and then close the editor.

The *Contact* entity already has a table view into which it is necessary to add the *Customer* this contact belongs to.

To modify the table view for the *Contact* entity:

1. In the **Model Edition** view, expand the **Entities > Contact > Table Views** node.
2. Right-click the **DefaultTableView** node and then select  **Add Table Attributes**. The **Create New Table Attribute** wizard opens.
3. In the **Attributes** list, double-click the *FDN\_Customer* attribute.
4. Click **Finish**. The **Table View: DefaultTableView** for the *Contact* entity opens.
5. The *FDN\_Customer* is added at the end of the **Attributes** table.
6. Select this attribute in the **Attributes** table and then use the **Move Up** button to move it under the *Phone2* attribute in the list.


Name	SemQL Expression	Use Custom Label	Label
IsInfluencer	IsInfluencer	<input type="checkbox"/>	
EmailAddress	EmailAddress	<input type="checkbox"/>	
Phone1	Phone1	<input type="checkbox"/>	
Phone2	Phone2	<input type="checkbox"/>	
<b>FDN_Customer</b>	<b>FDN_Customer</b>	<input type="checkbox"/>	
CreationDate	CreationDate	<input type="checkbox"/>	
Creator	Creator	<input type="checkbox"/>	
UpdateDate	UpdateDate	<input type="checkbox"/>	

7. Press **CTRL+S** to save the editor and then close it.

## Form Views

Similar changes must be performed on the existing *Contact* form views to extend them with attributes from the new *Customer* entity.

First, the *DefaultFormView* for the *Contact* entity must be extended to include the customer information. This includes the display name for the related customer (FDN\_Customer), as well as the postal code, city and country of this customer.

1. In the **Model Edition** view, expand the **Entities > Contact > Form Views** node.
2. Double-click the **DefaultFormView** node. The **Form View: DefaultFormView** editor opens. This form view is composed of several sections.
3. In the **Sections** table, double-click the *CustomerDetails* section. The editor changes to the **Form Section: CustomerDetails** editor. This editor shows the list of attributes displayed in this section.
4. In the **Attributes** table, click the  **Add Form Attributes** button. The **Create New Form Attribute** wizard opens.
5. In the **Attributes** list, double-click the *FDN\_Customer* attribute. It is added to the **Displayed Attributes** list.
6. In the **Attributes** list, expand the **Customer > InputAddress** node.

- Under this node, double-click the following attributes to add them to the **Displayed Attributes** list: *PostalCode*, *City* and *Country*.
- Click **Finish**. The attributes are added in the table view **Attributes** list in the **Form Section: CustomerDetails** editor.

**Form Section: CustomerDetails**

CustomerAndFinancialMDM > Contact > DefaultFormView > CustomerDetails

**Name and Definition**

Auto Fill: ☐

Name: CustomerDetails

Label: Customer Details

Layout: 1 Column Layout


Position: 1

**Attributes**

Name	Expandable	SemQL Expression	Expanded by Default	Use Custom Label	Label
IsInfluencer	<input type="checkbox"/>	IsInfluencer	<input type="checkbox"/>	<input type="checkbox"/>	
FDN_Customer	<input checked="" type="checkbox"/>	FDN_Customer	<input type="checkbox"/>	<input type="checkbox"/>	
PostalCode	<input type="checkbox"/>	Customer.InputA	<input type="checkbox"/>	<input type="checkbox"/>	
City	<input type="checkbox"/>	Customer.InputA	<input type="checkbox"/>	<input type="checkbox"/>	
Country	<input type="checkbox"/>	Customer.InputA	<input type="checkbox"/>	<input type="checkbox"/>	

- Press **CTRL+S** to save the form view and then close the editor.

The form view used for data entry in the *Contact* entity must also be modified to include a field to select the customer a given contact is attached to.

- In the **Model Edition** view, expand the **Entities > Contact > Form Views** node.
- Double-click the **DataEntryFormView** node. The **Form View: DataEntryFormView** editor opens. This form view is composed of several sections.
- In the **Sections** table, double-click the *ContactDetails* section. The editor changes to the **Form Section: ContactDetails** editor. This editor shows the list of attributes displayed in this section.
- In the **Attributes** table, click the  **Add Form Attributes** button. The **Create New Form Attribute** wizard opens.
- In the **Attributes** list, double-click the *FDN\_Customer* attribute. It is added to the **Displayed Attributes** list.
- Click **Finish**.
- Press **CTRL+S** to save the form view and then close the editor.

In addition to these changes, form views must be created for the *Customer* entity:

- A *DefaultFormView* used for general purposes (for viewing the Customer in a form)
- A *DataEntryFormView* for entering customer data. The latter has some required fields such as the *SourceID*.

To create the *DefaultFormView* form view:

1. In the **Model Edition** view, select the **Entities > Customer > Form Views**.
2. Right-click and select **Add Form View**. The **Create New Form View** wizard opens.
3. In the **Name** field, enter *DefaultFormView*.
4. In the **Layout** field, select *2 Column Layout*.
5. Click **Finish**. The **Form View: DefaultFormView** editor opens.
6. In the **Sections** table, click the



**Add Form Section** button. The **Create New Form Section** wizard opens.

7. In the **Name**, enter *CustomerDetails*.
8. Click **Finish**. The *CustomerDetails* section is added to the **Sections** table.
9. In the **Sections** table, click the



**Add Form Section** button. The **Create New Form Section** wizard opens.

10. In the **Name**, enter *RecordProperties*.
11. Click **Finish**. The *RecordProperties* section is added to the **Sections** table.

Name	Label	Layout
CustomerDetails	Customer Details	1 Column Layout
RecordProperties	Record Properties	1 Column Layout

12. In the **Sections** table, double-click the *CustomerDetails* section. The editor changes to the **Form Section: CustomerDetails** editor.

13. In the **Attributes** table, click the



**Add Form Attributes** button. The **Create New Form Attribute** wizard opens.

14. In the **Attributes** list, double-click the following attributes to add them to the **Displayed Attributes** list: *CustomerID*, *Gold\_CustomerID*, *CustomerName*, *TotalRevenue*, *FDN\_AccountManager*, *InputAddress* and *GeocodedAddress*.

15. Click **Finish** to return to the **Form Section: CustomerDetails** editor.

Name	Expandable	SemQL Expression	Expanded by Default	Use Custom Label	Label
CustomerID	<input type="checkbox"/>	CustomerID	<input type="checkbox"/>	<input type="checkbox"/>	
Gold_CustomerID	<input type="checkbox"/>	Gold_CustomerID	<input type="checkbox"/>	<input type="checkbox"/>	
CustomerName	<input type="checkbox"/>	CustomerName	<input type="checkbox"/>	<input type="checkbox"/>	
TotalRevenue	<input type="checkbox"/>	TotalRevenue	<input type="checkbox"/>	<input type="checkbox"/>	
FDN_AccountMan	<input checked="" type="checkbox"/>	FDN_AccountMan	<input type="checkbox"/>	<input type="checkbox"/>	
InputAddress	<input checked="" type="checkbox"/>	InputAddress	<input type="checkbox"/>	<input type="checkbox"/>	
GeocodedAddress	<input checked="" type="checkbox"/>	GeocodedAddress	<input type="checkbox"/>	<input type="checkbox"/>	

16. In the editor breadcrumb, select the **DefaultFormView** button to return to the **Form View: DefaultFormView** editor.

17. In the **Sections** table, double-click the *RecordProperties* section. The editor changes to the **Form Section: RecordProperties** editor.

18. In the **Attributes** table, click the



**Add Form Attributes** button. The **Create New Form Attribute** wizard opens.

19. In the **Attributes** list, double-click the following attributes to add them to the **Displayed Attributes** list: *CreationDate*, *Creator*, *UpdateDate*, *Updator* and *BatchID*.

20. Click **Finish**.

Name	Expandable	SemQL Expression	Expanded by Defa	Use Custom Label	Label
CreationDate	<input type="checkbox"/>	CreationDate	<input type="checkbox"/>	<input type="checkbox"/>	
Creator	<input type="checkbox"/>	Creator	<input type="checkbox"/>	<input type="checkbox"/>	
UpdateDate	<input type="checkbox"/>	UpdateDate	<input type="checkbox"/>	<input type="checkbox"/>	
Updator	<input type="checkbox"/>	Updator	<input type="checkbox"/>	<input type="checkbox"/>	
BatchID	<input type="checkbox"/>	BatchID	<input type="checkbox"/>	<input type="checkbox"/>	

21. Press **CTRL+S** to save the form view and then close the editor.

To create the *DataEntryFormView* form view:

1. In the **Model Edition** view, select the **Entities > Customer > Form Views**.
2. Right-click and select **Add Form View**. The **Create New Form View** wizard opens.
3. In the **Name** field, enter *DataEntryFormView*.
4. Click **Finish**. The **Form View: DataEntryFormView** editor opens.
5. In the **Sections** table, click the



**Add Form Section** button. The **Create New Form Section** wizard opens.

6. In the **Name**, enter *CustomerDetails*.
7. In the **Layout** field, select *2 Column Layout*.
8. Click **Finish**. The *CustomerDetails* section is added to the **Sections** table.
9. In the **Sections** table, double-click the *CustomerDetails* section. The editor changes to the **Form Section: CustomerDetails** editor.



10. In the **Attributes** table, click the



**Add Form Attributes** button. The **Create New Form Attribute** wizard opens.

11. In the **Attributes** list, double-click the following attributes to add them to the **Displayed Attributes** list:

*SourceID, CustomerName, TotalRevenue, FDN\_AccountManager* and *InputAddress*.

12. Click **Finish** to return to the **Form Section: CustomerDetails** editor.

13. Press **CTRL+S** to save the form view and then close the editor.

## Creating Business Objects and Business Object Views

A **Business Object** assembles several entities using their relations to create an object that talks to the business. For example, the “Hierarchy of Employees” or the “Customers with their Contacts” are business objects.

In an application, **Business Object Views** are created and display Business Objects with specific table and form views.

In this section, we will create:

- Two business objects representing the customers and their contacts (*CustomerBO*) and the customers by account manager (*CustomersByEmployeeBO*).
- Business object view to display these business objects and to edit the customers and contacts.

### Creating Business Objects

The first Business Object (*CustomerBO*) groups customers with their contacts through the *ContactBelongsToCustomer* relation.

1. In the **Model Edition** view, right-click the



**Objects** node and select **Add Business Object....** The **Create New Business Object** wizard opens.

- In the **Entity** select *Customer [Entity]*
- Click **Next**

2. Enter the following values:

- **Name:** *CustomerBO*
- **Label:** *Customer*
- **Plural Label:** *Customers*

3. Click **Finish** to close the wizard. The **Business Object: CustomerBO** editor opens.

4. In the editor scroll down to the **Transitions** table.

5. Click the



**Add Transition** button. The **Create New Business Object Transition** wizard opens.

6. In the **Reference**, select *ContactBelongsToCustomer [Reference Relationship]*

7. Click **Next**

8. Select **New Entity Object** and then click **Next**.

9. Click **Finish**. The transition is added to the **Transitions** table.

**Business**

CustomerBO

Objet Métier: CustomerBO

CustomerAndFinancialMDM CustomerBO

Détails

**Nom et Définition**

Objet Entité Racine: CustomerEO

Saisie Auto: ☐

Nom: CustomerBO

Libellé: Customer

Pluriel: Customers

Description:

**Définition de Filtre**

Filtre Racine:

**Transitions**

Nom
Contacts
ContactEO

10. Press **CTRL+S** to save the editor and then close it.

Another business object *CustomersByEmployeeBO* groups the account managers with their managed customers. Account managers are in the cost centers named 'Executive', 'Shipping' and 'Sales'.

1. In the **Model Edition** view, right-click the



**Objects** node and select **Add Business Object....** The **Create New Business Object** wizard opens.

- In the **Entity** select *Employee [Entity]*
- Click **Next**

2. Enter the following values:

- **Name:** *CustomersByEmployeeBO*
- **Label:** *Customers By Account Manager*

3. Click **Next**.

4. In the **Root Filter**, enter `CostCenter.CostCenterName in ('Executive', 'Shipping', 'Sales')`

5. Click **Finish** to close the wizard. The **Business Object: CustomersByEmployeeBO** editor opens.

6. In the editor scroll down to the **Transitions** table.

**Business**

7. Click the



**Add Transition** button. The **Create New Business Object Transition** wizard opens.

8. In the **Reference**, select *CustomerHasAccountManager [Reference Relationship]*

9. Click **Next**

10. Select **New Entity Object** and then click **Next**.

11. Click **Finish**. The transition is added to the **Transitions** table.

CustomersByEmployeeBO

Business Object: CustomersByEmployeeBO

CustomerAndFinancialMDM CustomersByEmployeeBO

Details

**Name and Definition**

Root Entity Object: EmployeeEO

Auto Fill: ☐

Name: CustomersByEmployeeBO

Label: Customers By Account Manager

Plural Label: Customers By Account Manager

Description:

**Filter Definition**

Root Filter: CostCenter.CostCenterName in ('Executive', 'Shipping', 'Sales')

**Transitions**

Name
Customers
CustomerEO

12. Press **CTRL+S** to save the editor and then close it.

## Creating Business Object Views

The newly created business objects will be used in the *DemoApplication* that exists in the model.

To use them, several Business Object Views must be created to define how these business objects will display. Business object views associate form and tables views to business objects.

The *CustomersByEmployeeView* is a view to display the *CustomerByEmployeeBO* business object: The employees and their managed customers.

1. In the **Model Edition** view, expand the **Applications > DemoApplication > Folders** node.
2. Select the **CustomersAndContacts** folder, right-click and select **Add Business Object View...**. The **Create New Business Object View** wizard opens.

3. In the **Create New Business Object View** wizard, enter the following values:
  - **Name:** *CustomersByEmployeeView*
  - **Label:** *Customers By Account Manager*
4. Click **Next**
5. In the **Business Object**, select *CustomerByEmployeeBO [Business Object]*.
6. Click **Finish** to close the wizard. The **Business Object View** editor opens.
7. Scroll down to the **Business Object View Entities** table, and expand the nodes to review the Form/ Table Views used for the various entities involved in the business object that this view displays.

Entity Object / Transition	Form / Table View
EmployeeEO	DefaultFormView
Customers	DefaultTableView
CustomerEO	DefaultFormView

8. Press **CTRL+S** to save the editor and then close it.

The *CustomersView* is a view used to display the *CustomerBO* business object: The customers and their attached contacts.

1. In the **Model Edition** view, expand the **Applications > DemoApplication > Folders** node.
2. Select the **CustomersAndContacts** folder, right-click and select **Add Business Object View...**. The **Create New Business Object View** wizard opens.
3. In the **Create New Business Object View** wizard, enter the following values:
  - **Name:** *CustomersView*
  - **Label:** *All Customers*

4. Click **Next**
5. In the **Business Object**, select *CustomerBO* [Business Object].
6. Click **Finish** to close the wizard. The **Business Object View** editor opens.
7. Scroll down to the **Business Object View Entities** table, and expand the nodes to review the Form/ Table Views used for the various entities involved in the business object that this view displays.
8. For the **Contacts** transition, select *CompactTableView* [TableView]. This view shows a simplified list of the contacts and is more convenient in this case.

Entity Object / Transition	Form / Table View
CustomerEO	DefaultFormView
Contacts	CompactTableView
ContactEO	DefaultFormView

9. Press **CTRL+S** to save the editor and then close it.

Usually, different business object views are used for viewing and authoring the golden data in human workflows. Views for authoring the golden data include more information (for example, the *SourceID* field or the fields marked as mandatory, etc.).

The *CustomersDataEntryView* view is created for authoring customers.

1. In the **Model Edition** view, expand the **Applications > DemoApplication > Business Object Views** node.
2. Select the **Business Object Views** node, right-click and select **Add Business Object View....** The **Create New Business Object View** wizard opens.

3. In the **Create New Business Object View** wizard, enter the following values:
  - **Name:** *CustomersDataEntryView*
4. De-select the **Visible** option. This hides the view from the application. It will be available only for the purpose of human workflows.
5. Click **Next**
6. In the **Business Object**, select *CustomerBO [Business Object]*.
7. Click **Finish** to close the wizard. The **Business Object View** editor opens.
8. Scroll down to the **Business Object View Entities** table, and expand the nodes to review the Form/ Table Views used for the various entities involved in the business object that this view displays.
9. For the **CustomerEO** line, select *DataEntryFormView [Form View]*. This form view is suitable for editing customers.
10. For the **Contacts** line, select *CompactTableView [Table View]*.
11. For the **ContactEO** line, select *DataEntryFormView [Form View]*.

Entity Object / Transition	Form / Table View
CustomerEO	DataEntryFormView
Contacts	CompactTableView
ContactEO	DataEntryFormView

12. Press **CTRL+S** to save the editor and then close it.

Good Job! You have designed the various elements for viewing (and editing) the data for the newly created *Customer* entity.

## Creating Human Workflows

**Human Workflows** enable business users to manage the data in the MDM hub via an application. In this section, we will configure two workflows to enter customers/contacts and to manage customer duplicates.

When users want to manage the master data, they initiate an **Activity** based on a human workflow. This activity follows the workflow through **Transitions** and **Tasks** which are assigned to roles, claimed, processed and then completed by users. The last task of a workflow can submit (or cancel) the data changes done in the activity, and start a data certification process with these changes.

There are two types of human workflows in Semarchy Convergence for MDM:

- **Data Entry Workflows:** These workflows allow data stewards or business users to contribute to the hub as manual publishers. The data entered via these workflows goes through the data certification process to create golden data.
- **Duplicate Management Workflows:** These workflows allow data stewards to override the decisions taken by the matchers running in the hub. Through these workflows, stewards can either manually match unmatched records, or split duplicate groups that were incorrectly matched (false matches).

### Creating a Data Entry Workflow

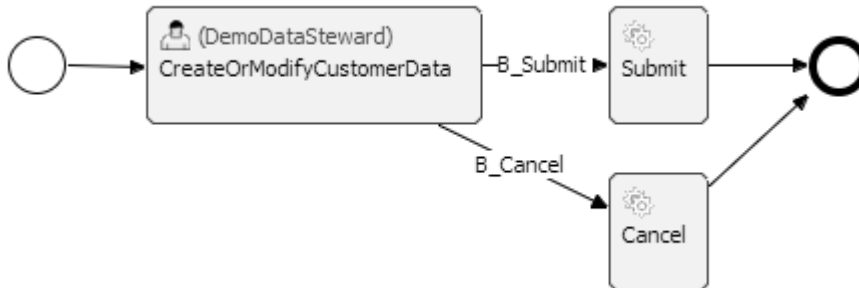
Authoring data for the new *Customer* entity will be performed through a new data entry workflow called *CustomersCreationProcess*.

This single-task workflow involves only users with the *DemoDataSteward* role.

1. In the **Model Edition** view, expand the **Applications > DemoApplication > Human Workflows** node.
2. Right-click the **Human Workflows** node and select **Add Data Entry Workflow....** The **Create New Data Entry Workflow** wizard opens.
3. In the **Create New Data Entry Workflow** wizard, enter or select the following values:
  - **Name:** *CustomersCreationProcess*
  - **Label:** *Edit Customers*
  - **Data Entry Publisher:** *DataEntry [Publisher]*
  - **On Submit Job:** *DE\_CUSTOMER\_CONTACT [Job]*
  - **Initiator Role:** *DemoDataSteward*
4. Click **Next**. This step shows the various actions allowed in this workflow: *Checkout* actions allow editing existing records or fixing rejects. *Create New Record* allows adding records via the workflow. Leave all the actions selected.
5. Click **Next**.
6. In the **Available BO Views** list, double-click the *CustomersDataEntryView* to add it to the **Selected BO Views**.
7. Click **Finish** to close the wizard. The workflow editor opens on the **Diagram** tab.
8. In the workflow diagram, select the **Add Task** tool in the **Palette**.
9. Click the diagram. The **Create New Task** wizard appears.
10. In the **Create New Task** wizard, enter or select the following values:
  - **Name:** *CreateOrModifyCustomerData*
  - **Label:** *Create / Modify Customer Data*

- **Assigned to Role:** *DemoDataSteward*

11. Click **Finish** to close the wizard. The new task is created, and is connected from the starting point. You must link it to the **Submit** and **Cancel** final tasks.
12. Select the **Add Transition** tool in the palette. Click the **CreateOrModifyCustomerData** task, and while keeping the mouse pressed, move the cursor on top of the **Submit** task. Release the button. A transition is created between these two tasks.
13. Repeat the previous operation to link the **CreateOrModifyCustomerData** task to the **Cancel** task. The workflow diagram appears as shown below.



14. Press **CTRL+S** to save the editor and leave it open.

Data entry workflows support interactivity through enrichers and data quality validations triggered while entering data or moving the workflow to a new task.

We will configure all validations and enrichers to run when the focus leaves the data entry fields, except one email enricher that will be triggered manually. We will also configure validations to prevent submitting data that does not comply with certain validations, and warn for one of them.

Configure the Task:

1. Select the **Properties** view (lower panel of the workbench).
2. Select the **CreateOrModifyCustomerData** task in the workflow.
3. In the **Properties** view, select the **Enrichers** tab. This tab shows the list of enrichers triggered while entering data in this task.
4. Select all the lines in the enrichers list, right-click and then select **Field Exit**.
5. Now select the *StandardizeContactEmail* enricher in the list, right-click and then select **Manual**.
6. Select the **Validations** tab in the **Properties** view.
7. Select all the lines in the validations list, right-click and then select **Field Exit**.
8. Press **CTRL+S** to save the editor and leave it open.

Configure the Transition:

1. In the workflow, select the **B\_Submit** transition that links the *CreateOrModifyCustomerData* task to the *Submit* task.
2. Select the **Enrichers** tab in the **Properties** view.
3. Select all the lines in the enrichers list, right-click and then select **Execute Enricher**.
4. Select the **Validations** tab in the **Properties** view.
5. Select all the lines in the validations list, right-click and then select **Block**.



6. Now select the *Account Manager – Reference validation* in the list, right-click and then select **Warn**.
7. Press **CTRL+S** to save the editor and then close it.

## Creating a Duplicate Management Workflow

We will now create a *CustomerDuplicatesProcess* workflow is created to manage the customer duplicates.

1. In the **Model Edition** view, expand the **Applications > DemoApplication > Human Workflows** node.
2. Right-click the **Human Workflows** node and select **Add Duplicates Management Workflow....** The **Create Duplicates Management Workflow** wizard opens.
3. In the **Create New Duplicates Management Workflow** wizard, enter or select the following values:
  - **Name:** *CustomerDuplicatesProcess*
  - **Label:** *Manage Duplicate Customers*
  - **Managed Entity:** *Customer [Entity]*
  - **On Submit Job:** *DE\_CUSTOMER\_CONTACT [Job]*
  - **Initiator Role:** *DemoDataSteward*
4. Click **Finish** to close the wizard. The workflow editor opens on the **Diagram** tab.
5. In the workflow diagram, select the **Add Task** tool in the **Palette**.
6. Click the diagram. The **Create New Task** wizard appears.
7. In the **Create New Task** wizard, enter or select the following values:
  - **Name:** *VerifyCustomerDups*
  - **Label:** *Verify and Validate Customer Duplicates*
  - **Assigned to Role:** *DemoDataSteward*
8. Click **Finish** to close the wizard. The new task is created, and is connected from the starting point. You must link it to the **Submit** and **Cancel** final tasks.
9. Select the **Add Transition** tool in the palette. Click the **VerifyCustomerDups** task, and while keeping the mouse pressed, move the cursor on top of the **Submit** task. Release the button. A transition is created between these two tasks.
10. Repeat the previous operation to link the **VerifyCustomerDups** task to the **Cancel** task.
11. Press **CTRL+S** to save the editor and then close it.

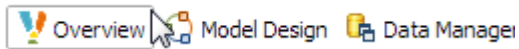
Congratulations! Your first application is ready to go live!

## Using the Application

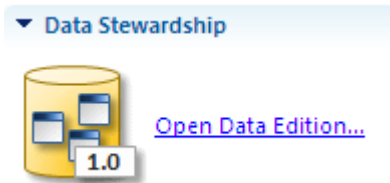
In an **Application**, data stewards and business users can browse the MDM hub content using the **Business Object Views**. They can also author (create or modify) master data in the hub and manage duplicates using the **Human Workflows**.

### Connecting to the Application

1. In the top-right toolbar, select the **Overview** perspective.



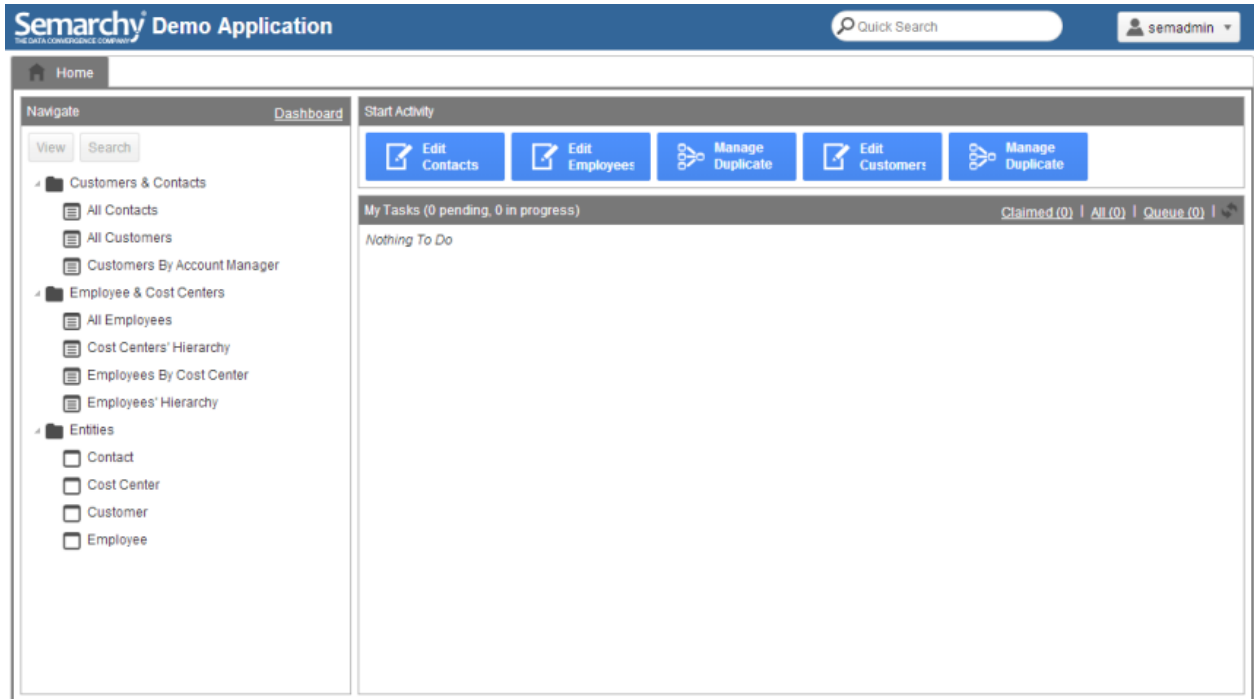
2. In the **Overview** screen, press the CTRL key and then click the **Open Data Edition** link under the **Data Stewardship** group to open it in a new tab.



3. A new tab opens, showing the applications available. Click the **Demo Application CustomerAndFinancialMDM** button to open this application. The application opens.

**Note:** At design-time, the application is generated every time you access it (the startup time is in fact the application generation time). In production, it is generated once.

## The Demo Application



The application **Home** page provides an overview of the hub:

- The **Navigate** view shows the business object views classified in their folders, as well as the list of the entities in the hub. Double-clicking on a business object view opens a new editor showing the data.
- The **My Tasks** view is a summary of the human workflows in progress.
- The **Start Activity** view contains shortcut to start new workflows.

**Note:** This data access is secured by privileges. As the data steward, you can access all the entities, but other users have a more limited access as defined in the model security, which is not covered in this tutorial.

## Navigating the Master Data

In this section, we take the role of a business user who wants to browse the customer data.

### Browsing

We look for a customer record called *GADGETRON*.

1. Double-click the **All Customers** view in the **Customers and Contacts** folder. A new editor opens showing the list of customer data certified in the hub in the **Hierarchy** view and in a table view.

2. In the editor toolbar, click the



**Hide/Show Lineage** button to disable the lineage.

3. Click the **Customer Name** column header on the table to sort records by customer name.
4. Use the



**Next** and



**Previous** buttons to move from page to page until you see *GADGETRON*.

## Quick Search

Another way to access GADGETRON is using a quick search.

1. Click the *Quick Search* field in the application header.
2. Enter *Gad* in this field, and then press ENTER.



A dialog opens, asking you the business object view to search into.

3. Select the **All Customers** business object view and then click **OK**. The **All Customers** editor opens with two records matching the search.

## Advanced Search

Let's try some advanced filtering. We search for "US customers with revenue greater than 10 000", and we will export them.

1. Click the



**Search** button in the editor toolbar to open the search menu, and then select **Search....** The **Define Filter** dialog opens with the current search %Gad% marked as a filter.

2. Select the **Search Using Criteria** option.
3. Click the ... (**Select Attribute**) button.
4. In the **Select Attribute** dialog that opens, select *TotalRevenue* and then click **OK**.
5. Select the "greater than" (>) operator.
6. In the value field, enter 10000.
7. Click the **Add** button.
8. Select the ... (**Select Attribute**) button. In the **Select Attribute** dialog that opens, expand the *InputAddress* complex field, select *Country* under that node and then click **OK**.
9. Click the ... selector button to select the country *USA*.

10. Select the **Save Filter As** and name the filter *Important US Customers*.

Define Filter

Search Using Criteria

Define a list of criteria for your filter

Filter Type

☐ Full-Text Search ☐ Search by Form ☒ Search Using Criteria ☐ Advanced Search Using SemQL

Add Remove

☒ and ☐ or

TotalRevenue ... > v 10000 ...

InputAddress.Country ... = v USA ...

☒ Save Filter As: Important US Customers

Finish Cancel

11. Click **Finish**. The filter is applied and only US customers with total revenue greater than 10,000 appear.

## Export

1. Select **Actions > Export** in the editor toolbar.
2. Select the suitable **Export Format** (Excel or CSV).
3. Click the **Download data-export.xls** link to download the data file and open it with the appropriate editor.
4. Click **Close** to close the dialog.

## Browsing References

1. In the list of customers, use the



**Next** and



**Previous** buttons to move from page to the *GADGETRON* record.

2. Double-click this record in the list. A master/detail page appears. In the upper part, the details of the customer appear as specified in the form view we created during the tutorial. The lower part of the editor shows the child records, including the contacts attached to this customer.
3. In the hierarchy, expand the *GADGETRON > Contacts* node to view the contacts attached to this customer.
4. Double-click the *Britney Bell* contact to open its form.
5. In this form, click the *GADGETRON* link to go back to the parent customer record.

## Authoring Master Data

The same business user now wants to change *GADGETRON*'s total revenue, and create a new customer called *SEMARCHY*.

## Creating and Modifying Records

1. In the *GADGETRON* editor toolbar, select **Actions > Edit**. A new activity automatically starts and an editor opens to edit *GADGETRON* in this activity.
2. Change the **Total Revenue** value to 145000 and then click **Save**. You return to the **Edit Customer** activity editor.
3. Select **Actions > New**. An editor opens to create a new customer.
4. Enter the following values:
  - **Customer Name:** *SEMARCHY*
  - **Total Revenue:** 6946000
  - **Account Manager:** Click the [...] button. The list of employee opens. Double-click *Steven King* in this list. The list closes and this employee is set as the account manager.
5. Expand the **Input Address** field. In the input address fields, enter:
  - **Address:** *750 Menlo Ave #250*
  - **Postal Code:** *CA 94025*
  - **City:** *Menlo Park*
  - **Country:** *USA*
  - Click the **Save** button to save this entry.
6. Click the **Complete Task** button in the editor toolbar to complete this task.
7. In the **Complete Task** dialog, make sure that the **Action to Perform** is set to *Submit* and then click **Finish**. The data is validated and published to the hub.

**Note:** As you enter data in the forms markers indicate the various errors detected. If you move your mouse cursor on these indicators, the detail of the error is displayed in a tooltip. Data quality rules are enforced as part of the workflow, and you cannot submit incorrect data.

## Reviewing the Changes

Perform a quick search for the *SEM* and *GADGE* customers to see the new and updated customer records:

- A new customer called SEMARCHY must appear now.
- GADGETRON must now have a Total Revenue of 145,000.

## Data Stewardship

The data steward is responsible for maintaining the data in the hub. In addition to the navigation and authoring operations, he uses data lineage, duplicate management and rejects detection.

## Using the Lineage

The **Lineage** reflects the process that led to the master records, and allows drilling down into the various stages of this process.

1. Select the **Home** tab.
2. Double-click the **All Customers** view in the **Customers and Contacts** folder. A new editor opens showing the list of customer data certified in the hub.
3. In the editor toolbar, click the



**Hide/Show Lineage** button to enable it.

4. Sort the records by customer name and then use the























**Next** and



**Previous** buttons to move from page to page until you see *GADGETRON*.

5. Expand the node corresponding to a GADGETRON Golden Record in the **Hierarchy**.
6. Expand the **Master Records** node. The list containing the master records that lead to the consolidated golden record appears.

## Hierarchy

- ▶  DUNDER MIFFLIN
- ▶  GADGETRON
  - ▶  Contacts
    - ▶  James Mikkilineni
    - ▶  Alexandre Ates
    - ▶  Laura Tan
    - ▶  Britney Bell
    - ▶  Steven Landry
    - ▶  Ramit Scheer
    - ▶  Sarah Perkins
    - ▶  Irene Nayer
    - ▶  Samuel Everett
  - ▶  Master Records
    - ▶  GADGETRON
    - ▶  GADGETROM
    - ▶  GADGETRON
    - ▶  GADGETRON
    - ▶  GADGE-TRON
- ▶  GENERAL FORGE AND FOUNDRY
- ▶  GENERAL PRODUCTS

7. Double-click the GADGETRON record in the **Hierarchy**. The form for this record opens.

8. Click the



**Show/Hide Hierarchy** to temporarily hide the hierarchy view.

9. Select the **Master Records** list in the **Child Records** section. You can see the same master records.



Home Dashboard All Customers x

**GADGETRON**

Actions 24 (...)

Customer Details

Customer ID: 105

Customer Name: GADGETRON

Total Revenue: 145,000

Account Manager (Name): [Matthew Weiss](#)

Input Address: 711 KAPIOLANI BOULEVARD HONOLULU I

Geocoded Address: 711 96813 Honolulu US

Record Properties

Created On: Sep 23, 2013 6:46:05 PM CEST

Created By: semadmin

Updated On: Sep 23, 2013 7:10:12 PM CEST

Updated By: semadmin

Batch ID: 26

Child Records

Contacts Master Records

1 - 5 (5)

Customer ID	Customer ID (Golden ID)	Customer Name	Account Manager (Name)	Total Revenue	Input Address
CRM.12019	105	GADGETRON	Matthew Weiss	2,558	711 KAPIOLANI
CRM.1251	105	GADGETROM	Matthew Weiss	1,405	711 KAPIOLANI
CRM.X3452	105	GADGETRON	Lex De Haan	14,500	711 KAPIOLANI
DE.ee8bf781-6a05-465d-ac96-1c6608499224	105	GADGETRON	Matthew Weiss	145,000	711 KAPIOLANI
MKT.1260	105	GADGE-TRON	Matthew Weiss	4,770	710 KAPIOLANI

You can review the following:

- The *Customer.ID* field indicates that three records are duplicates in the CRM application, and one comes from the MKT (Marketing) application. These have been detected as duplicate per the SemQL Matcher created in this tutorial. Another record correspond to the one authored in the hub during the tutorial.
- During the consolidation, the *Customer Name* field and *Account Manager* were picked up from the most frequent value as per the consolidator defined in this tutorial.
- The *Customer Name* and the *Input Address* have been transformed to uppercase as per the SemQL Enricher.
- If you have activated the Plug-in Enricher, the *Geocoded Address* is loaded. Otherwise it is empty.

10. Click the



**Show/Hide Hierarchy** to show the hierarchy view.

## Managing Duplicates

When the automated duplicates detection does not provide appropriate results, it is possible to split or merge duplicate groups.

A number of records, published from source applications, have been automatically matched and consolidated into the GADGETRON golden record. One of them was incorrectly merged. In reality, *GADGE-TRON* is in fact a different legal entity than *GADGETRON*. They share the same address, and the matching process interpreted the name and address similarity for a match. The data steward must fix this.

## Un-Merging Duplicates

1. In the **Hierarchy** view, select the *GADGE-TRON* master record, right-click and then select **Move to New Golden**.
2. A new **Manage Duplicate Customers** workflow opens. In the editor, the two records resulting from the un-merge operation appear.

Home

All Customers

Manage Duplicate Customers [29] ×

Manage Duplicate Customers: Verify and Validate Customer Duplicates

Complete Task...

←

→

↺

🔍

🔼🔽

🔄

📄

📄

Actions

(Filtered) 1 - 2 (2)

Customer ID	Is Confirmed	Customer Name	Address	City	Country
2	<input checked="" type="checkbox"/>	GADGETRON	711 KAPIOLANI BOULEVARD	HONOLULU	USA
DE.ee8bf781-6a05-465d-ac96-1c6608499224	N/A	GADGETRON	711 KAPIOLANI BOULEVARD	HONOLULU	USA
CRM.X3452	N/A	GADGETRON	711 KAPIOLANI BLVD	HONOLULU	USA
CRM.1251	N/A	GADGETROM	711 KAPIOLANI BLVD	HONOLULU	USA
CRM.12019	N/A	GADGETRON	711 KAPIOLANI BOULEVARD	HONOLULU	USA
104	<input checked="" type="checkbox"/>	GADGE-TRON	710 KAPIOLANI BV	HONOLULU	USA
MKT.1260	N/A	GADGE-TRON	710 KAPIOLANI BV	HONOLULU	USA

Task Overview

Duplicates Management

3. Click the **Complete Task** button in the editor toolbar to complete this task.
4. In the **Complete Task** dialog, make sure that the **Action to Perform** is set to *Submit* and then click **Finish**. The data is validated and published to the hub.

## Merging Duplicates

We can revert this operation as easily.

1. Make a quick search for **GADGE** for **All Customers**. The GADGETRON and GADGE-TRON records that we have unmerged appear.
2. Select them, right-click and select **Merge**.
3. A new **Manage Duplicate Customers** workflow opens. In the editor, the record resulting from the merge operation is shown.
4. Click the **Complete Task** button in the editor toolbar to complete this task.
5. Click **Finish**.

**Note:** The duplicate management process allows confirming, splitting or merging detected duplicates. Note that the decisions taken by the steward are preserved in future loads and override any automated duplicate detection.

## Using the Home Dashboard

The **Dashboard** provides an overview of the hub contents.

1. Select the **Home** tab.
2. Click the **Dashboard** link in the **Home** page view. The **Dashboard** page opens.

The dashboard displays several sections:

- **Workflow Activities** displays a summary of the activities in progress.
- **Data Certification Jobs** shows the status of the certification jobs in progress, queued or completed.
- **Data Location Statistics** shows the aggregated metrics for all the entities of the MDM hub. If you select one entity in this list (for example: *Contact*), the detailed metrics for this entity are displayed in the **Selected Entity Details** section of the dashboard.

Home

Dashboard

Dashboard

Workflow Activities

New Activity...

Pending:

0

View

Claimed by me:

0

View

Claimed by others:

0

View

All tasks:

3

View

Data Certification Jobs

Open Data Certification Queue...

Latest Batch: 4

Latest Job: DE\_CUSTOMER\_CONTACT

Status	Number of Jobs
Loads/Data Entry in progress	0
Queued Jobs	0
Running Jobs	0
Successful Jobs (No Rejects)	3
Successful Jobs (With Rejects)	1
Jobs in Error	0

Data Location Statistics

Entity	Golden Records	Master Records	Latest Batch Size	Latest Batch Rejects (Source)	Latest Batch Rejects (Post)
<input checked="" type="checkbox"/> Contact	201	212	9	0	0
<input type="checkbox"/> Cost Center	27	27	27	0	0
<input type="checkbox"/> Customer	82	98	2	0	0
<input type="checkbox"/> Employee	106	106	107	1	0

Selected Entity Details

Entity: Contact

Latest Batch: 2

Latest Job: DE\_CUSTOMER\_CONTACT

Certified Data	Golden Records	Master Records
Total Records	201	212
Records With Duplicates	9	20
Pending Validation	9	20

Source Data History	Source Records
Total Records	214
Latest Batch Size	9
Average Batch Size	107

Rejected Data	Source	Post-consolidation
Latest Batch Rejects	0	0
Latest Batch Errors	0	0
Total Errors	2	0
Total Rejects	2	0

## Detecting Rejects

1. In the **Dashboard**, click the *Contact* entity in the **Data Location Statistics**.
2. In the **Rejected Data** table. 2 source rejects exist. We want to review them.
3. Click the **Home** tab.
4. In the **Navigate** view, select **Entities > Contact**.

68

5. Right-click and select **Rejects > Source Errors**. The two errors appear.
6. Click the **Select Displayed Attributes**



button.

7. In the **Displayed Attribute** dialog, select the **ConstraintName** and **ConstraintType** attributes and click the **Add >>** button to add them to the list of displayed attributes.
8. Use the **Move Top** button to move them to the top of the list.
9. Press **Finish**
10. The two errors (Customer - FOREIGN) are now identified as contacts from the CRM and MKT applications, but attached to non-existing customers.

Home

Dashboard

All Customers

Source Errors: Contacts

Source Errors

Navigation icons

Hierarchy

Search

Filter

Sort

Export

1 - 2 (2)

Page navigation

Bruce Hunold

Constraint Name

Constraint Type

Contact ID

First Name

Last Name

Gender

Job Title

Is Influencer

Email

Lex Kochhar

Customer

FOREIGN

CRM.41060

Bruce

Hunold

M - Male

Programmer

☐

bruce

Customer

FOREIGN

MKT.5533

Lex

Kochhar

M - Male

Administration Vice President

☐

lex.ko

These rejects have been detected by the data certification process on the data published from source applications. It is possible to fix them in the source applications and re-publish them to the hub, or fix them directly in the hub using a data entry workflow.

**Congratulations!** As a data steward, you have reviewed duplicates, managed false duplicates, monitored the hub's dashboard and found two errors!

# Summary

## Summary

Congratulations! You have now completed your first MDM project with Semarchy Convergence for MDM.

In this tutorial, you learned how to:

- Install Convergence for MDM,
- Design a model, including customized data types, entities, integration artifacts (enrichers, constraints, matchers and consolidators),
- Deploy the Model in an MDM Hub,
- Load the MDM Hub and automatically run integration jobs,
- Create an Application for business users and data stewards to access and perform changes in the MDM Hub,
- Use this application to orchestrate business process and to navigate through your golden and master data,
- Review the Golden Data certified by Convergence for MDM and drill through the data lineage.

## Going Further with Convergence for MDM

You have learned how to use Convergence for MDM for a typical MDM project. But Convergence for MDM is capable of addressing any type of master data projects and functional areas, including Products, Customer, Parties, Cost Centers, etc.

Convergence for MDM includes the following features to support your MDM initiatives:

- **Comprehensive Logical Modeling** for faster development and optimal reactivity to changes at a lower cost.
- **Data and Metadata Versioning** increasing confidence for data compliance, supporting what if analysis.
- **Non-Intrusive Convergence Hub** allowing applications to seamlessly push data to Convergence for MDM and consume certified data.
- **Generated Integration and Certification Processes** guaranteeing fully certified data.
- **Leverage the Power of Database Engines** for high performance and maximal scalability for master data certification and access.
- **Web 2.0 and Cloud Ready Solution** providing the lowest cost of deployment on the market.

## Learn More

**Note:** The **Semarchy Convergence Documentation Library**, including the development, administration and installation guides is available online at the following URL: <http://semarchy.com/download/>

In addition to the product manuals, Semarchy provides other resources including whitepapers, datasheets, and a complete set of videos demonstration the product features.

The resources are available on the Semarchy Website.

© Copyright Semarchy 2011-2013. All Right Reserved.