TIBCO® MDM Studio

Repository Designer User Guide

Version 6.1.1 | June 2024

Document Updated | October 2024

Contents

Getting Started

This chapter explains how to get started using the Repository Designer.

Repository Designer Overview

Repositories are the cornerstone of TIBCO MDM. All your data and records, both master and reference, are created, stored, and managed in repositories.

The Repository Designer adds a visual element to designing repositories and makes the process quicker and more intuitive.

The Repository Designer is based on TIBCO Business Studio and acts as an 'add on' component to TIBCO Business Studio. The Repository Designer now comes with a new, MDM specific model editor through which you can design and edit your repository model. Repository models are now in a .rep format, contained in a special folder called "Repository Models".

What you can do with the Repository Designer

With the Repository Designer, you can create repositories, modify them, rename them, add and modify attributes and attribute groups, create self or cross repository relationships, and export or deploy the final repository model to TIBCO MDM to use it.

You can also import repository metadata exported from TIBCO MDM into the Repository Designer for further edits.

The Repository Designer also supports the creation of input maps, output maps, and synchronization format and classification schemes.

Starting the Repository Designer

Procedure

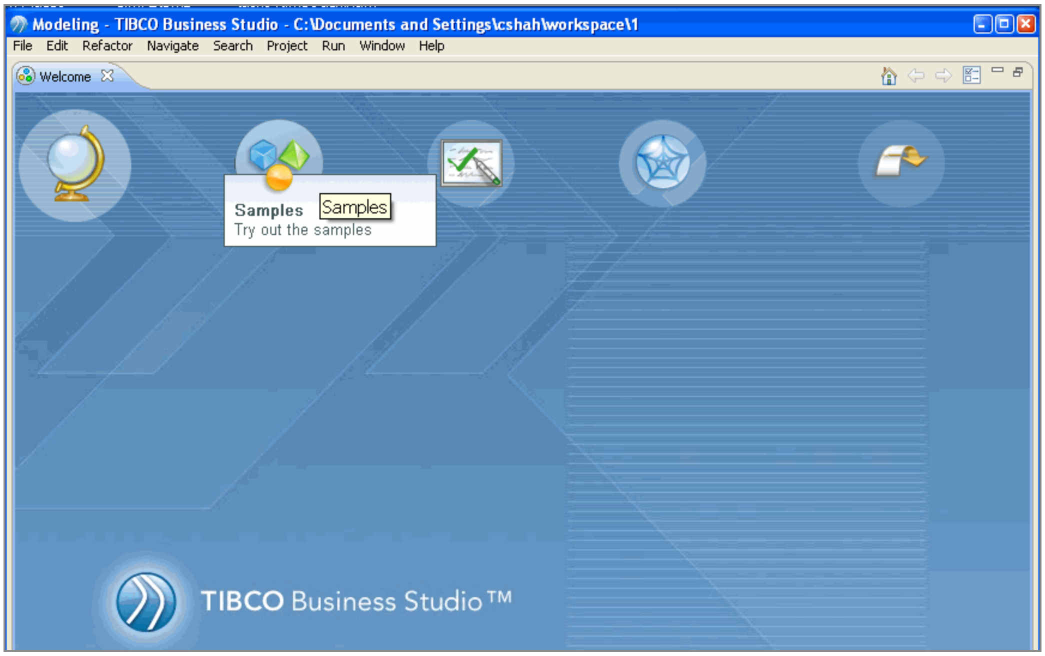
1. After the installation completes, start the Repository Designer by selecting **Start** > **Program Files** > **TIBCO** > **TIBCO_HOME** > **TIBCO MDM Studio <*version*>** > **Studio Designer**.

2. Provide a workspace location (folder where projects are saved).

> **Note:** You can create multiple workspaces.

Welcome Screen

After you select the workspace for the first time, Eclipse opens up with the Welcome screen. This contains icons to samples and tutorials among other things.


Modeling - TIBCO Business Studio - C:\Documents and Settings\cshah\workspace\1
File Edit Refactor Navigate Search Project Run Window Help
Welcome
Samples
Samples
Try out the samples
TIBCO Business Studio™

> **Note:** This Welcome screen shows up only the first time and will not be displayed for subsequent openings of Eclipse. If you want to go to this screen again, you can access it from **Help->Welcome**.

Accessing Samples

MDM Studio Samples are a collection of the MDM standard processes, process modelling tutorials, repository data models, rulebase model, custom import project, MDM Model templates, and Process java transitions.

The sample models are provided to illustrate the modelling capabilities of MDM Studio. Each of these models needs further elaborations for their intended purpose. Install the sample projects to view the MDM processes, data models, and their associated rules. All the samples are available in the TIBCO Home directory.

Follow these steps to install the **Samples**.

Procedure

1. On the **File** menu, click **Import**. The import wizard is displayed.
2. From the **General** folder, select **Existing Studio Projects into Workspace**.
3. Click **Next**. The import wizard for selecting the directory path is displayed.
4. Click **Select archive file** option. Click **Browse** and select the sample project zip archives from `\<TIBCOHome>\studio-mdm\<version>\samples` folder.
5. Click **Finish**. The selected project opens in the workspace.

Creating a new Repository Model

To create a repository model from scratch, you first need to create a Project that holds your model. The following are the steps involved:

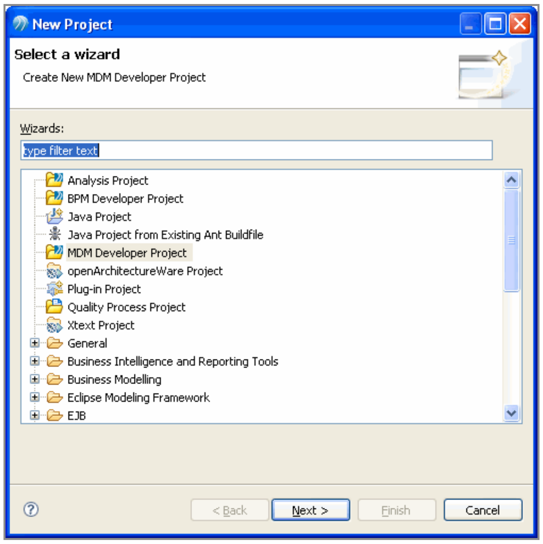
Creating a new Project to hold your Repository Model (Step 1)

Creating a Repository Model (Step 2)

Creating a new Project to hold your Repository Model (Step 1)

Procedure

1. Go to **File- > New- > Project**. Click **Next**. The Create New MDM Developer Project wizard is displayed.



New Project

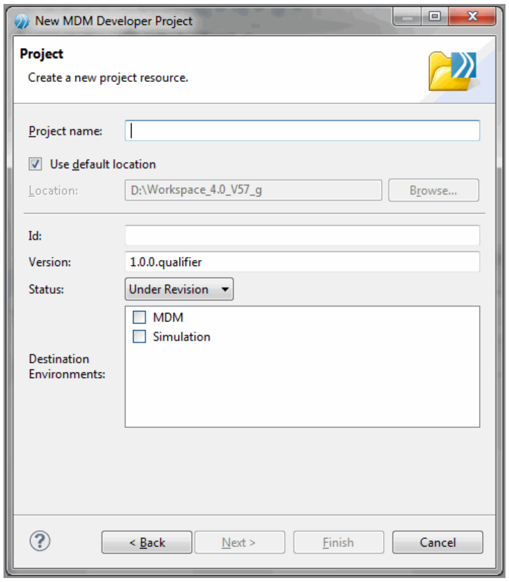
Select a wizard

Create New MDM Developer Project

Wizards: type filter text

2. Select **MDM Developer Project** and click **Next**.

3. Provide a name for the Project. Clear the **Use default location** checkbox if you want to provide a different location for the project (by default, the current workspace). Select Destination Environment as **MDM**. Click **Next**.



New MDM Developer Project

Project

Create a new project resource.

Project name:

☑ Use default location

Location: D:\Workspace_4.0_V57_g

Id:

Version: 1.0.0.qualifier

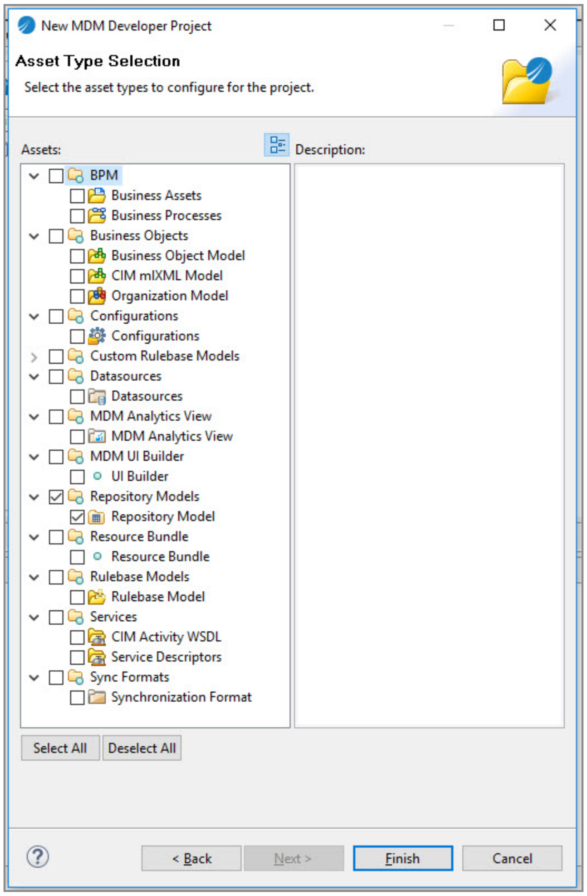
Status: Under Revision

Destination Environments:

☐ MDM

☐ Simulation

4. The Asset Type Selection dialog is displayed - select **Repository Models** and click **Next**.



New MDM Developer Project

Asset Type Selection

Select the asset types to configure for the project.

Assets:

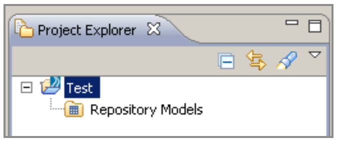
- □ BPM
  - □ Business Assets
  - □ Business Processes
- □ Business Objects
  - □ Business Object Model
  - □ CIM mIXML Model
  - □ Organization Model
- □ Configurations
  - □ Configurations
- □ Custom Rulebase Models
- □ Datasources
  - □ Datasources
- □ MDM Analytics View
  - □ MDM Analytics View
- □ MDM UI Builder
  - □ UI Builder
- ☑ Repository Models
  - ☑ Repository Model
- □ Resource Bundle
  - □ Resource Bundle
- □ Rulebase Models
  - □ Rulebase Model
- □ Services
  - □ CIM Activity WSDL
  - □ Service Descriptors
- □ Sync Formats
  - □ Synchronization Format

Description:

5. The folder for the Repository Model is displayed. Click **Finish**.

Result

This creates a new special folder **Repository Models**. Expand the project in the Project Explorer to see it.

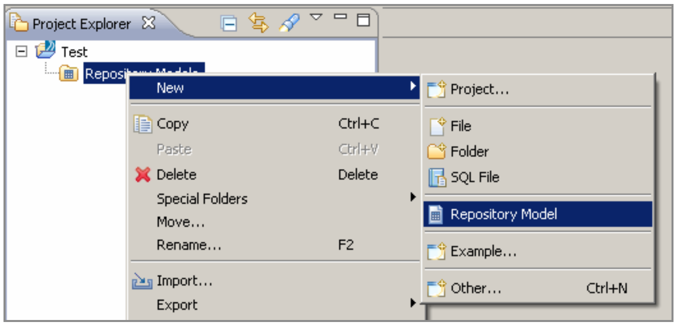


Creating a Repository Model (Step 2)

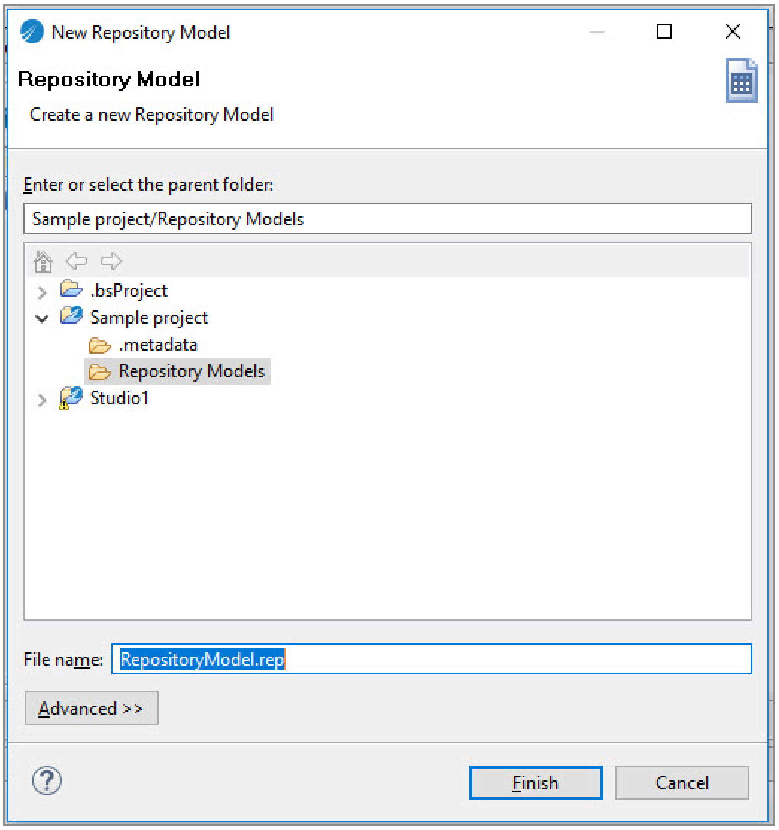
Procedure

1. Right-click the Repository Models folder in the Project Explorer and select New -

Repository Model.



2. Accept the default name for repository model (RepositoryModel.rep) and default location or enter a new location and name. Click **Finish**.



New Repository Model

Repository Model

Create a new Repository Model

Enter or select the parent folder: Sample project/Repository Models

File name: RepositoryModel.rep

Project Explorer

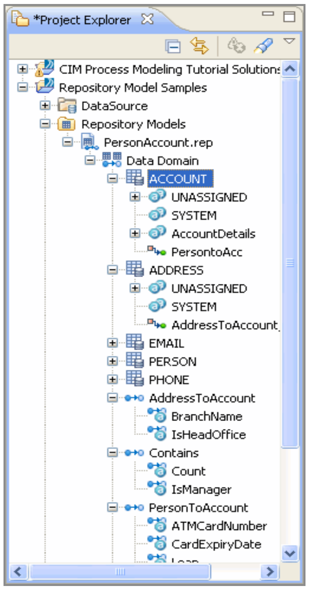
The Project Explorer contains the various MDM project folders.

MDM Project

The MDM project contains a **Repository Models** folder.

When you create a new (empty) repository model .rep file (see Creating a new Repository Model), the repositories you define (along with attributes and attribute groups) and relationships (along with relationship attributes and relationship attribute groups) are contained in a Data Domain within the .rep file. The Data Domain can be renamed by right clicking 'Data Domain' in the Project Explorer.

Expanding the Repository Models folder and all elements under it provides a hierarchical tree view and you can clearly see the constituents (repositories, inputs maps, attributes, relationships and so on) of your repository model.



If you double click the Repository Model in the Project Explorer, the editor shows an empty drawing canvas in which you can design your repository.

If you double click any other element, for example an attribute, the editor opens up (if not already open) with the selected element highlighted in the design view and its details are shown in the Properties tab.

Property tabs

The Property tab contains the following:

- Repository Property section
- Relationship Property section
- Relationship Attribute Group section

Repository Property section

On selecting a repository, you can see the following tabs through which you can view or modify Repository properties:

General Tab

The **General** tab displays basic repository properties. For more details, see General Tab

Attributes Tab

The **Attributes** tab displays Repository Attributes and details. For more details, see Attributes Tab

Group Sequence Tab

The **Group Sequence** tab displays attribute group sequence. For more details, see Group Sequence.

Input Maps Tab

The **Input Maps** tab displays Repository Input Maps. For more details, see Input Maps Tab.

Output Maps Tab

The **Output Maps** tab displays Repository Output Maps. For more details, see Output Maps.

Classification Schemes Tab

The **Classification Schemes** tab displays the Classification Scheme. For more details, see Classification Scheme Editor.

Perspectives Tab

The **Perspectives** tab displays the perspectives. For more details, see Perspective Editor.

Configuration Tab

The **Configuration** tab displays the perspectives. For more details, see Configuration Tab.

Appearance Tab

The **Appearance** tab provides the parameters for modifying the appearance. For more details, see Appearance Tab.

Relationship Property section

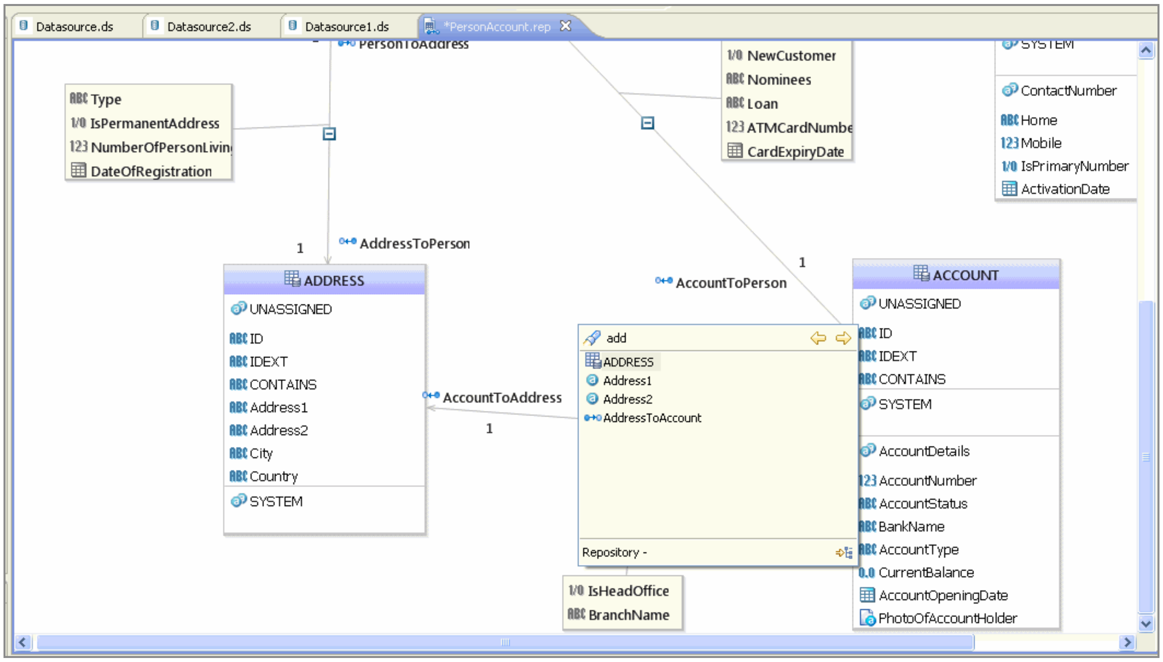
On selecting a Relationship, you can view or modify Relationship properties through the Properties pane, **General** tab. For more details, see Relationship Properties.

Relationship Attribute Group section

On selecting a Relationship Attribute Group, you can view or modify properties through the Properties pane, **General** tab and **Attributes** tab. For more details, see Relationship Attribute Properties.

Quick Search

Using the type ahead search dialog you can find diagram entities such as repositories, attributes or relationships. Press Ctrl + F or click 🔦 in the toolbar on selected diagram.

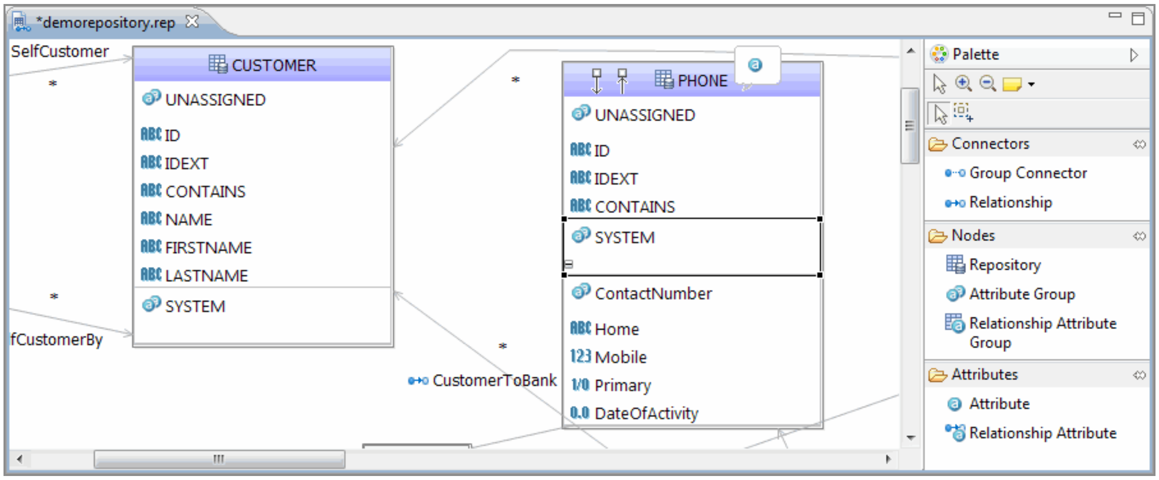


Attribute and Attribute Group Repositioning

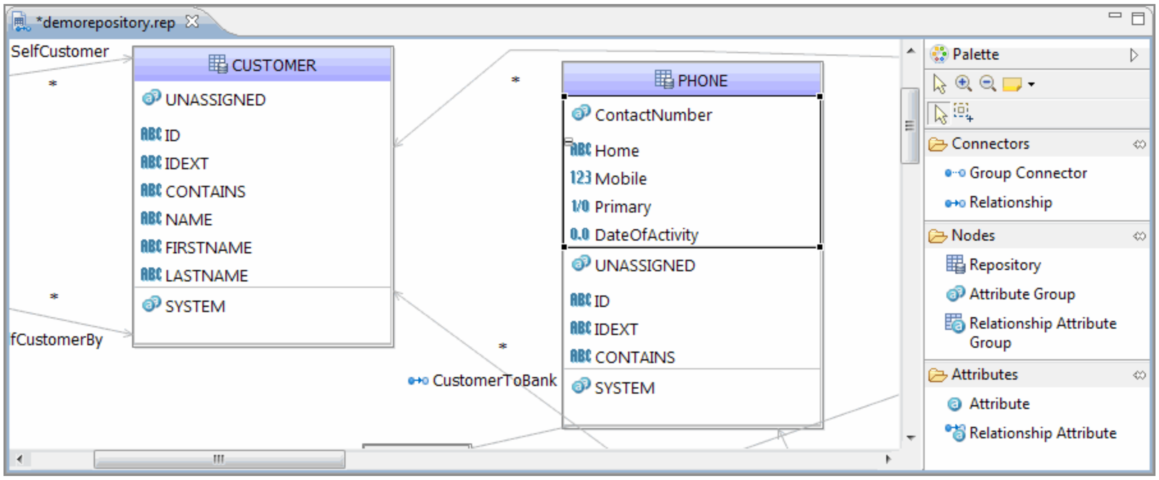
The attributes and Attributes groups can be repositioned within the same repository.

Attribute Group Reposition

The repository attribute group can be repositioned within the repository with drag and drop. For example, the "Contact Number" attribute group is repositioned from the bottom of the repository to the top of the repository.

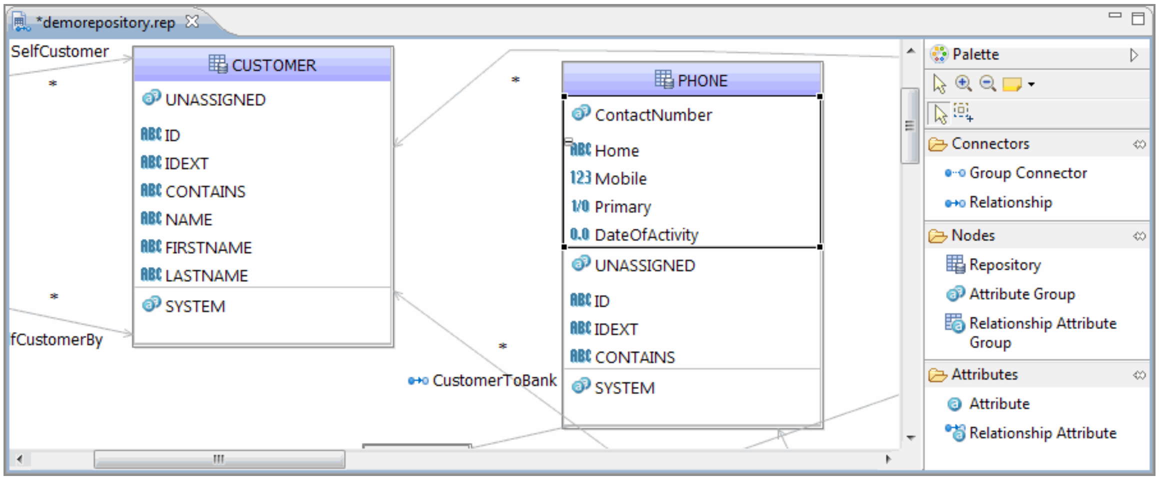


The "Contact Number" attribute group is displayed on the top of the repository.

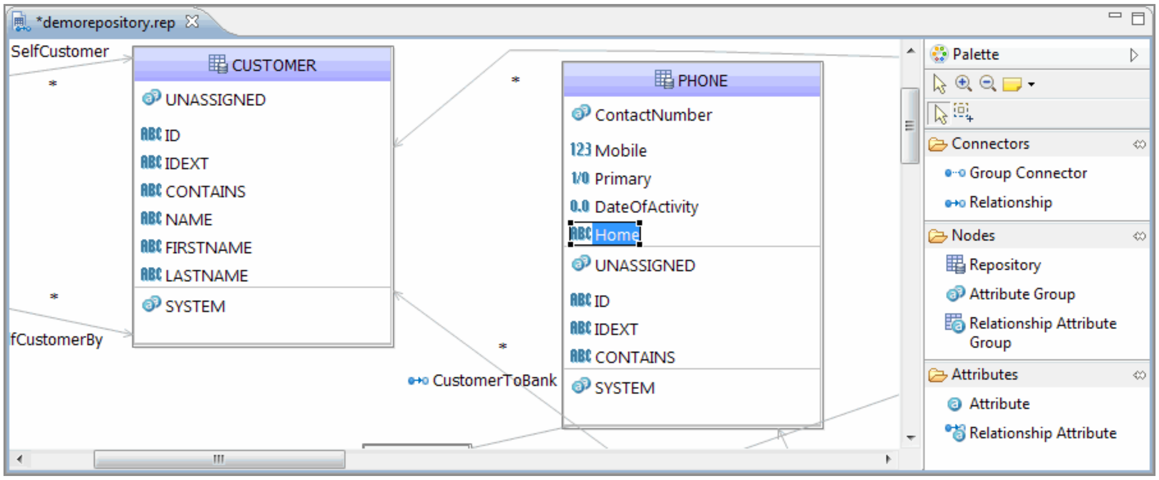


Attribute Reposition within same Attribute Group

The repository attributes can be repositioned within the same attribute group in a repository with drag and Drop. For example, the "Home" attribute is repositioned from top of the attribute group to the bottom of the same attribute group.

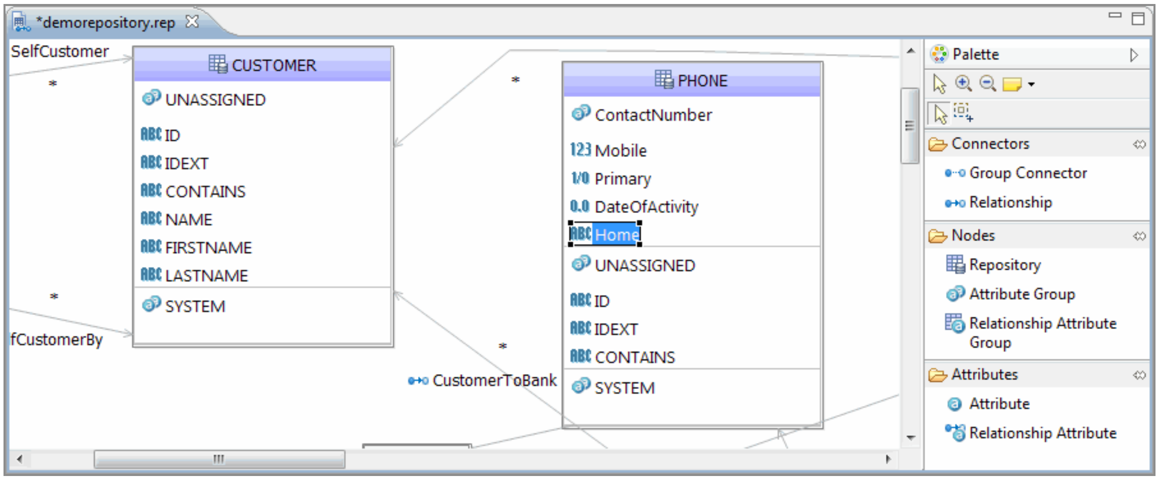


The "Home" attribute is displayed at the bottom of the "Contact Number" attribute group.

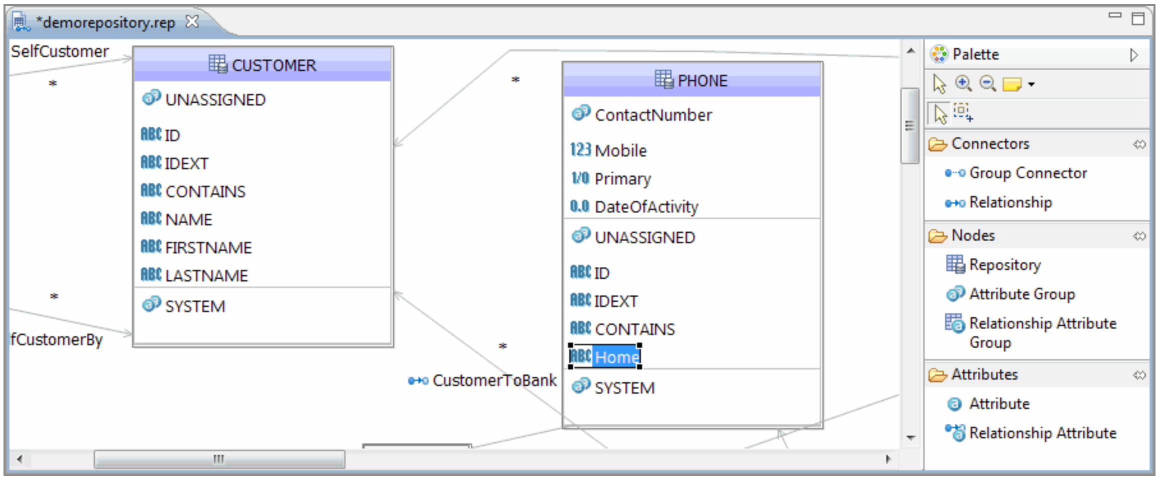


Attribute Reposition from one Attribute Group to another Attribute Group

The repository attribute can be repositioned from one attribute group to another attribute group within the same repository with drag and drop. If a attribute is repositioned from one attribute group to other, then it is added as a last attribute in the target attribute group. For example, the "Home" attribute is repositioned from the "Contact Number" attribute group to the "Unassigned" attribute group. It is added as the last attribute.

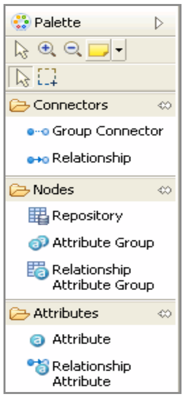


The "Home" attribute is repositioned from "Contact Number" attribute group to the "Unassigned" attribute group.



Palette

The Palette (to the right of the screen) contains different artifacts to help you build your repository model. Select and drop into the main drawing pane to define or modify your repository model. You can do the following with the Palette:



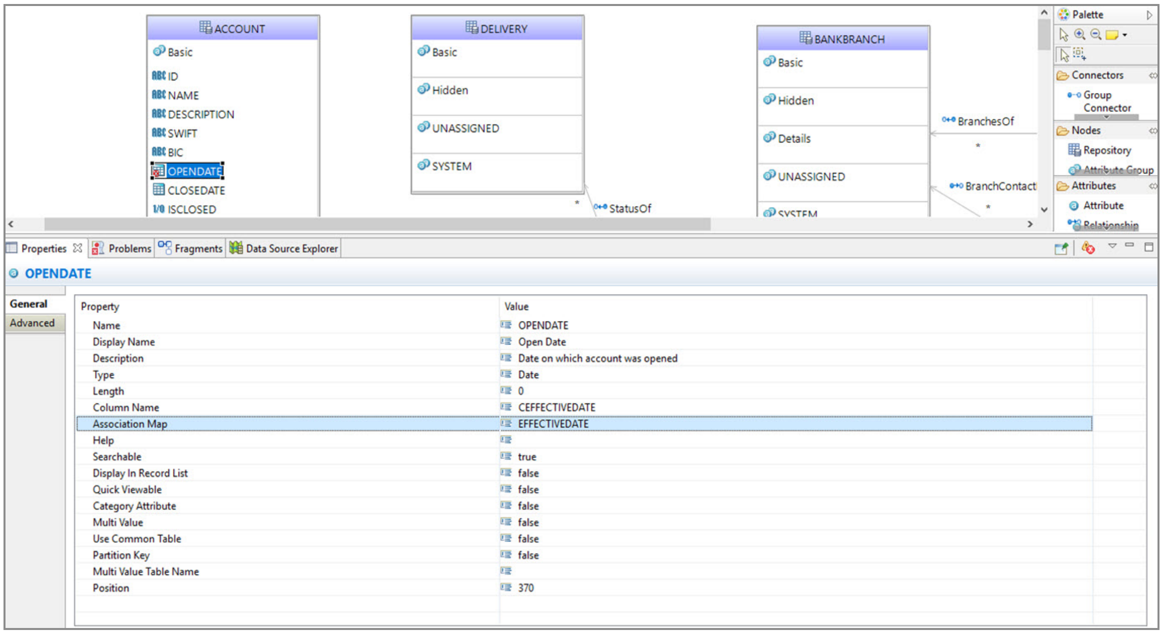
Connectors	
Relationship	Create a relationship between repositories.
Group Connector	Connect a Relationship Attribute Group to a Relationship.
Nodes	
Repository	Create a MDM repository.
Relationship Attribute Group	Create a Relationship Attribute Group.
Attribute Group	Create an Attribute Group.
Attributes	
Attribute	Create an attribute within a repository.
Relationship Attribute	Create a relationship attribute within a Relationship Attribute Group.

Association Mapping

Using the Association Mapping property, you can choose any attribute as ID (Record Id), IDEXT (Record ID Extension), or EFFECTIVEDATE. The available mappings in the Association Mapping drop-down list are **NONE**, **ID**, **IDEXT**, and **EFFECTIVEDATE**.

EFFECTIVEDATE Mapping

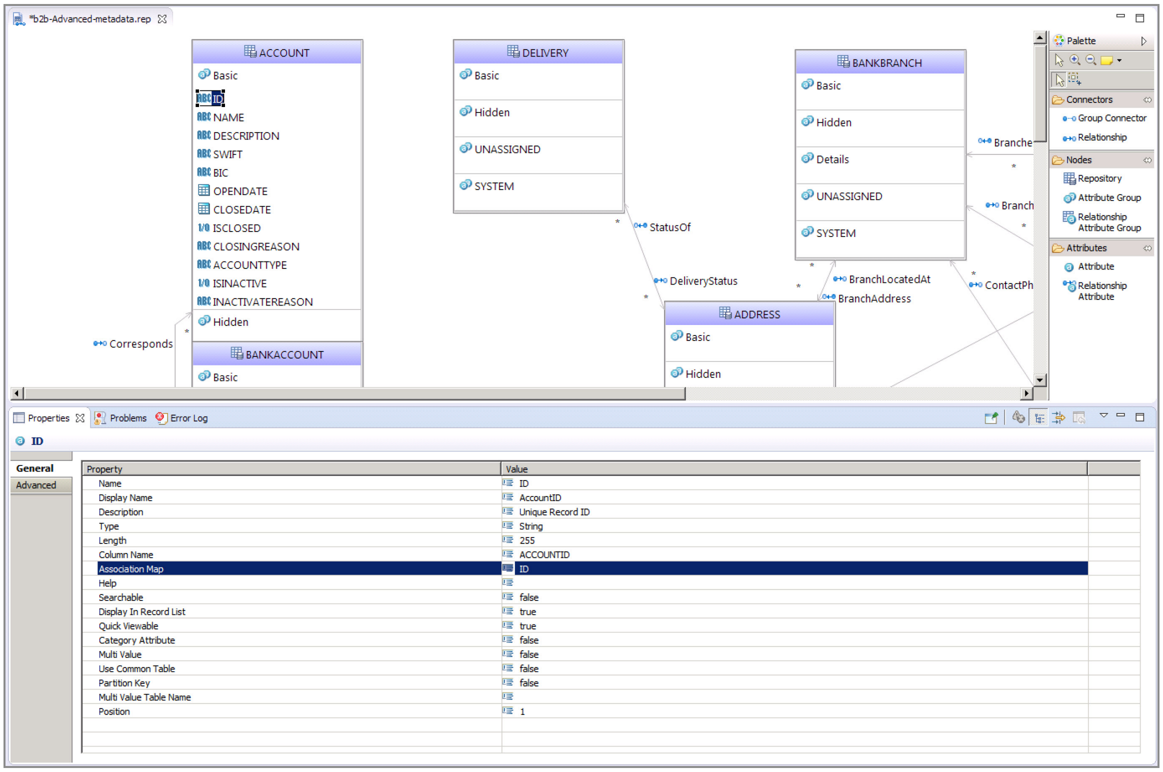
The’Name’ and ’Display Name’ property shows attribute name. The Display name does not change when you set the association property. So when you deploy a Repository Model then the Attribute name gets changed to EFFECTIVEDATE and the Display Name would be same as defined while creating the attribute group.



OPENDATE

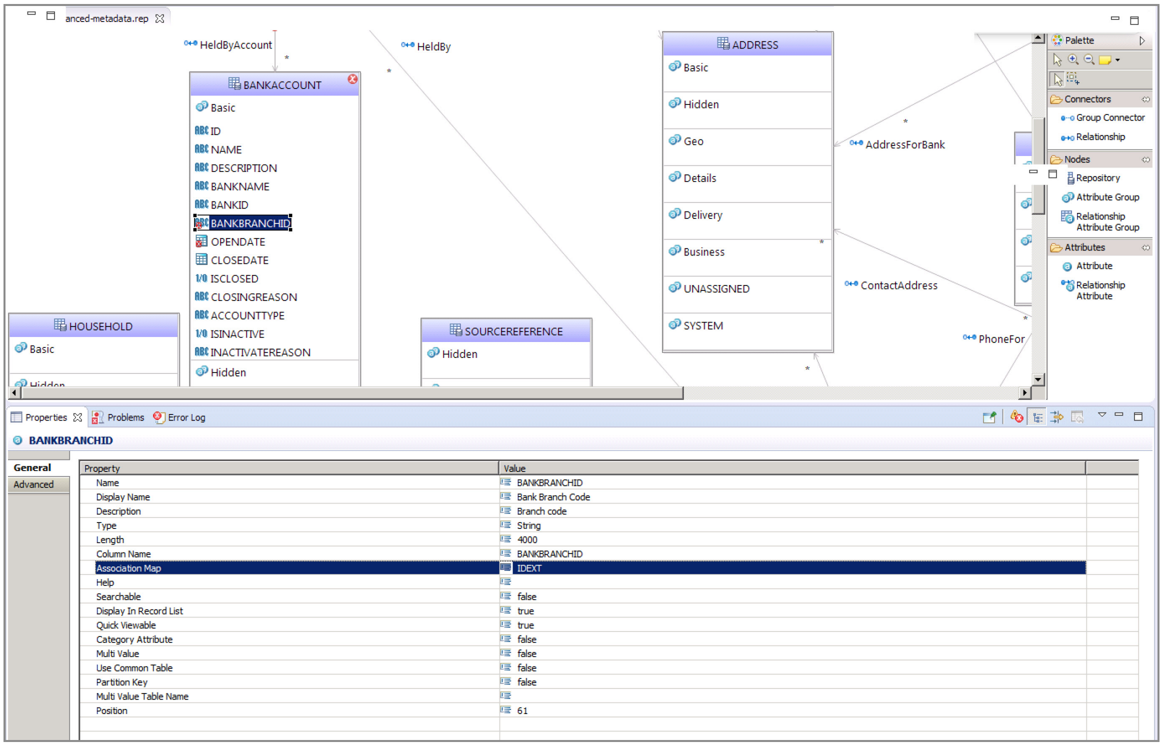
Property	Value
Name	OPENDATE
Display Name	Open Date
Description	Date on which account was opened
Type	Date
Length	0
Column Name	CEFFECTIVEDATE
Association Map	EFFECTIVEDATE
Help	
Searchable	true
Display In Record List	false
Quick Viewable	false
Category Attribute	false
Multi Value	false
Use Common Table	false
Partition Key	false
Multi Value Table Name	
Position	370

PRODUCT ID Mapping



Property	Value
Name	ID
Display Name	AccountID
Description	Unique Record ID
Type	String
Length	255
Column Name	ACCOUNTID
Association Map	ID
Help	
Searchable	false
Display In Record List	true
Quick Viewable	true
Category Attribute	false
Multi Value	false
Use Common Table	false
Partition Key	false
Multi Value Table Name	
Position	1

PRODUCTIDTEXT Mapping



BANKBRANCHID

Property	Value
Name	BANKBRANCHID
Display Name	Bank Branch Code
Description	Branch code
Type	String
Length	4000
Column Name	BANKBRANCHID
Association Map	IDEXT
Help	
Searchable	false
Display In Record List	true
Quick Viewable	true
Category Attribute	false
Multi Value	false
Use Common Table	false
Partition Key	false
Multi Value Table Name	
Position	61

Repository Attributes and Relationships

This chapter explains how to design your repository complete with attributes, relationships, and relationship attributes.

Repository Definition

The following are the major steps involved in designing your repository:

- Adding Repositories
- Defining Attribute Groups
- Adding Attributes
- Creating Relationships
- Creating Relationship Attribute Groups
- Defining Relationship Attributes within a Relationship Attribute Group
- Connecting Relationship Attribute Groups to Relationships
- Validating your repository model
- Exporting your repository model

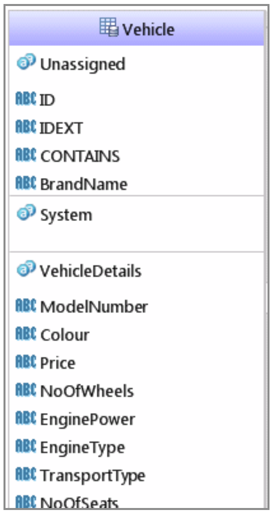
Adding a Repository (Step 1)

You can add the repository from the Palette and also from editor by hovering cursor on data domain.

From the Palette

To add a new repository, click the Repository icon in the Palette and then click in the main drawing pane to insert the new Repository; you will be prompted to provide a name for the repository.

By default, the repository contains 2 attribute groups Unassigned and System (more attribute groups can be added as required) with predefined attributes.



Vehicle

Unassigned

ABC ID

ABC IDEXT

ABC CONTAINS

ABC BrandName

System

VehicleDetails

ABC ModelNumber

ABC Colour

ABC Price

ABC NoOfWheels

ABC EnginePower

ABC EngineType

ABC TransportType

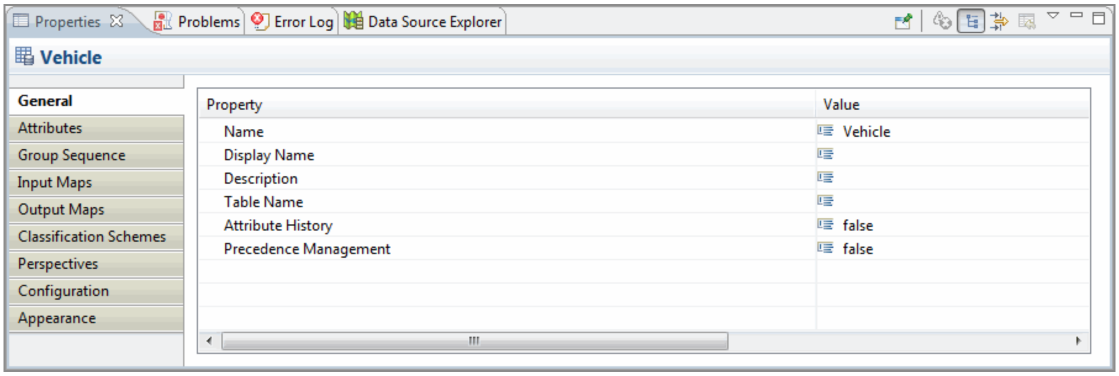
ABC NoOfSeats

When you select a repository in the drawing pane, the Properties Pane displays repository properties in nine tabs:

General Tab

The **General** tab displays basic repository properties such as the Name, Display Name, Description, Table Name, Attribute History and Precedence Management. For more information on History and Precedence Management refer to Precedence Management section in *TIBCO MDM Customization Guide*.

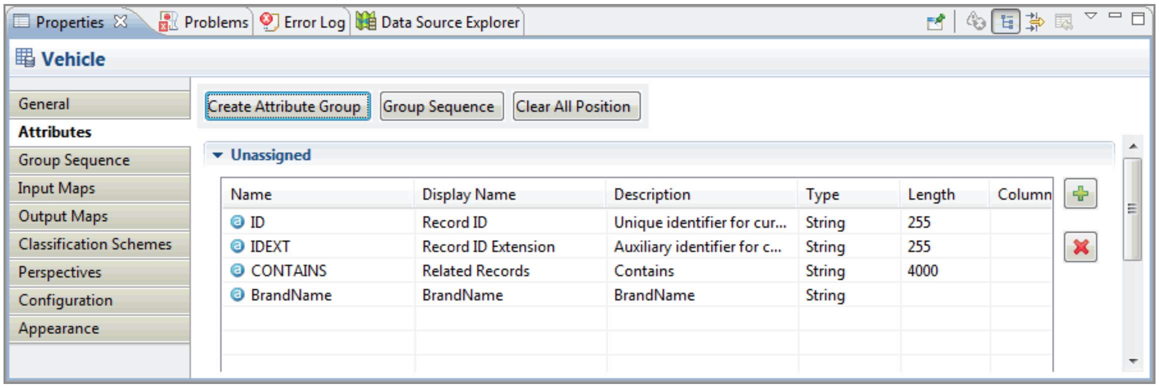
- Attribute History - Specifies whether to collect attribute history for the repository.
- Precedence Management - Specifies whether to enable precedence management for the repository. This flag cannot be switched on unless Attribute History is set to true. When it set to true, a new attribute SOURCEID is added under the System group in the repository. You can move the SOURCEID attribute to another group but this attribute cannot be deleted. If you disable the precedence management, the SOURCEID attribute is not removed.



Property	Value
Name	Vehicle
Display Name	
Description	
Table Name	
Attribute History	false
Precedence Management	false

Attributes Tab

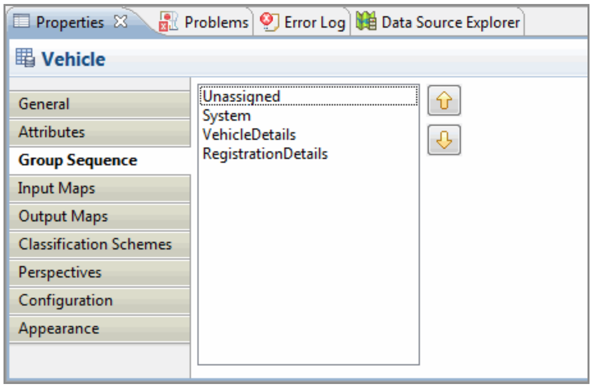
The **Attributes** tab displays Repository Attributes and details. You can view, add, delete attributes, edit attribute properties, move attribute positions, and create Attribute Groups.



Name	Display Name	Description	Type	Length	Column
ID	Record ID	Unique identifier for cur...	String	255	
IDEXT	Record ID Extension	Auxiliary identifier for c...	String	255	
CONTAINS	Related Records	Contains	String	4000	
BrandName	BrandName	BrandName	String		

Group Sequence Tab

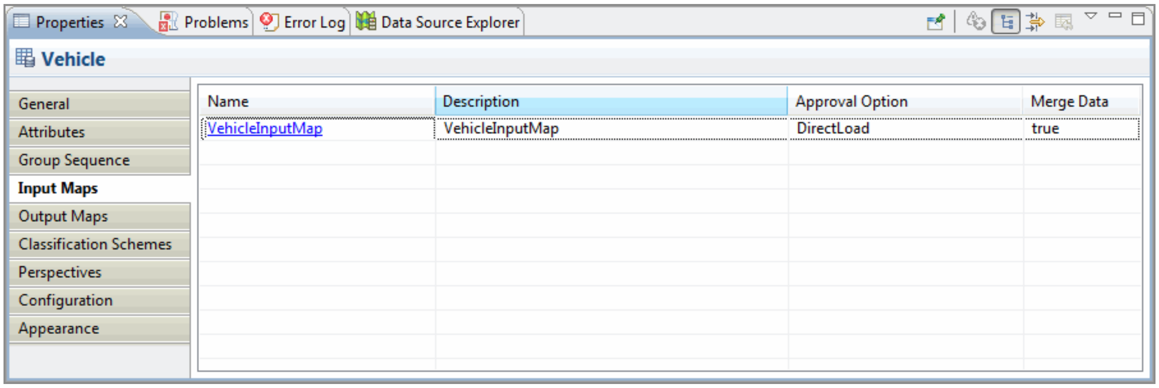
The **Group Sequence** tab displays the current sequence of attribute groups which you can reorder as required.



Unassigned
System
VehicleDetails
RegistrationDetails

Input Maps Tab

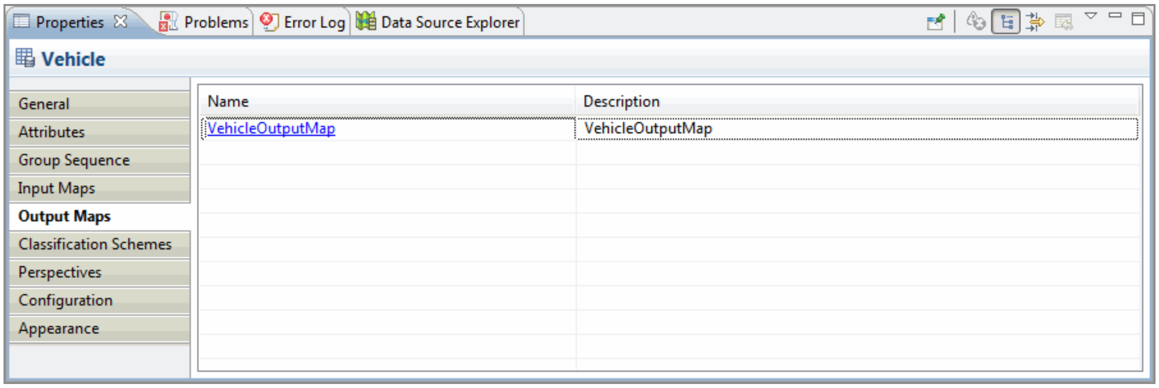
The **Input Maps** tab displays Repository Input Maps. For more information on input map refer, Input Maps



Name	Description	Approval Option	Merge Data
VehicleInputMap	VehicleInputMap	DirectLoad	true

Output Maps Tab

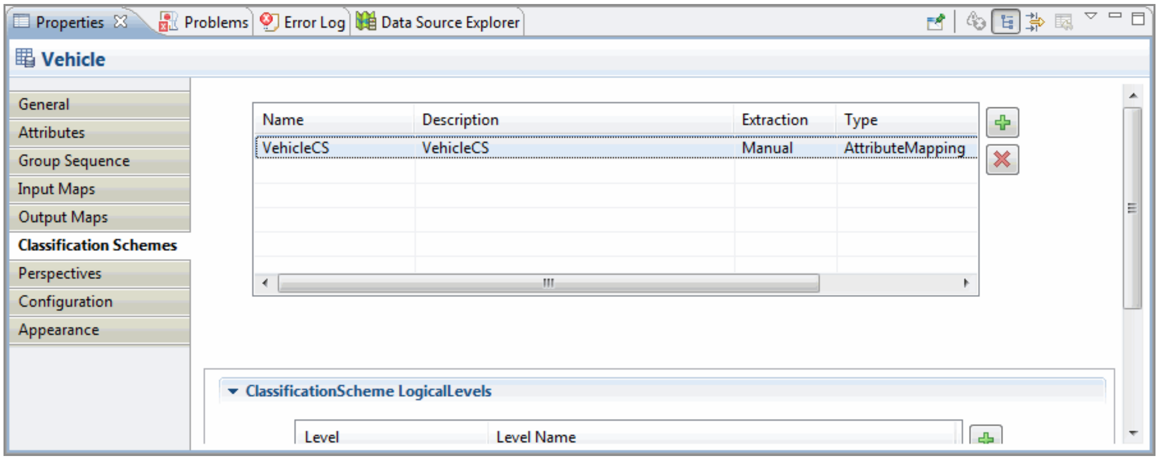
The **Output Maps** tab displays Repository Output Maps. For more information on output map refer, Output Maps



Name	Description
VehicleOutputMap	VehicleOutputMap

Classification Schemes Tab

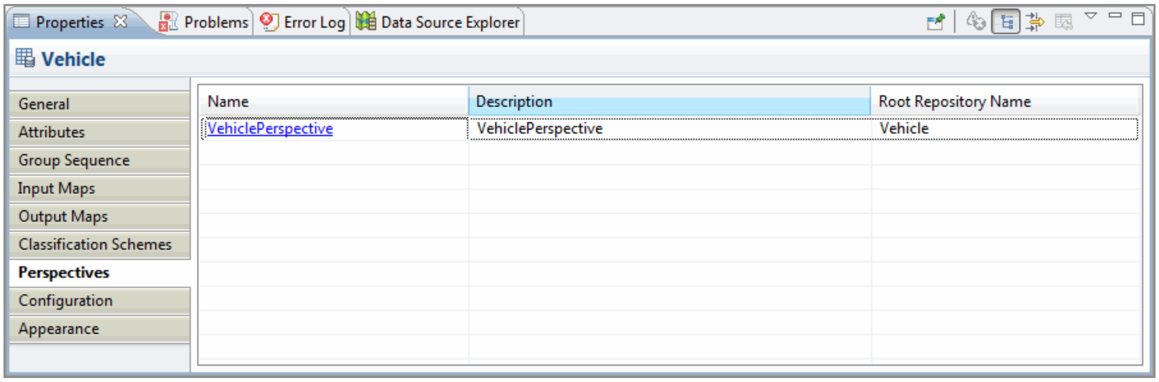
The **Classification Schemes** tab allows you to view and create classification scheme. For more information on creating classification scheme using the property section refer, Using Classification Scheme Property Section



Name	Description	Extraction	Type
VehicleCS	VehicleCS	Manual	AttributeMapping

Perspectives Tab

The **Perspectives** tab displays the Perspective defined for a particular repository. For more information on perspective refer, Perspective Editor Overview

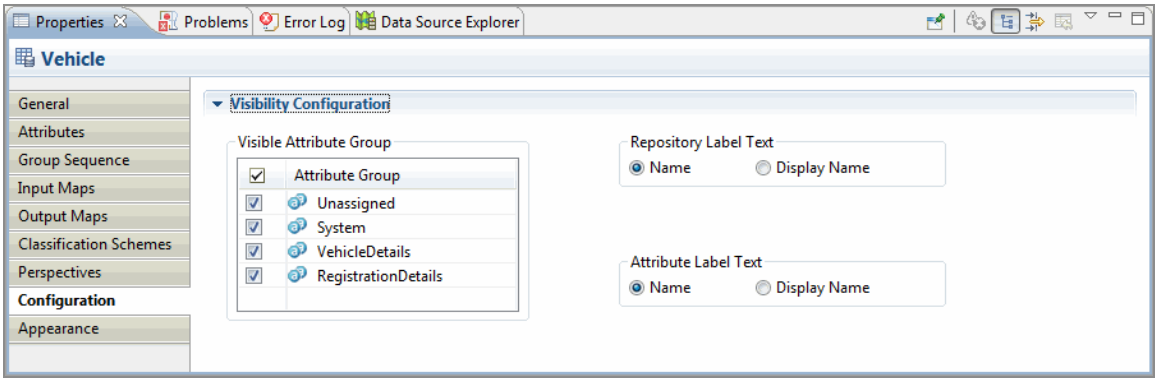


Name	Description	Root Repository Name
VehiclePerspective	VehiclePerspective	Vehicle

Configuration Tab

The **Configuration** tab can be used to modify the visibility configuration of the repository. By default, the attributes groups are visible, to switch the visibility of the attribute group, uncheck the checkbox corresponding to the attribute group.

You can change the **Repository Label Text** of the repository to display either the Name or Display Name. Similarly you can change the **Attribute Label Text** to either Name or Display Name.



Visible Attribute Group

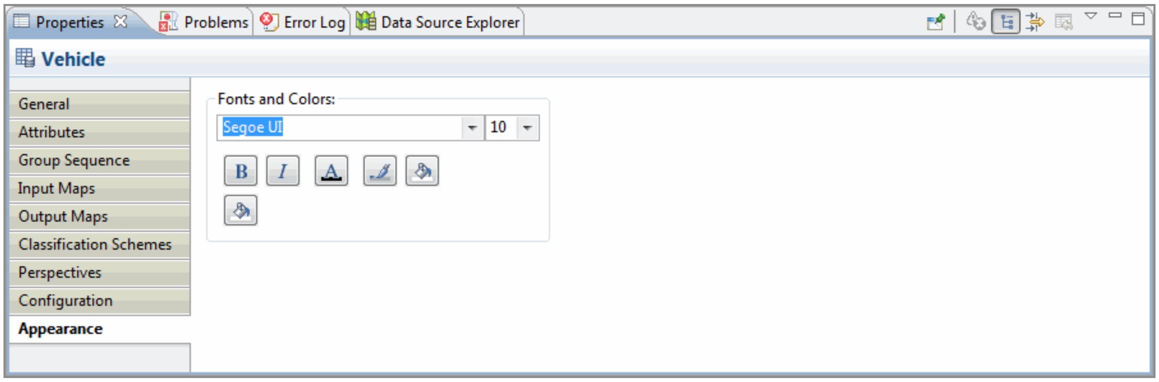
☑	Attribute Group
☑	Unassigned
☑	System
☑	VehicleDetails
☑	RegistrationDetails

Repository Label Text
● Name ○ Display Name

Attribute Label Text
● Name ○ Display Name

Appearance Tab

The **Appearance** tab is used to modify the appearance of the particular repository, attribute group, relationship attribute, and relationships. For more information on Appearance refer, Appearance Tab.



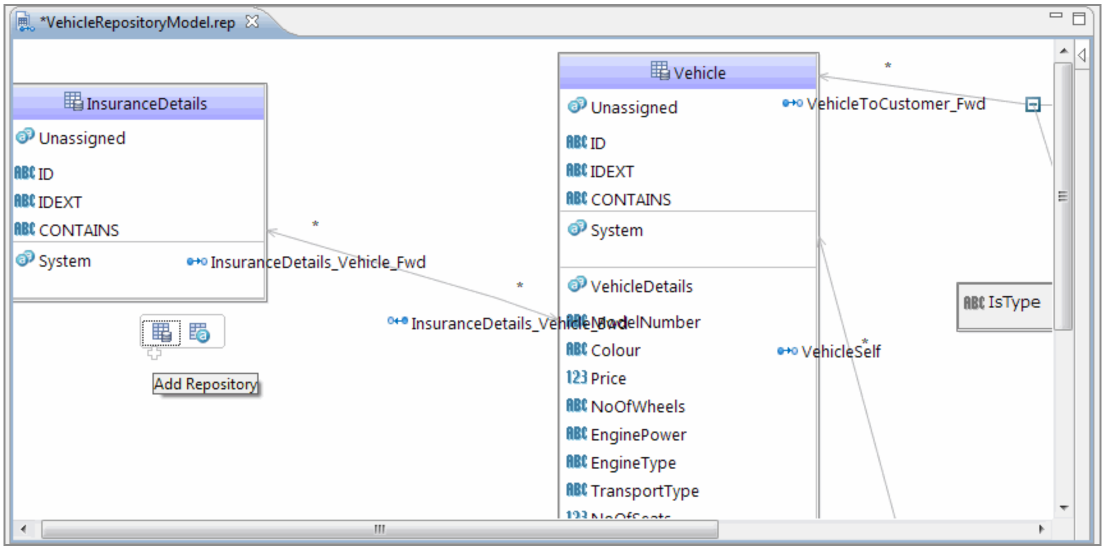
Fonts and Colors:
Segoe UI 10

From the Callout Handler

You can add the Repository using the callout handler in the editor.

Procedure

1. Hover the mouse on the editor, the **Add Repository** and **Add Relationship Attribute Group** icons are displayed.



2. Click the **Add Repository** icon, the new Repository gets added in the editor.

3. Similarly click the **Add Relationship Attribute Group** icon, the new Add Relationship Attribute Group gets added in the editor.

Defining Custom Properties for Repositories

You can define custom properties for a repository by using the **Configurations** > `repositoryAdvancedProperties.properties` file. To define new custom properties for a repository, complete the following the steps:

Procedure

1. From the Project Explorer, navigate to **Configurations** and open the `repositoryAdvancedProperties.properties` file.

2. Add new properties in the following format `key=Datatype#Value`. For example, `match=String#false,true`.

   For a multivalue property, add comma separated values. You can view repository specific properties and edit their values in **Properties > Advanced**.

Defining Attribute Groups (Step 2)

An Attribute Group represents a logical grouping of the attributes in a repository. Each attribute is assigned to an attribute group.

To create an attribute group:

Procedure

1. Click the Attribute Group icon in the Palette and then click in the Repository to add the Attribute Group.
2. Enter a name for the attribute group. You can now add new attributes to the group.

   By default, a repository contains 2 attribute groups Unassigned and System.

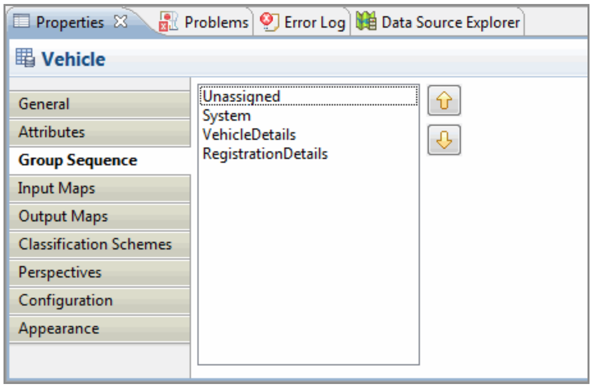
   - **Unassigned Attribute Group**

     This group contains the ID, IDEXT and CONTAINS predefined attributes.

     - ID represents the primary identity attribute of a data record. It maps to the TIBCO MDM attribute PRODUCTID and it is a unique identifier for each of the records.
     - IDEXT is the secondary identifying attribute for a data record and maps to PRODUCTIDEXT in TIBCO MDM.
     - CONTAINS maps to the CONTAINS attribute (Related Record) in TIBCO MDM.
   - **System Attribute Group**

     This contains system attributes (such as Last Modified On, Last Modified By, Active, Created On, Owner Id, Last Confirmed Version) that cannot be seen but are present to maintain version and audit information. If you define any custom system attributes, you can see them when you expand the group.
   - **Group Sequence**

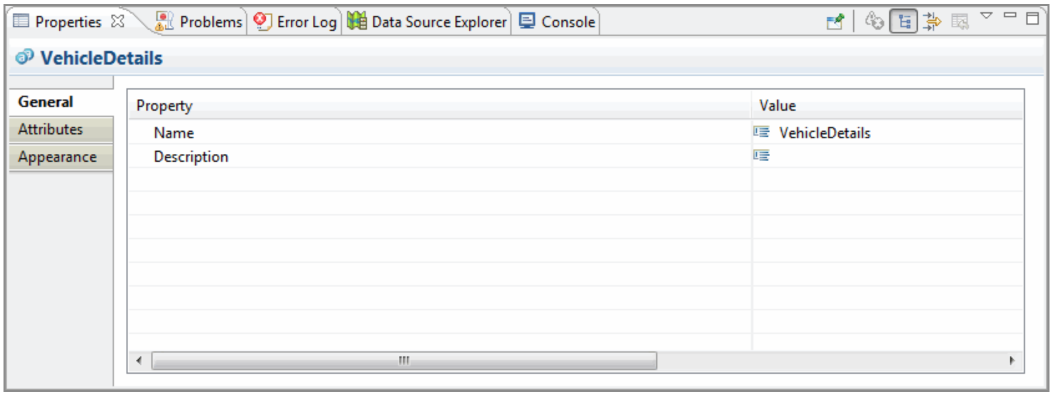
     The order of attribute groups can be controlled from the **Group Sequence** tab. Select a repository in the drawing pane and click the **Group Sequence** tab in the **Properties** Pane. A list of all attributes groups for that repository is displayed and you can change the order by selecting the group to move and using the up and down buttons.



Unassigned
System
VehicleDetails
RegistrationDetails

- **Attribute Group Properties**

  The General tab displays basic attribute group properties such as the Name and Description.



Property	Value
Name	VehicleDetails
Description	

Note: You can modify the Attribute Group name in the editor by double clicking on the group name and inline editing.

Defining Custom Properties for Attribute Groups

You can define custom properties for an attribute group by using the **Configurations** > `attrGroupAdvancedProperties.properties` file. To define new custom properties for an attribute group, complete the following the steps:

Procedure

1. From the Project Explorer, navigate to **Configurations** and open the `attrGroupAdvancedProperties.properties` file.

2. Add new properties in the following format `key=Datatype#Value`. For example, `match=String#false,true`.

   For a multivalue property, add comma separated values. You can view the attribute group specific properties and edit their values in **Properties > Advanced**.

Adding Predefined Attribute Groups

By using the predefined attribute group feature in TIBCO MDM Studio, you can create customized attribute groups based on your specific needs. You can use these attribute groups in any repository model across the project.

These attribute groups can be defined in the `predefinedAttributeGroups.xml` configuration file. This file is added in the Configurations folder when you create a new project in TIBCO MDM Studio. By default, this file has no content.

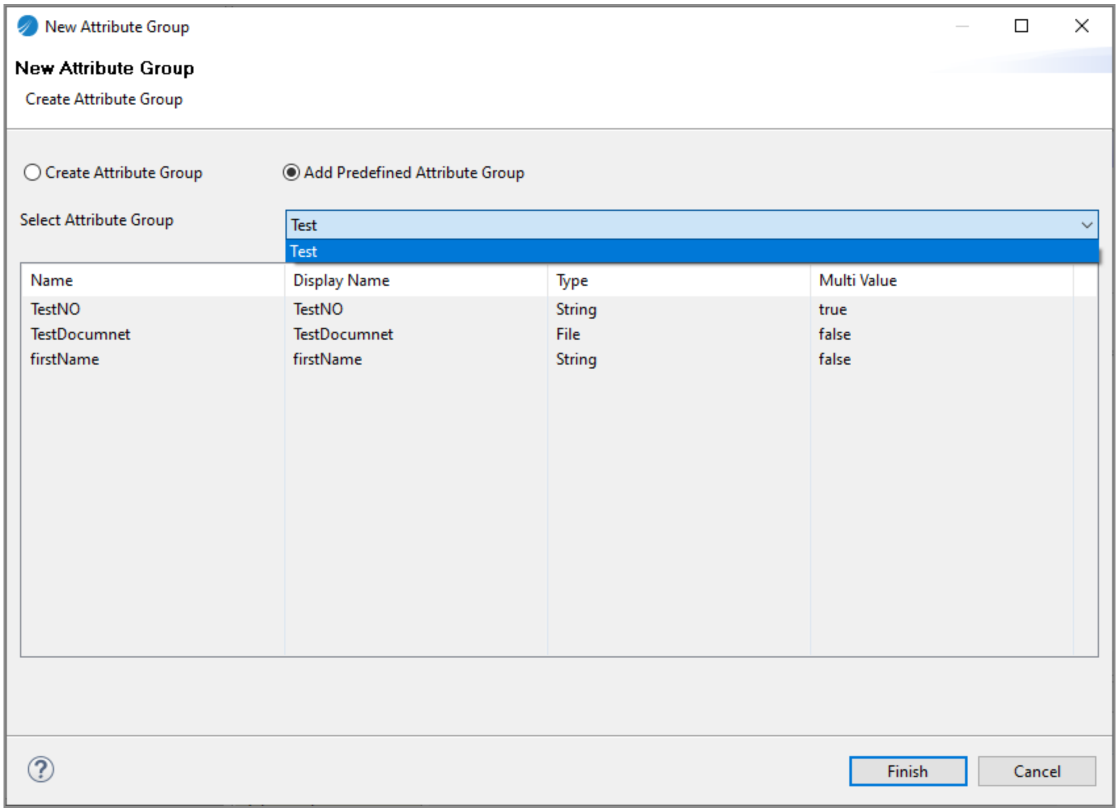
Before you begin

Ensure that you have configured your attribute groups in the `predefinedAttributeGroups.xml` configuration file and placed it in the Configurations folder.

Complete the following steps to add predefined attribute groups:

Procedure

1. Select a repository and then click **Properties > Attributes > Add Attribute Group**. The **New Attribute Group** dialog is displayed.



New Attribute Group

New Attribute Group

Create Attribute Group

○ Create Attribute Group ● Add Predefined Attribute Group

Select Attribute Group: Test

Name	Display Name	Type	Multi Value
TestNO	TestNO	String	true
TestDocumnet	TestDocumnet	File	false
firstName	firstName	String	false

Note: Ensure that the new attribute group has a unique name. A validation error message is displayed if there is already an attribute group with the same name in the selected repository.

Note: Ensure that the new attribute group has unique attributes. A validation error message is displayed if the same attributes are present in other attribute groups in the selected repository.

2. Select **Add Predefined Attribute Group** and then select an attribute group from the **Select Attribute Group** dropdown list. The list of attribute groups displayed in this dropdown list are defined in the `predefinedAttributeGroups.xml` configuration file.

3. After you have selected an attribute group, click **Finish**. The selected attribute group is added to the repository.

Sample predefinedAttributeGroups.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<AttributeGroups>
<AttributeGroup>
<AttributeGroupInfo>
<Name>SampleAttributeGroup1</Name>
<Description>SampleAttributeGroup1</Description>
<Property type="Boolean" name="sampleProperty1">true</Property>
<Property type="String" name="sampleProperty2">sampleProperty</Property>
</AttributeGroupInfo>
<Attributes>
<Attribute>
<Name>SampleAttribute1</Name>
<Description>Sample attribute 1</Description>
<DisplayName>SampleAttribute1</DisplayName>
<Length>255</Length>
<ColumnName>attributeColumn1</ColumnName>
<Help/>
<DataType>String</DataType>
<Searchable>N</Searchable>
<DisplayInRecordList>Y</DisplayInRecordList>
<QuickViewable>Y</QuickViewable>
<MultiValued>N</MultiValued>
<CategorySpecific>N</CategorySpecific>
<PartitionKey>N</PartitionKey>
<MultiValueTableName/>
<UseCommonTable>N</UseCommonTable>
<Property type="Boolean" name="attribute1property1">true</Property>
<Property type="String"
name="attribute1property2">AttributeProperty</Property>
</Attribute>
<Attribute>
<Name>SampleAttribute2</Name>
<Description>Sample Attribute 2</Description>
<DisplayName>SampleAttribute2</DisplayName>
<Length>0</Length>
<ColumnName>attributeColumn2</ColumnName>
<Help/>
<DataType>Decimal</DataType>
<Searchable>N</Searchable>
<DisplayInRecordList>N</DisplayInRecordList>
<QuickViewable/>
<MultiValued>N</MultiValued>
<CategorySpecific>N</CategorySpecific>
<PartitionKey>N</PartitionKey>
<MultiValueTableName/>
<UseCommonTable>N</UseCommonTable>
```

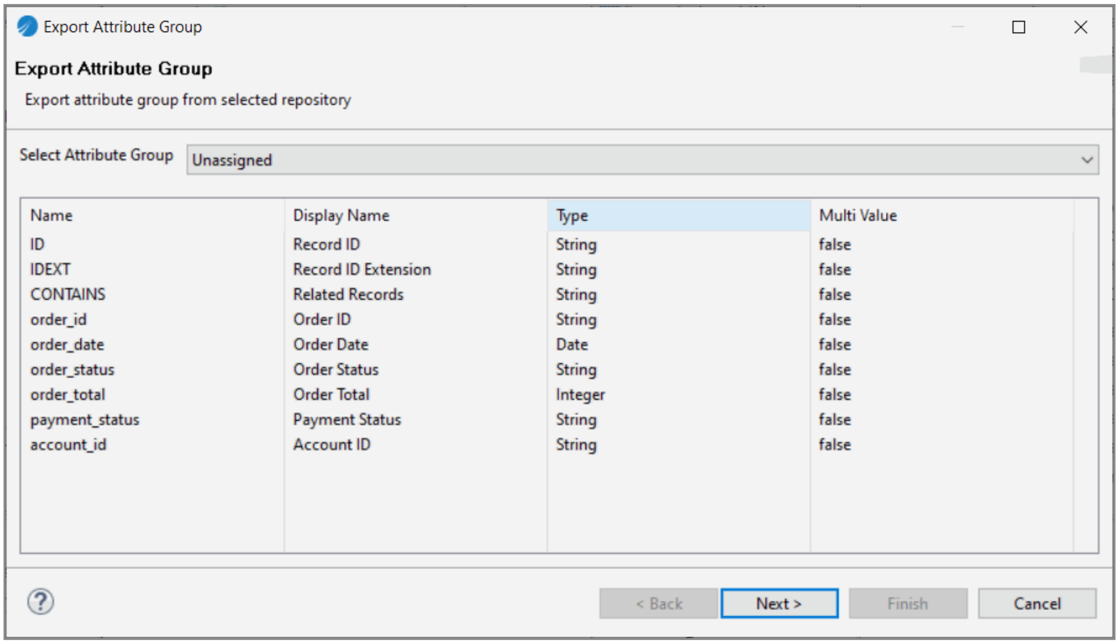
```
<Property type="Boolean" name="attribute2Property1">false</Property>
<Property type="String"
name="attribute2Property2">AttributeProperty</Property>
</Attribute>
</Attributes>
</AttributeGroup>
</AttributeGroups>
```

Exporting Attribute Groups

You can export existing attribute groups to use in other repositories. Follow these steps to export an attribute group:

Procedure

1. Select a repository from where you want to export the attribute group.
2. In the Properties tab, click **Attributes** and then click **Export Attribute Group**. The Export Attribute Group wizard is displayed.



Select Attribute Group: Unassigned

Name	Display Name	Type	Multi Value
ID	Record ID	String	false
IDEXT	Record ID Extension	String	false
CONTAINS	Related Records	String	false
order_id	Order ID	String	false
order_date	Order Date	Date	false
order_status	Order Status	String	false
order_total	Order Total	Integer	false
payment_status	Payment Status	String	false
account_id	Account ID	String	false

The Select Attribute Group dropdown menu lists all the attribute groups present in the selected repository.

3. Select the attribute group that you want to export from the **Select Attribute Group**

dropdown list and click **Next**. The next page is displayed where you can validate the selected attribute group.

4. Provide a unique name for the attribute group and click **Validate**. Ensure that the Attribute Group Name is unique and not already present in the `predefinedAttributeGroups.xml` file. If the validation is successful, it means the name is unique. If the validation fails and you still click **Finish**, the selected attribute group in the `predefinedAttributeGroups.xml` is overwritten.
5. Click **Finish** to export the selected attribute group.

Defining Attributes (Step 3)

Attributes define the structure of a repository. To create attributes:

- Click the Attribute icon in the Palette and then click in the Repository (under an attribute group) to add the Attribute.
- Enter an attribute name.

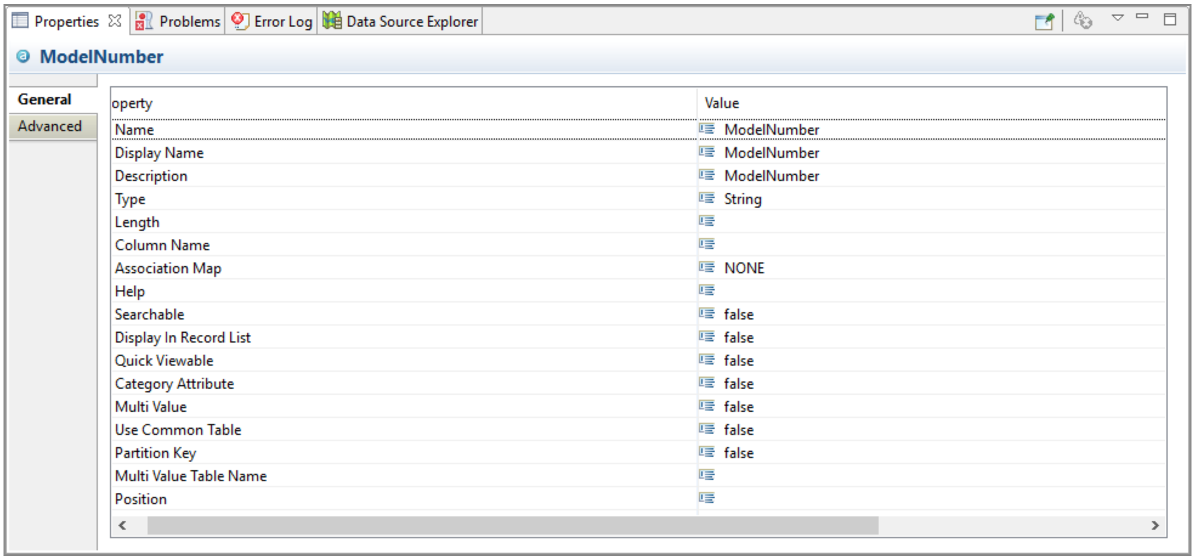
Attribute properties can be edited directly by clicking the attribute or selecting the repository and then clicking the **Attributes** tab in the **Properties** window.

Attributes can be reordered by dragging and dropping the attribute properties. **Add** and **Delete** icons are provided to add and delete an attribute property. Select an attribute and use the ✖ delete icon to delete it, or use the ✚ add icon to add an attribute.

You can change the flag property by a single mouse click. You can modify all the attribute properties in the property section. The size of attribute columns in property section is arranged to show more data.

Click the **Add Attribute Group** button if you want to create a new attribute group or add a predefined attribute group. Click the **Group Sequence** button if you want to reorder attribute groups.

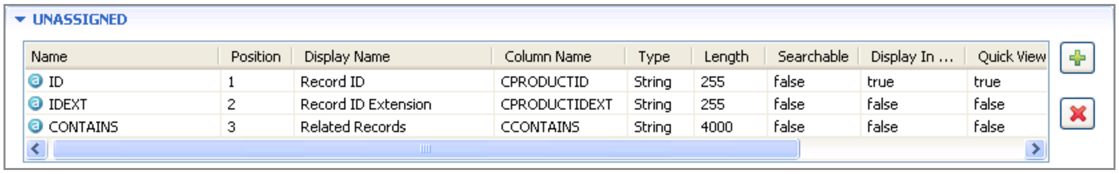
When you select an attribute group in the drawing pane, the Properties tab displays all attributes within that attribute group. You can see and add the General and Advanced information per attribute:
General Tab



operty	Value
Name	ModelNumber
Display Name	ModelNumber
Description	ModelNumber
Type	String
Length	
Column Name	
Association Map	NONE
Help	
Searchable	false
Display In Record List	false
Quick Viewable	false
Category Attribute	false
Multi Value	false
Use Common Table	false
Partition Key	false
Multi Value Table Name	
Position	

- **Name** - The name for the attribute (maximum of 30 characters, a-z, A-Z, 0-9 and _)
- **Display Name** - The display name for the attribute (maximum of 60 characters, less restrictive naming conventions compared to name). Duplicate display names are allowed.
- **Description** - Description for the attribute (maximum of 4000 characters).
- **Type** - The data type for the attribute (String, File, Amount, Boolean, Integer, Date, Decimal, Custom Decimal, Timestamp, Long, and URL). The Long data type is native data type which maps to Java Long. The maximum length is 19. The URL is a derived data type based on String. The maximum length is 4000 and default length is 256. If the length is zero or blank value, default length is assigned. The Custom Decimal is like the Decimal data type. The maximum length is 29. The custom decimal is indicated by the decimal length 29 minus the assigned length. For example, (18,11).
- **Length** - Attribute length
- **Column Name** - A unique database column name for the attribute. Name must comply with database requirements if database has validations defined for attribute column name.
- **Association Map** - Can choose any attribute as ID (Record Id), IDEXT (Record ID Extension), or EFFECTIVEDATE. The available mappings in the Association Mapping drop-down list are NONE, ID, IDEXT, and EFFECTIVEDATE.
- **Help**
- **Searchable** - Select to mark the attribute for display in search results.

- **Display In Record List** - Select this option to display the attribute in the Record List
- **Quick Viewable** - Select to include the attribute in the Quick view section of the Record Relationship tab.
- **Category Attribute** - Select this option to define the attribute as a Category Specific. When an attribute is defined as Category Specific, you can either enter a table name or use common tables. Category Specific attribute share the same predefined common tables used for multi-value attributes. In addition, the new predefined MV_ SHARED_BOOLEAN_TABLE table is available for the Boolean data type.
- **Multi Value** - Whether attribute is multi-value.
- **User Common Table** - Whether the multi-valued/category specific attribute uses predefined common table.
- **MultiValue Table Name** - The multi-value table name if the attribute is a multi-value attribute.
- **Partition Key** - Select to define the attribute as partition key.
- **Position** - The position for the attribute.



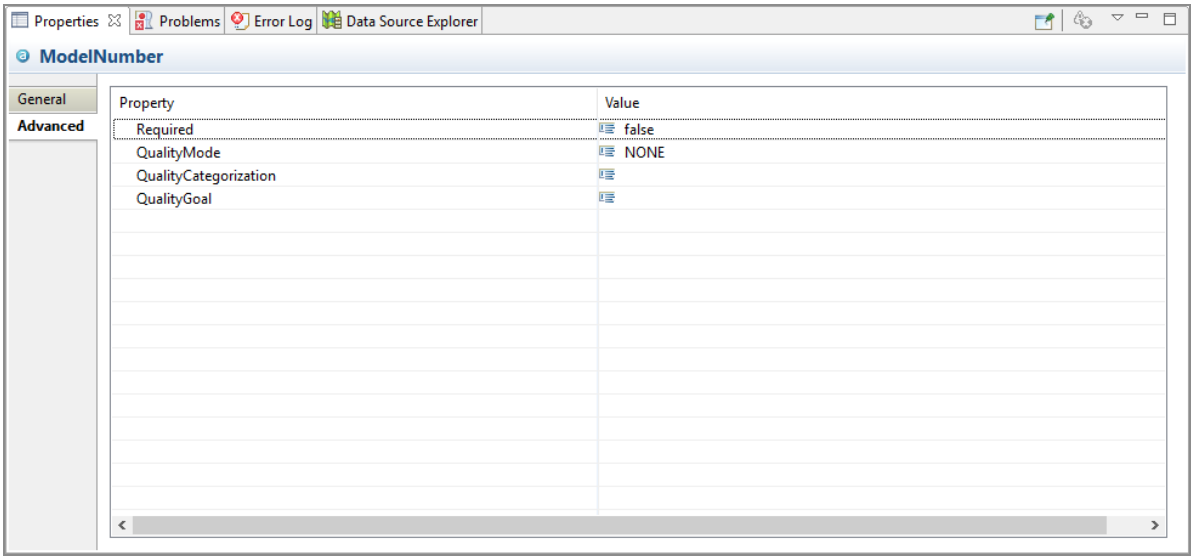
UNASSIGNED

Name	Position	Display Name	Column Name	Type	Length	Searchable	Display In ...	Quick View
ID	1	Record ID	CPRODUCTID	String	255	false	true	true
IDEXT	2	Record ID Extension	CPRODUCTIDEXT	String	255	false	false	false
CONTAINS	3	Related Records	CCONTAINS	String	4000	false	false	false

Advanced Tab

The **Advanced** tab displays the advanced properties for attribute. The default advanced properties are as follows:

1. Required
2. Quality Mode
3. Quality Categorization
4. Quality Goal



Property	Value
Required	false
QualityMode	NONE
QualityCategorization	
QualityGoal	

In addition to the existing properties, you can define custom properties for attributes by using the **Configurations** > `attributeAdvancedProperties.properties` file of the studio project. After adding the custom property, the newly added custom property appears in the **Advanced** tab in the properties section.

- **Required** - Specify if you want attribute as mandatory attribute.
- **QualityMode** - Displays the selected quality mode. The available options are:
  - **NONE** : indicates that the quality mode is not selected.
  - **DEFINED** : indicates that the attribute has a value. That means, the value should not be empty.
  - **VALIDATED** : indicates that a rulebase constraint has been associated with the validation of the attribute.
  - **VERIFIED** : indicates the verification of an attribute, that is, external validation. For example, address verification and email verification.
- **QualityCategorization** - attribute name in which the quality attribute is categorized.
- **QualityGoal** - indicates the quality value of an attribute.

To add the new custom attribute property, open the `attributeAdvancedProperties.properties` file, enter the new property as `Propertyname=DataType#Value`. If you have more than one values, enter the values separated by semi colon. The values appear as a drop-down list. Only primitive data types like string, integer, double, boolean are supported. For example,

- `isEmployee=Boolean#True,False`

- `Address=String#PaloAlto`

> **Note:** The new custom attribute property to be effective, you must restart TIBCO MDM Studio.

Defining Relationships (Step 4)

A Relationship represents a connection between two repositories. You can define two types of relationships:

- **Self relationships** where relationships are defined between records in the same repository.
- **Cross repository relationships** which are defined across records in different repositories.

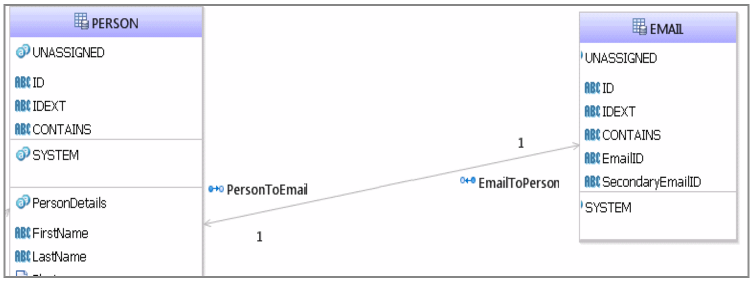
A - Creating Relationship(s)

To create a relationship:

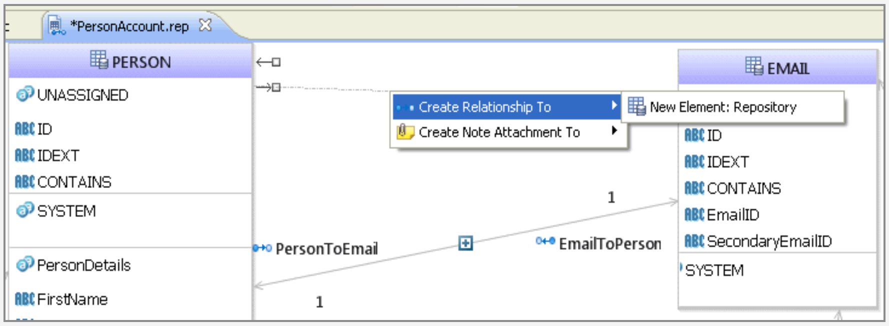
Procedure

1. Click the Relationship icon in the Palette and then click the edge or the header of the source Repository (that you want to create the relationship from).
2. Drag the mouse towards the target repository and release the mouse button.
3. Enter a relationship name and reverse relationship name. Appropriate icons (arrows) denote the forward and reverse relationships on the relationship connector.

Result

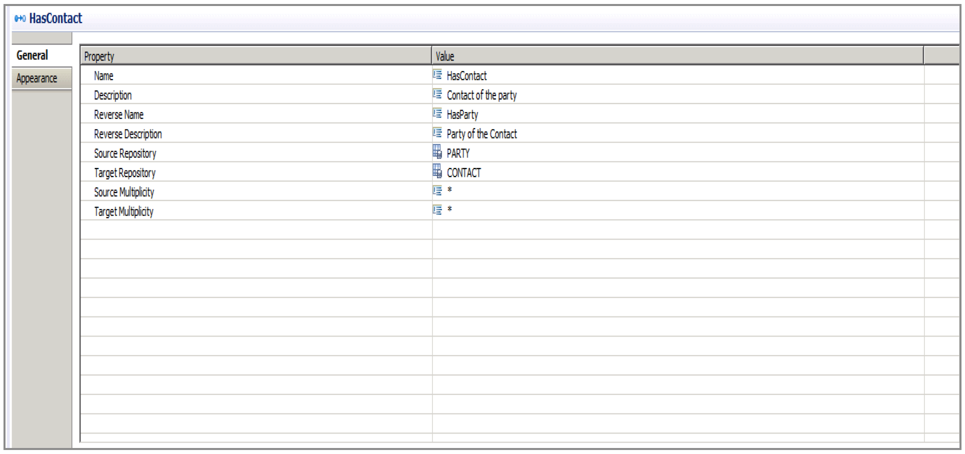


Note: If you attempt to create a relationship when you have only one repository or if you only select a source but not a target repository, you get an option to automatically create a second or new repository.



Relationship Properties

If you select a Relationship in the drawing pane, its properties are displayed in the **Properties** pane. This is a representation of the MDM Relationship Definition table. It displays Relationship properties such as the Name, Description, Reverse Name, Reverse Description, Source Repository and so on.



Property	Value
Name	HasContact
Description	Contact of the party
Reverse Name	HasParty
Reverse Description	Party of the Contact
Source Repository	PARTY
Target Repository	CONTACT
Source Multiplicity	*
Target Multiplicity	*

B- Creating Relationship Attribute Group(s)

A Relationship Attribute Group serves as a container for one or more Relationship Attributes.

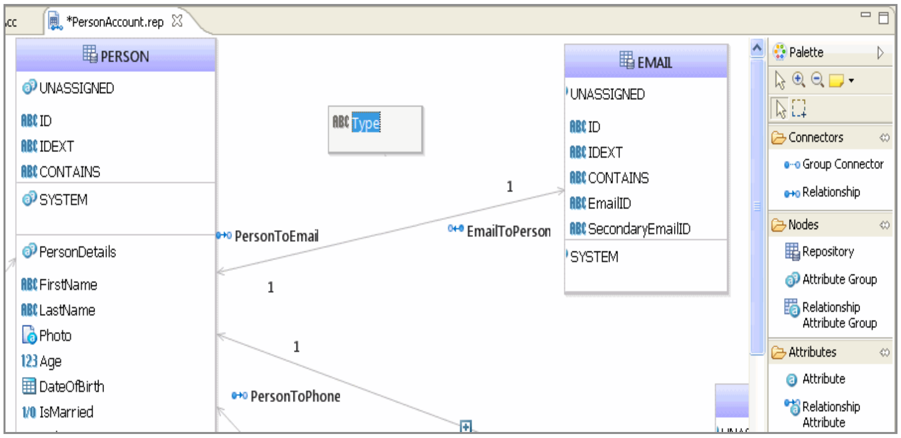
You need to first define a Relationship Attribute Group before you define Relationship Attributes (see C - Defining Relationship Attributes).

To define a Relationship Attribute Group:

Procedure

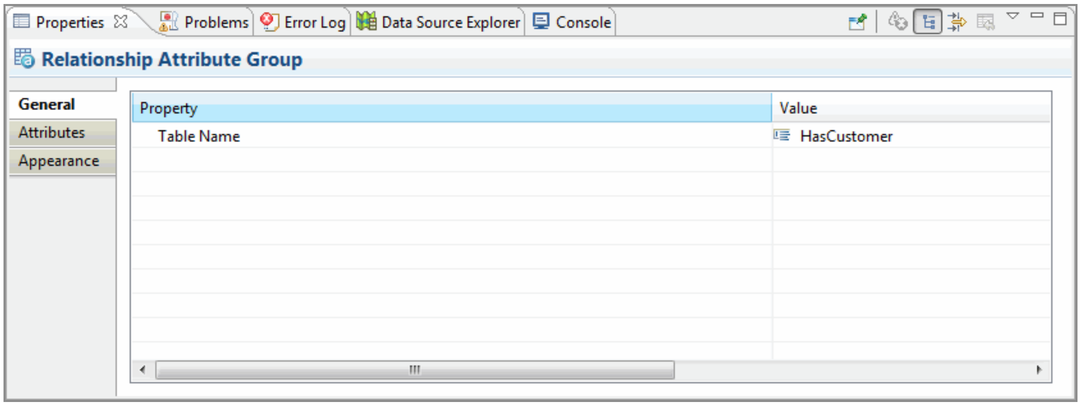
1. Select the Relationship Attribute Group icon in the palette.
2. Click in an area outside of the repositories, close to the desired relationship. A gray box is added, this is the Relationship Attribute Group.

Result



Relationship Attribute Group Property

The General tab displays relationship attribute group property Table Name.



Property	Value
Table Name	HasCustomer

C - Defining Relationship Attributes

When a relationship is defined, one or more attributes can be defined for the relationship (similar to repository attributes).

You need to first define a Relationship Attribute Group before you define relationship attributes (see B- Creating Relationship Attribute Group(s)).

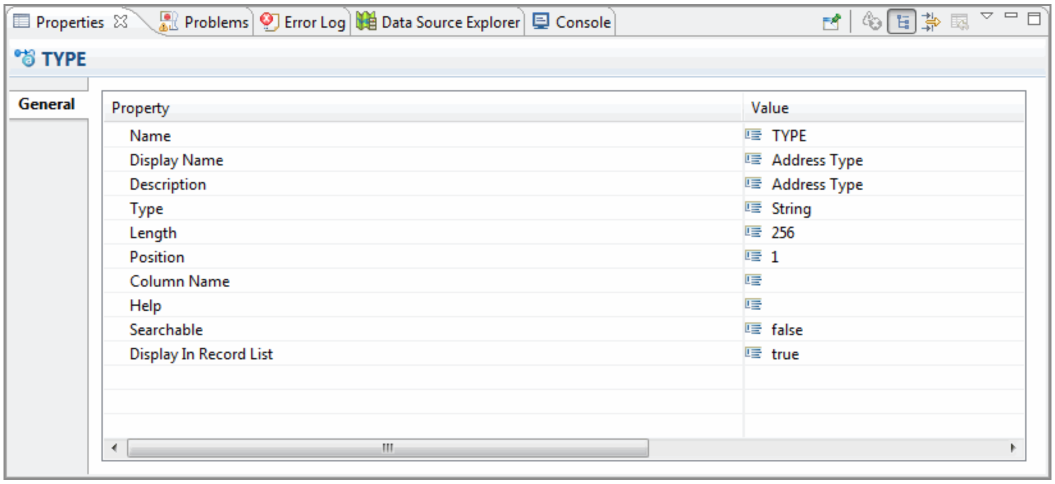
To create a relationship attribute:

Procedure

1. Click the Relationship Attribute icon in the Palette and then click in a (previously defined) Relationship Attribute Group.
2. Enter the Relationship Attribute name.
3. Relationship Attribute properties can be changed by clicking the relationship attribute and making edits in the **Properties** window **General** tab or by selecting the Relationship Attribute Group and making edits in the **Properties** Window **Attributes** tab.

Result

Note: Relationship attributes cannot be defined as Category Specific Attributes.

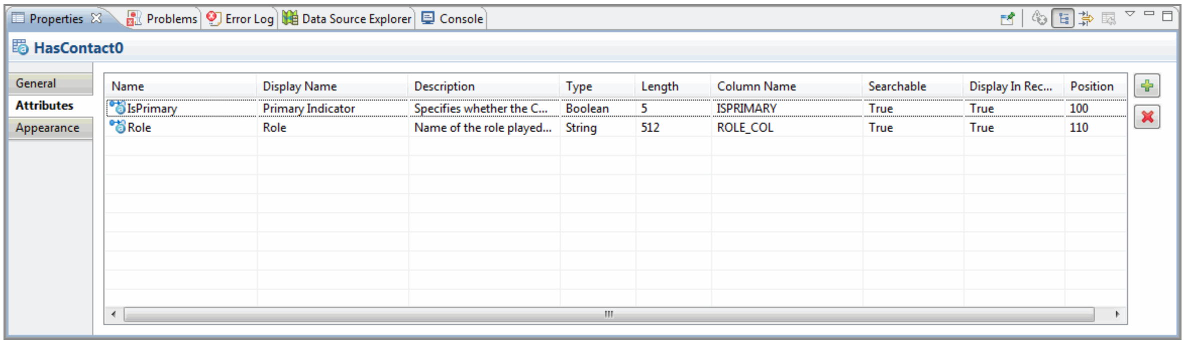


Property	Value
Name	TYPE
Display Name	Address Type
Description	Address Type
Type	String
Length	256
Position	1
Column Name	
Help	
Searchable	false
Display In Record List	true

Relationship Attribute Properties

If you select a Relationship Attribute in the drawing pane, its properties are displayed in the **Properties** pane, **General** tab.

If you select the entire Relationship Attribute Group, details of all the contained relationship attributes are displayed in the **Properties** pane, **Attributes** Tab.



Name	Display Name	Description	Type	Length	Column Name	Searchable	Display In Rec...	Position
IsPrimary	Primary Indicator	Specifies whether the C...	Boolean	5	ISPRIMARY	True	True	100
Role	Role	Name of the role played...	String	512	ROLE_COL	True	True	110

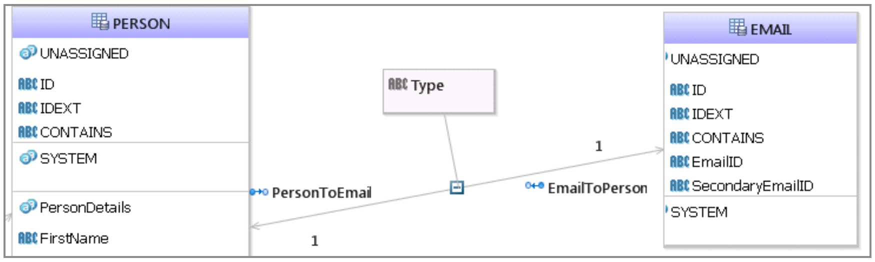
D - Connecting the Relationship Attribute Group to the Relationship

Once you define Relationship Attributes within a Relationship Attribute Group, you need to connect the Relationship Attribute Group to the Relationship. To do this:

Procedure

1. Select the **Group Connector** icon under Connectors in the Palette.
2. Make a connection from the relationship attribute group to the relationship by clicking the edge of the relationship attribute group box and dragging up to the relationship line.

Result



Relationship Attribute Groups (and their Attributes) have a collapsible display.

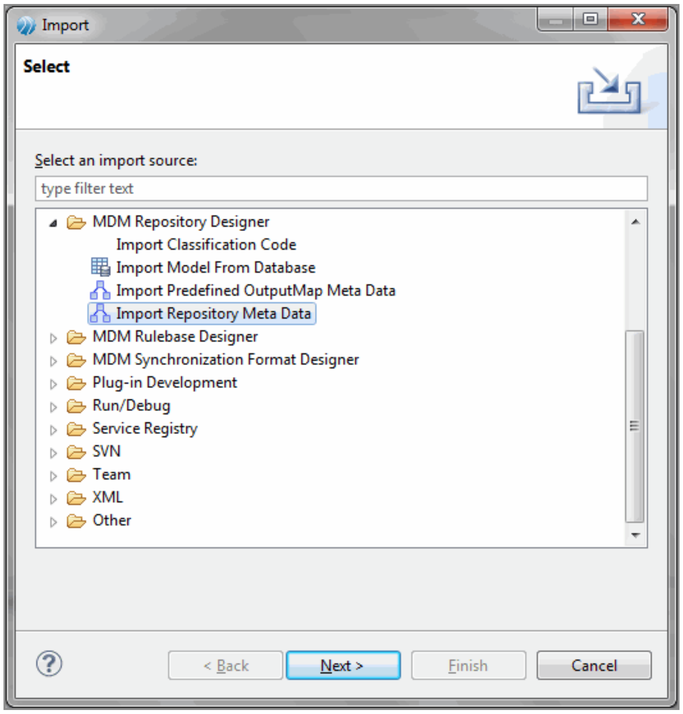
Once you successfully connect a **relationship attribute group** to a Relationship, you can see a small icon on the relationship which shows as a minus when the relationship attribute group and its attributes are maximized or as a plus when minimized.

Importing your TIBCO MDM Repository

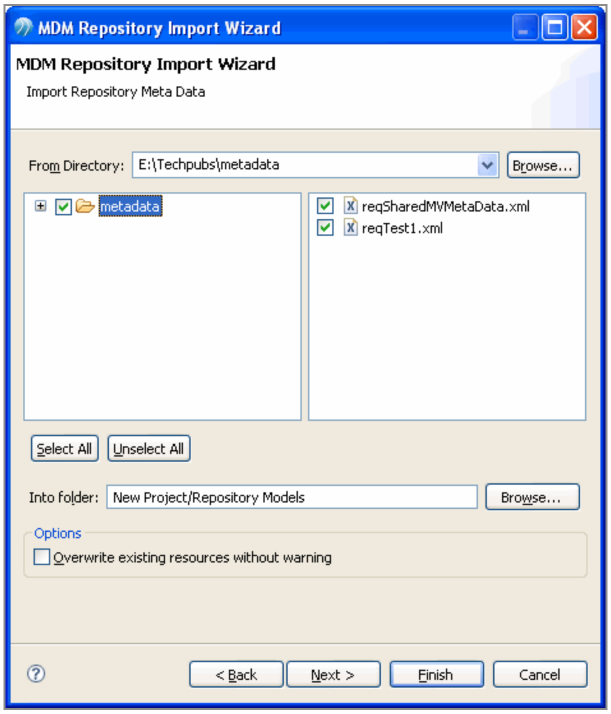
Note: This section presumes that you have defined a repository in TIBCO MDM (with or without artifacts such as attributes, relationships and so on), exported it by inputting the request XML (with the repository name) through the Export Metadata UI in TIBCO MDM, and taken the JAR file generated (containing the meta data XML).

Procedure

1. Create a Project file (See Creating a new Project to hold your Repository Model (Step 1)), right click the **Repository Models** folder and select **Import**.
2. Select **Import Repository Meta Data** under **MDM Repository Designer**. Click **Next**.



3. Browse and select the location where the JAR file has been extracted and ensure the associated XML file is displayed on the right.


MDM Repository Import Wizard
MDM Repository Import Wizard
Import Repository Meta Data
Browse...
Select All
Unselect All
Browse...
Options
< Back
Next >
Finish
Cancel

From Directory: E:\Techpubs\metadata

☑ metadata

☑ reqSharedMVMetaData.xml

☑ reqTest1.xml

Into folder: New Project/Repository Models

☐ Overwrite existing resources without warning

4. Click **Finish**.

Result

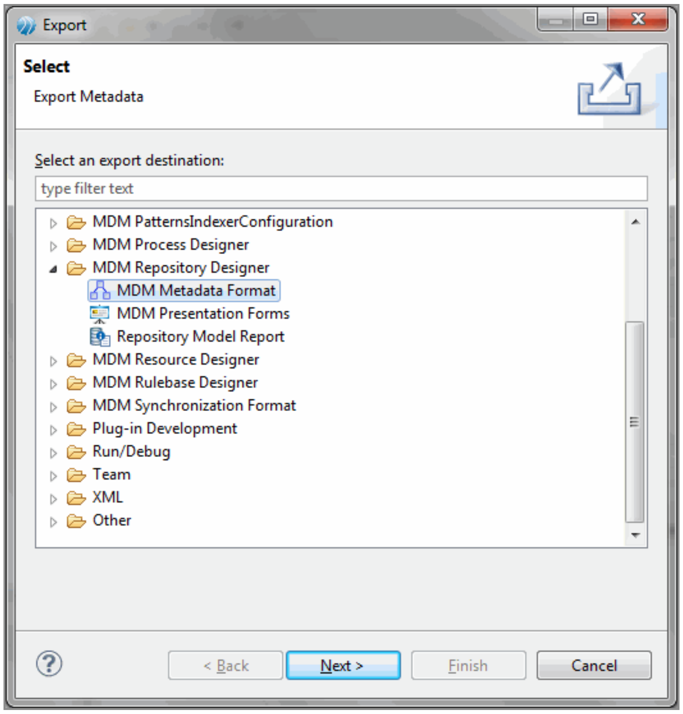
Your imported repository should now show in your project along with any related artifacts.

Exporting a Repository Model

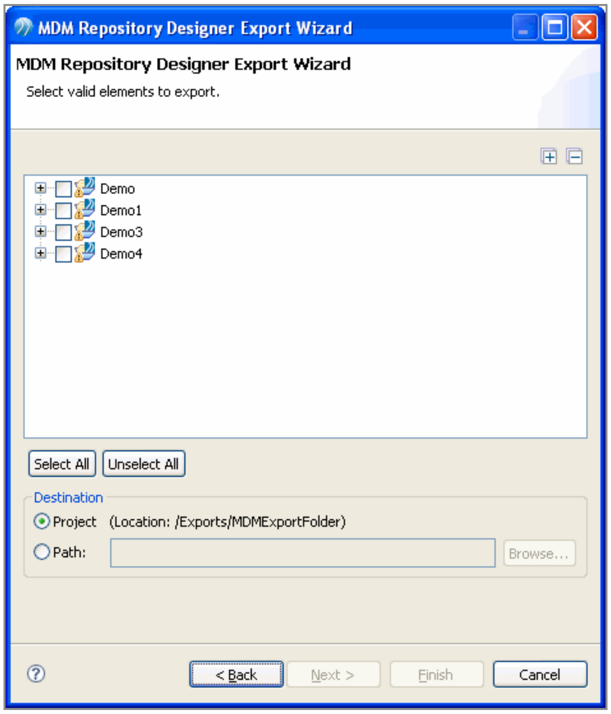
The Export Wizard is used to export the graphically designed repository model to TIBCO MDM in .xml file format.

Procedure

1. Select the Project in the Project Explorer and click **File- > Export** or right click the project and select **Export**.
2. Select **MDM Metadata Format** under **MDM Repository Designer**. Click **Next**.



3. Select the <repositorymodelName>.rep file for export by selecting the checkbox. The default location to which the file gets exported is displayed in the **Destination** section under Project (/Exports/MDMExportFolder). You can change the path if required. Click **Finish**.



MDM Repository Designer Export Wizard

MDM Repository Designer Export Wizard

Select valid elements to export.

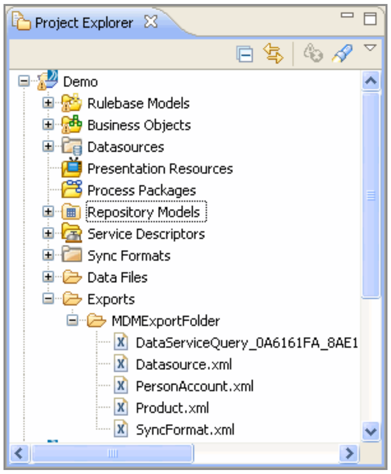
- □ Demo
- □ Demo1
- □ Demo3
- □ Demo4

Destination

● Project (Location: /Exports/MDMExportFolder)

○ Path:

If you chose to export to the Project itself, in the Project Explorer, you can now see a new folder Exports/MDMExportsFolder created which contains the XML file that was generated.



```
Demo
  Rulebase Models
  Business Objects
  Datasources
  Presentation Resources
  Process Packages
  Repository Models
  Service Descriptors
  Sync Formats
  Data Files
  Exports
    MDMExportFolder
      DataServiceQuery_0A6161FA_8AE1
      Datasource.xml
      PersonAccount.xml
      Product.xml
      SyncFormat.xml
```

Result

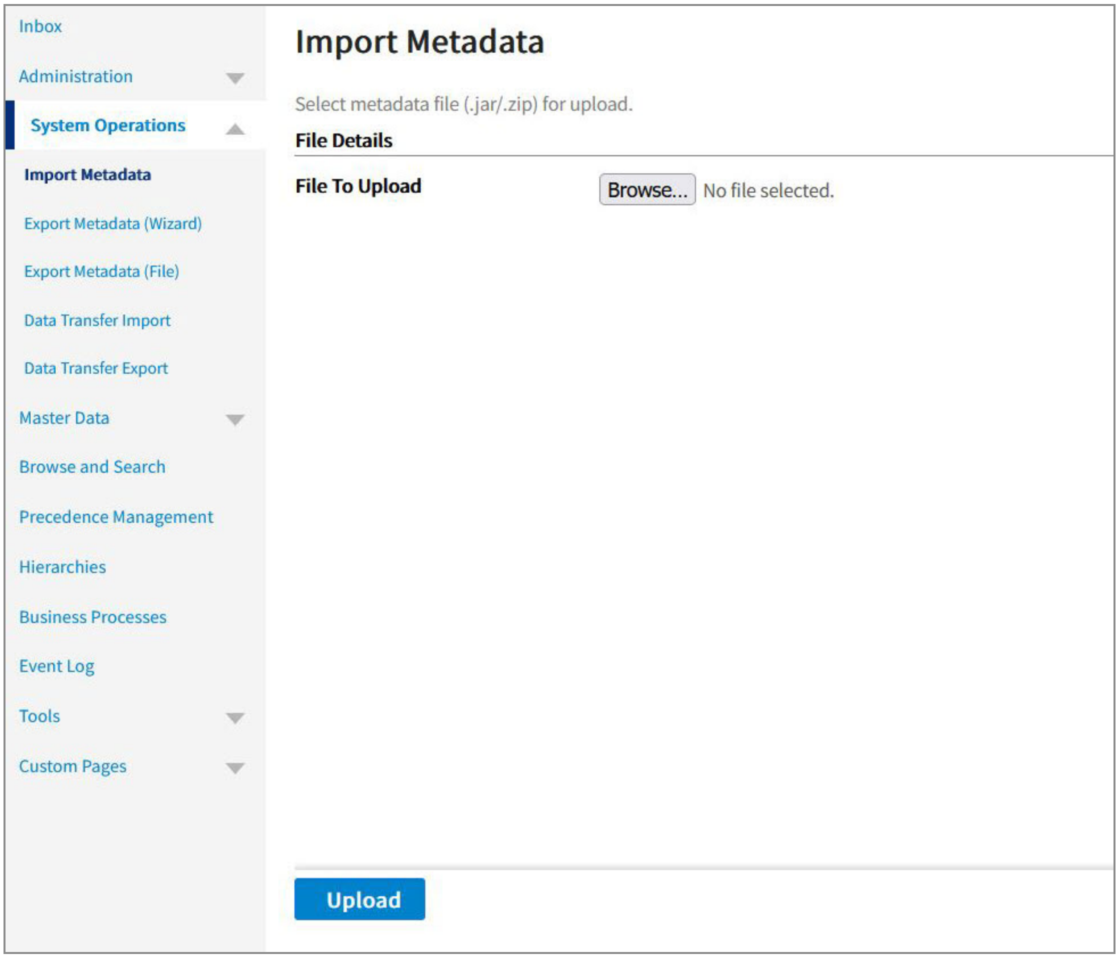
You can now import your Repository model into TIBCO MDM.

Importing a Repository Model

Once you export the Repository model (XML file), you can import it into TIBCO MDM.

Procedure

1. Start the TIBCO MDM application.
2. Go to **System Operations-Import Meta Data**. Browse to select the `.xml` file exported from the Repository Designer. Click **Upload**.



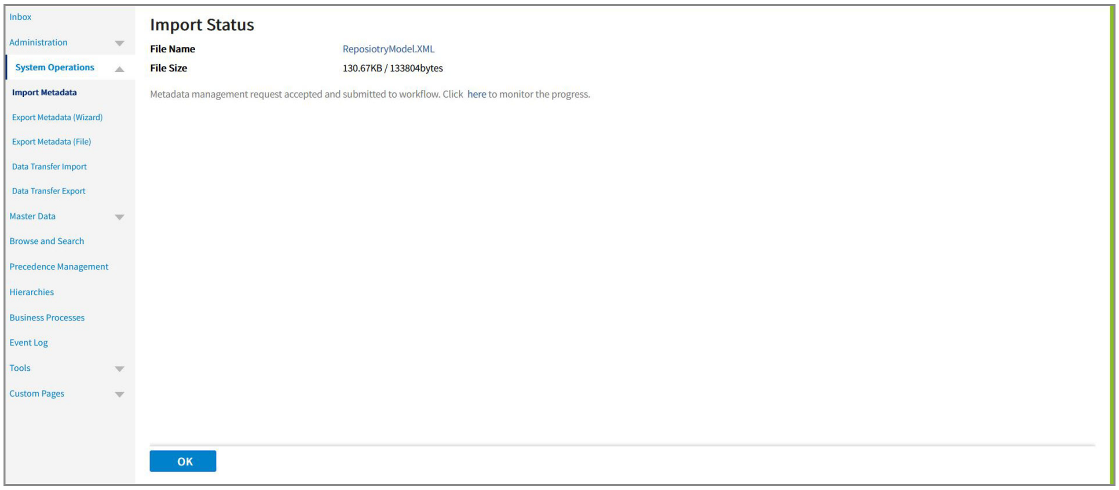
Import Metadata

Select metadata file (.jar/.zip) for upload.

File Details

File To Upload: Browse... No file selected.

3. You can see details of the file you selected for upload. You can click the **Check Progress** link to see the status.



Import Status

File Name ReposiotryModel.XML

File Size 130.67KB / 133804bytes

Metadata management request accepted and submitted to workflow. Click here to monitor the progress.

4. You can check the status of the process from the Event Log. If the Event shows as complete without any error, it means the repository model was successfully imported in TIBCO MDM.
5. Go to the **Repositories** page to check if your repositories have got added.
6. You can also see any relationships that you created between your Repositories in the Repository Designer. For instance, if you created Repository 1 and Repository 2 with a relationship between them you can see that in the TIBCO MDM UI.

Appearance Tab

You can change the appearance of the repository model elements. The Appearance tab provides different icons using which you can change the font, color and styles of the repository, attribute group, relationship attribute group, and the relationships.

Appearance Tab For Repository

You can change the appearance of the Repository. Use the following icons to change the appearance.

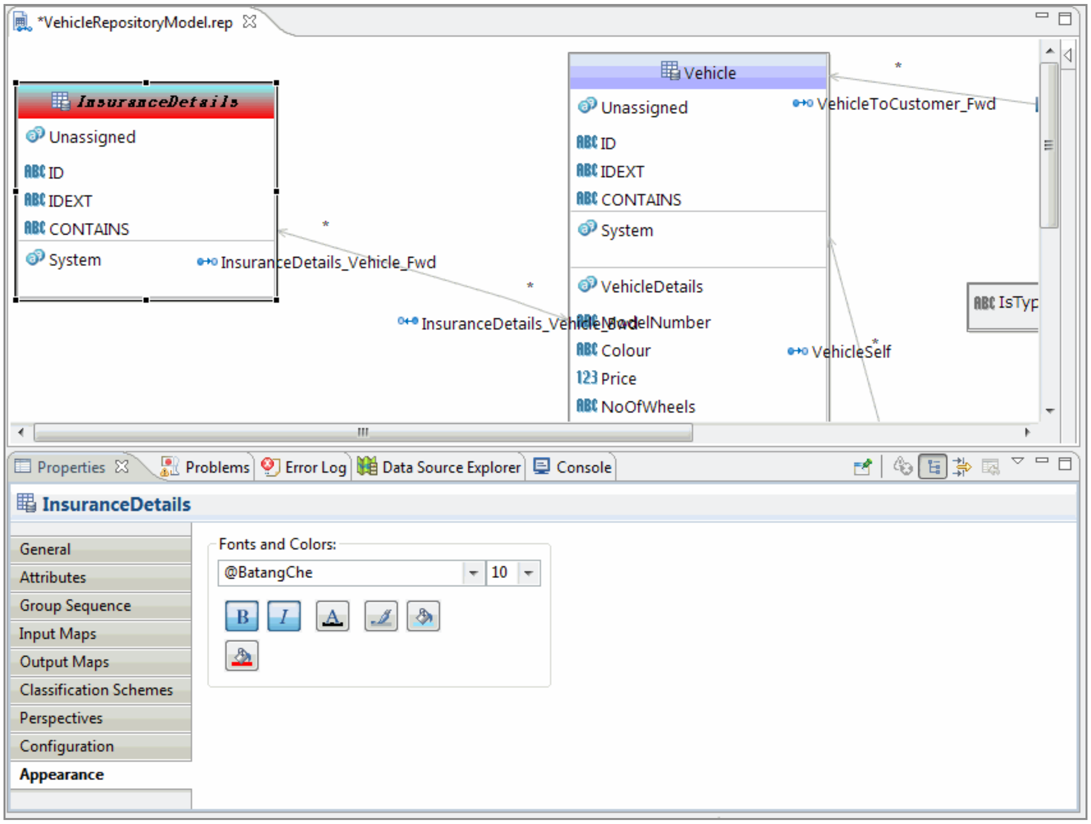
Fonts and Colors	Descriptions
B	To make bold style font of the repository header.
I	To make italic style of the repository header.
A	To change the color for the font name of the repository header.
✎	To change the line color of the repository header.
🪣	To change the upper gradient of the repository header.
🪣	To change the bottom gradient of the repository header.

Procedure

1. Select the repository for which you want to change the appearance in the editor and click the **Appearance** property tab in property section.

2. The **Fonts and Colors** group is displayed. Select the new font and size for the repository name.

3. To change the style click the respective icons. To change the colors for repository header click respective icon.



Fonts and Colors: @BatangChe 10

4. The repository with the new appearance is displayed in the editor.

Appearance Tab For Attribute Group

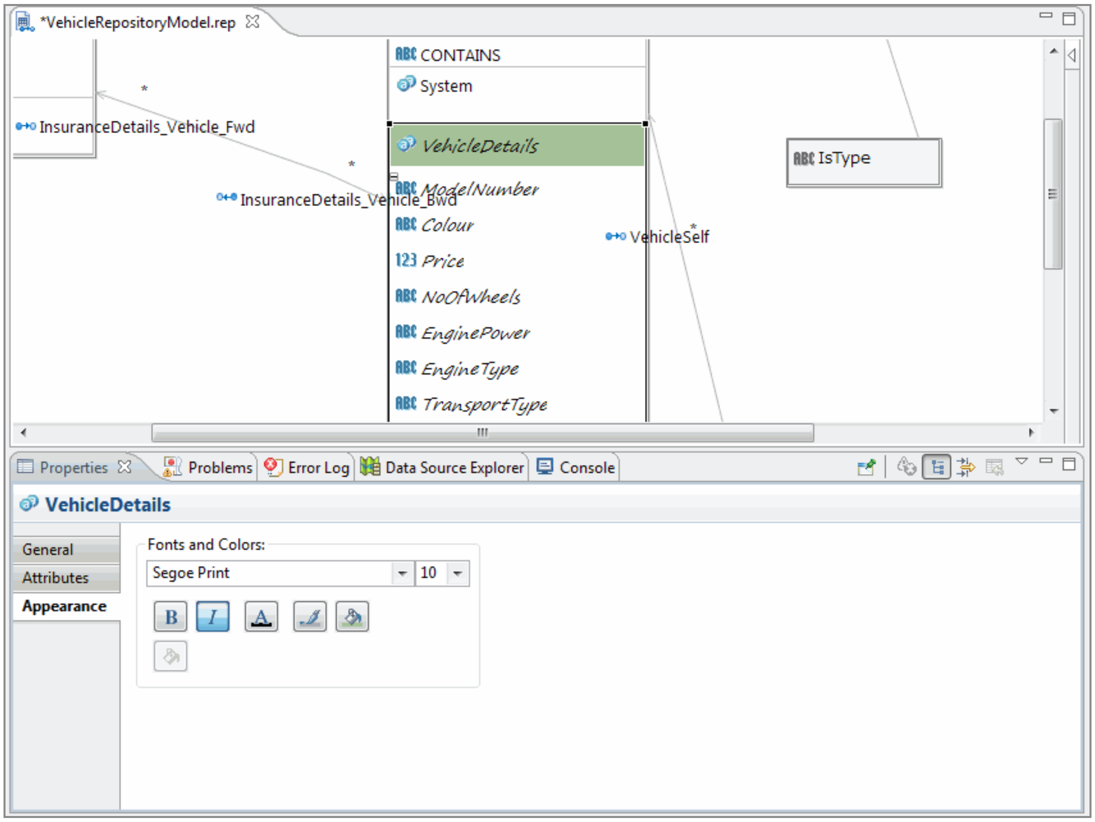
You can change the appearance of the repository attribute group. Use the following icons to change the appearance.

Fonts and Colors	Descriptions
B	To make bold style font for attribute groups and attributes in that group.

I	To make it italic style for attribute groups and attributes in that group
A	To change font color for attribute groups and attributes in that group
✎	To change the line color for attribute group.
🪣	To change the header color of the attribute group.

Procedure

1. Select the repository attribute group for which you want to change the appearance in the editor and click on the **Appearance** property tab in property section.
2. The **Fonts and Colors** group is displayed. Select the new font and size for the attribute group.
3. To change the style click on the respective icons. To change the colors for attribute group header click on respective icon.



4. The repository attribute group with the new appearance is displayed in the editor.

Appearance Tab For Relationship Attribute Groups

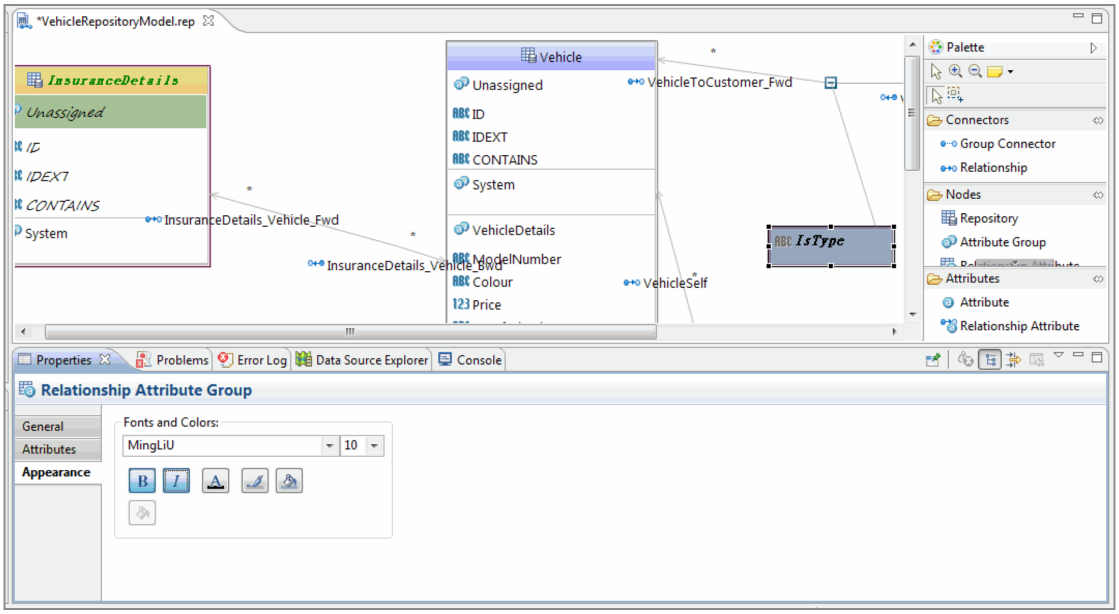
You can change the appearance of the relationship attribute group. Use the following icons to change the appearance.

Fonts and Colors	Descriptions
B	To make bold style font for relationship attributes in that group
I	To make it italic style for relationship attributes in that group

A	To change font color for relationship attributes in that group
✎	To change the line color for relationship attribute group.
🪣	To change the color for the relationship attribute group

Procedure

1. Select the relationship attribute group for which you want to change the appearance in the editor and click on the **Appearance** property tab in property section.
2. The **Fonts and Colors** group is displayed. Select the new font and size for the relationship attribute group.
3. To change the style click on the respective icons. To change the colors for relationship attribute group click on respective icon.



Fonts and Colors: MingLiU 10

4. The relationship attribute group with the new appearance is displayed in the editor.

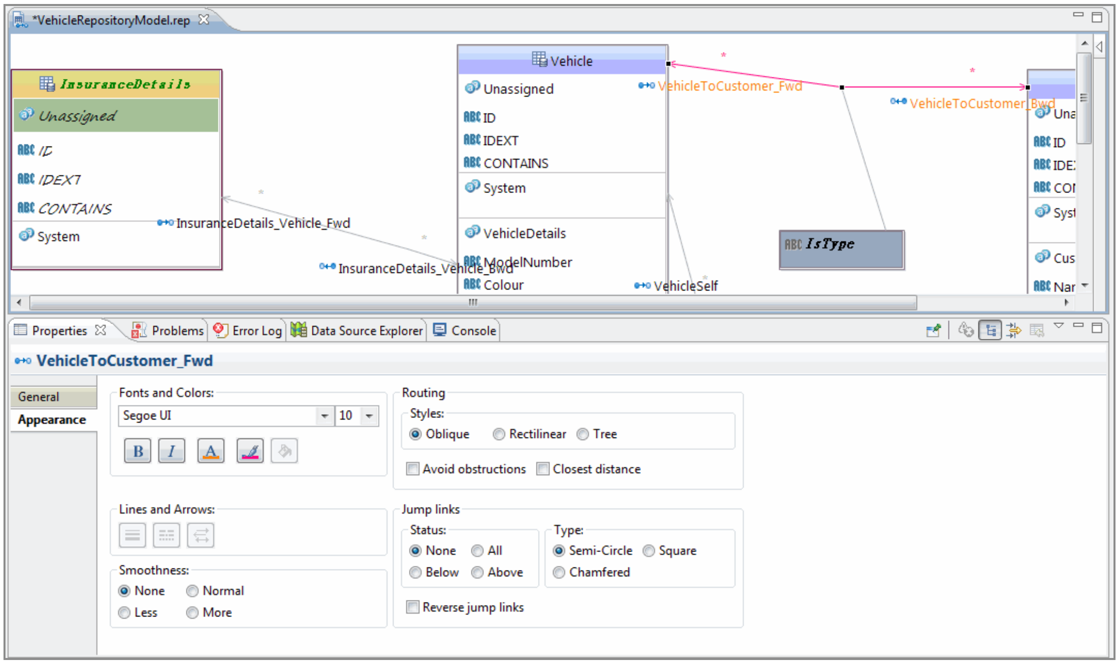
Appearance Tab For Relationship

You can change the appearance of the relationship. Use the following icons to change the appearance

Fonts and Colors	
B	To make bold style font for relationship.
I	To make it italic style for relationship.
A	To change font color for relationship.
✎	To change the line color for relationship.

Procedure

1. Select the relationship for which you want to change the appearance in the editor and click on the **Appearance** property tab in property section.
2. The **Fonts and Colors** group is displayed. Select the new font and size for the relationship.
3. To change the style click on the respective buttons. To change the colors for relationship click on respective icon.



VehicleToCustomer_Fwd

Fonts and Colors: Segoe UI 10

Routing

Styles: ● Oblique ○ Rectilinear ○ Tree

□ Avoid obstructions □ Closest distance

Smoothness: ● None ○ Normal ○ Less ○ More

Jump links

Status: ● None ○ All ○ Below ○ Above

Type: ● Semi-Circle ○ Square ○ Chamfered

□ Reverse jump links

4. The relationship with the new appearance is displayed in the editor.

Data Source Editor

This chapter introduces you to the concept of Data Sources in TIBCO MDM Studio and explains how to use this feature.

Data Sources Overview

Data sources are used to import data into repositories, identify records to create subsets and transform data. Data sources model external sources of information, such as any file that you want to upload.

Data sources can be used to:

- Imports record into repositories.
- List to define a subset.
- List of valid values for validation and cleansing.
- Input to data validation rules and so on.

The Data Source Designer is based on TIBCO Business Studio and integrated in Repository Designer. The Data Source Designer provide TIBCO MDM specific Data Source editor through which you can design and edit your data source model. Data Sources are in .ds format, contained in a special folder in TIBCO MDM Studio called "Datasources".

Data Source Properties

The Data Source Properties are as follows:

- **Name**: The Data Source name must be unique and can contain A-Z, 0-9, all special characters (except a space), and characters from other languages too.
- **Description**: The Data Source Description.
- **Delimiter Character**: The delimiter character can be a Comma, Pipe, Semicolon, Colon, Space, Tab or Other character chosen by the user.

- **Date Format**: The date format must match the actual date format. If specified incorrectly, data may be rejected or loaded incorrectly. For example, if the date format is specified as MM/DD/YY, and the actual data is specified in DD/MM/YY, then 02/01/12 may be imported as Feb-01-2012 instead of Jan-02-20120.
- **Text Qualifier Character**: The text qualifies can be a double quote, Single quote, or any other qualifies chosen by the user.
- **Use Column Names**: If selected, the column names are the first row of the data in the file you are uploading. If not selected, you have to create the attribute names manually while creating the data source.
- **Start Import at Row Number**: The row number from which the application starts importing data into the repository.
- **Format**:
    - Delimited
    - Fixed Length
    - Database
- **Table Name**: If you have selected the Database format, the **Table Name** field appears. The table name must be available in the TIBCO MDM database. You cannot connect to a different database.
- **Data File**: Click the **...** button next to the Data File field to navigate to and select the data source file. After you select the file, the selected filename appears.
- **Timestamp Format**: Select the timestamp format in the **Timestamp Format** drop-down list. By default, YYYY-MM-DD HH:mm:ss timestamp format is displayed. If the actual data contains a different format, you can change it. The timestamp format must match the timestamp format used in the actual data. If you specify incorrect timestamp format, data may be rejected or loaded incorrectly.

What you can do with the Data Source Designer

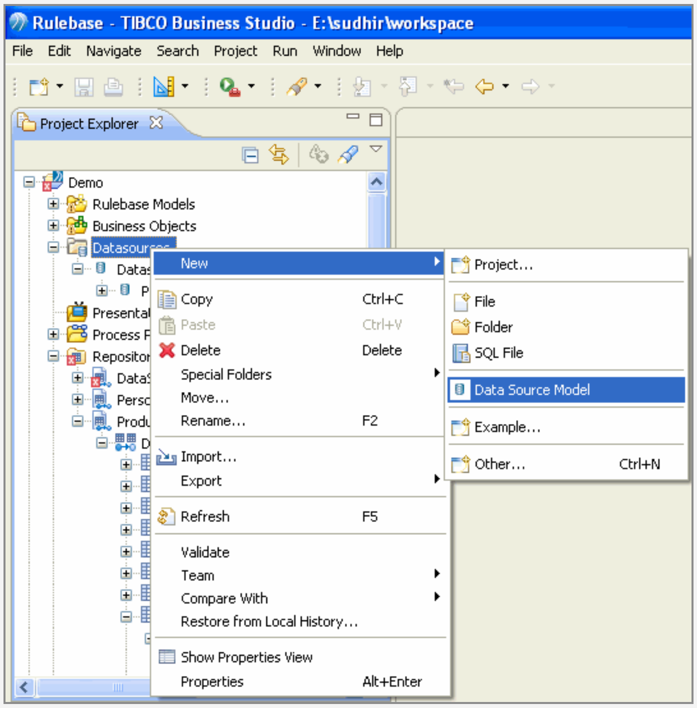
With the Data Source Designer, you can create data sources, modify them, customize attributes, and export the final data source model to TIBCO MDM. You can also import multiple Data Source metadata exported from TIBCO MDM into the Data Source Designer for further edits or you can use it to define Input map using Input Map Editor.

Creating a New Data Source Model

Procedure

1. Right-click on the **Datasources** folder in the Project Explorer and select **New - > Data Source Model.**

   Note:

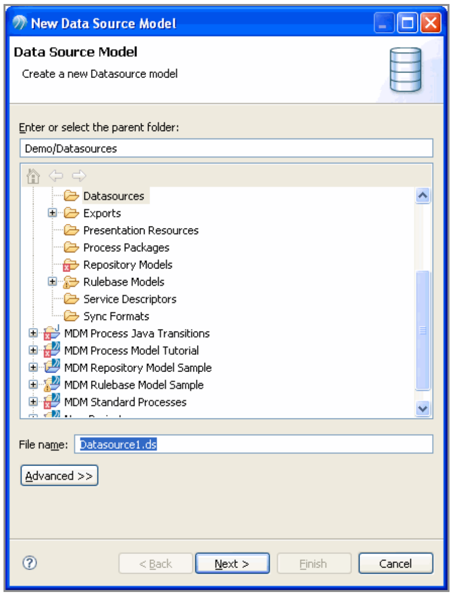
   

   The Data Sources special folder can be created if it is not available in project.

   To create new Data Source special folder do the following:

   - Right click on project, Select **New- > Folder**
   - Provide a name for the folder and Click **Finish**
   - Right click on newly created folder and Select **Special Folders - > Use as Datasource Folder**.

2. The New Data Source Wizard is displayed.



New Data Source Model

Data Source Model

Create a new Datasource model

Enter or select the parent folder:

Demo/Datasources

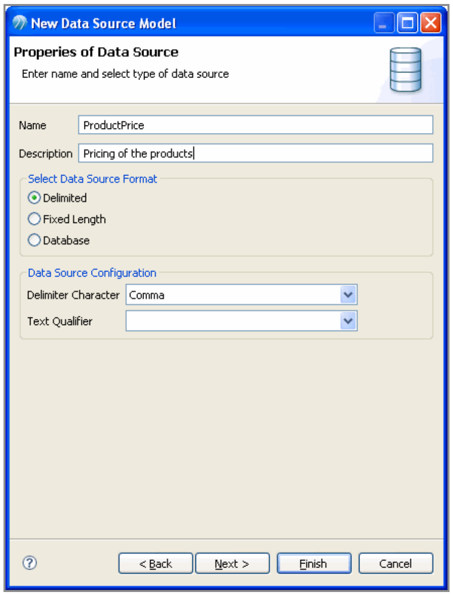
- Datasources
- Exports
- Presentation Resources
- Process Packages
- Repository Models
- Rulebase Models
- Service Descriptors
- Sync Formats
- MDM Process Java Transitions
- MDM Process Model Tutorial
- MDM Repository Model Sample
- MDM Rulebase Model Sample
- MDM Standard Processes

File name: Datasource1.ds

3. Specify a name for the Data Source Model file in the **File Name** field.

4. Click **Next**.

5. The data source property page is displayed.



New Data Source Model

Properies of Data Source

Enter name and select type of data source

Name: ProductPrice

Description: Pricing of the products

Select Data Source Format

● Delimited

○ Fixed Length

○ Database

Data Source Configuration

Delimiter Character: Comma

Text Qualifier:

6. Enter the Data Source Name and Description in the **Name** and **Description** fields.

7. Select the **Data Source Format**. The available options are:

   - Delimited

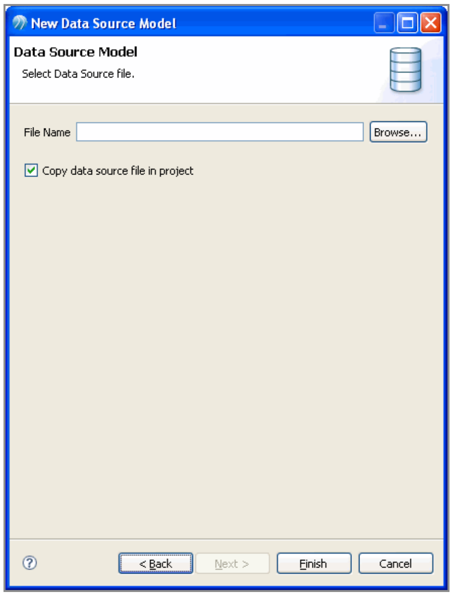
- Fixed Length
- Database

  The rest of the fields vary depending on the format selected. For example, if you have selected the **Database** format, only **Name**, and **Description** fields are displayed.

8. **Delimiter Character**. Select the appropriate delimiter character from the drop-down. The delimiter character can be a Comma, Pipe, Semicolon, Colon, Space, Tab, or Other character.
9. **Text Qualifier**. Select the appropriate Text Qualifier from the drop-down. The text qualifies can be a double quote, Single quote, or any other qualifier.
10. Click **Next**.

> **Note:**
>
> - Data field may contain Delimiter character. In such case data field should be enclosed with Text Qualifier.
> - The backslash (\) is not supported as delimiter.

11. The data source file selection page is displayed.



Data Source Model

Select Data Source file.

File Name

☑ Copy data source file in project

12. Select the **Copy data source file in the project** checkbox to include the data files in

the project. Browse to the folder where the data source file is saved and select the data file. Click **Finish**.

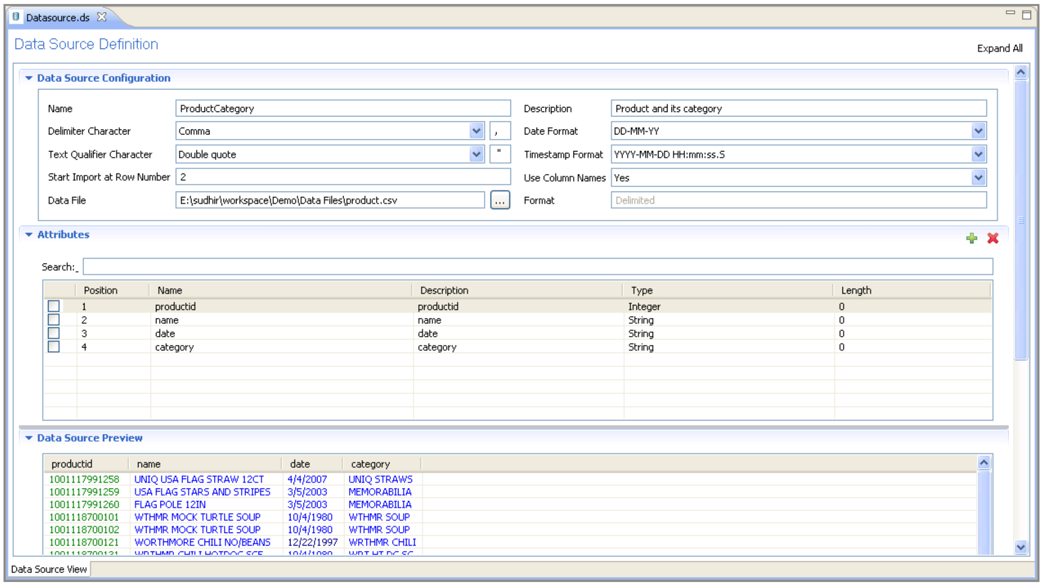
> **Note:** You can create a data source without selecting the **Copy data source file in the project** checkbox. The data source is created without the data file in the project folder.

13. When Data Source Format selected is **Fixed Length**. The **Data Source Attributes** section is displayed. Create the Data Source attributes for the Fixed Length Data Source.

Modifying an Existing Data Source

Using the Data Source Editor you can modify the existing Data Source in the project. The Data Source Editor has three sections.

- **Data Source Configuration**: Here you can modify the basic details, such as name of the data source, description, delimiter and other details. If you have given any cross reference to the data source then the Data Source Configuration will appear in read only mode.
- **Attributes**: All the attributes of data source can be viewed and modified in this section. Click ➕ to add new attributes. Attribute names cannot be more than 28 characters and must contain only ASCII (A-Z, a-z, 0-9, _). Attribute Position, Type and Length are set after finishing data source creation wizard. Delimited Data Source attribute data type is predicted on data source file column data. Data Type can be further changed from the Type column in attribute table. To delete an attribute select the checkbox corresponding to the attribute and Click ✖.
- **Data Source Preview**: Here you can preview the initial few lines of the Data Source files. Color scheme is imposed on data according to data type in Data Source preview. The following color is used:
  - Blue : String,
  - Green: Integer
  - Magenta: Boolean
  - Red: Error in data



Data Source Definition

Data Source Configuration

Name	ProductCategory	Description	Product and its category
Delimiter Character	Comma ; ,	Date Format	DD-MM-YY
Text Qualifier Character	Double quote ; "	Timestamp Format	YYYY-MM-DD HH:mm:ss.S
Start Import at Row Number	2	Use Column Names	Yes
Data File	E:\sudhir\workspace\Demo\Data Files\product.csv	Format	Delimited

Attributes

Search:_

	Position	Name	Description	Type	Length
□	1	productid	productid	Integer	0
□	2	name	name	String	0
□	3	date	date	String	0
□	4	category	category	String	0

Data Source Preview

productid	name	date	category
1001117991258	UNIQ USA FLAG STRAW 12CT	4/4/2007	UNIQ STRAWS
1001117991259	USA FLAG STARS AND STRIPES	3/5/2003	MEMORABILIA
1001117991260	FLAG POLE 12IN	3/5/2003	MEMORABILIA
1001118700101	WTHMR MOCK TURTLE SOUP	10/4/1980	WTHMR SOUP
1001118700102	WTHMR MOCK TURTLE SOUP	10/4/1980	WTHMR SOUP
1001118700121	WORTHMORE CHILI NO/BEANS	12/22/1997	WRTHMR CHILI
1001118700131	WRTHMR CHILI HOTDOG SCE	10/4/1980	WRT HT DG SC

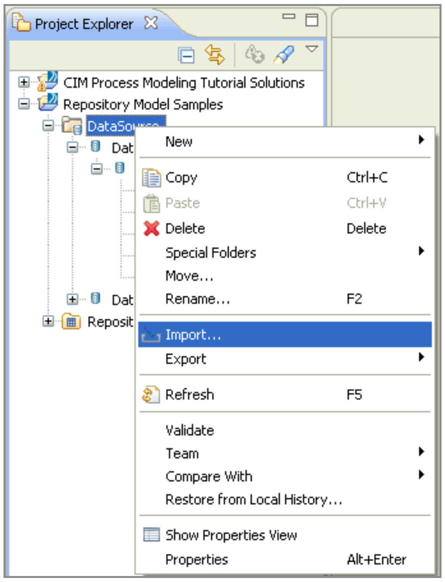
> **Note:** Data preview line can be configured from **Preview Lines** preference property (Default value is 50). Preview line property can be accessed from **Studio menu Windows->Preferences->Repository Designer->Data Source Editor.**

Importing an Existing Data Source Model

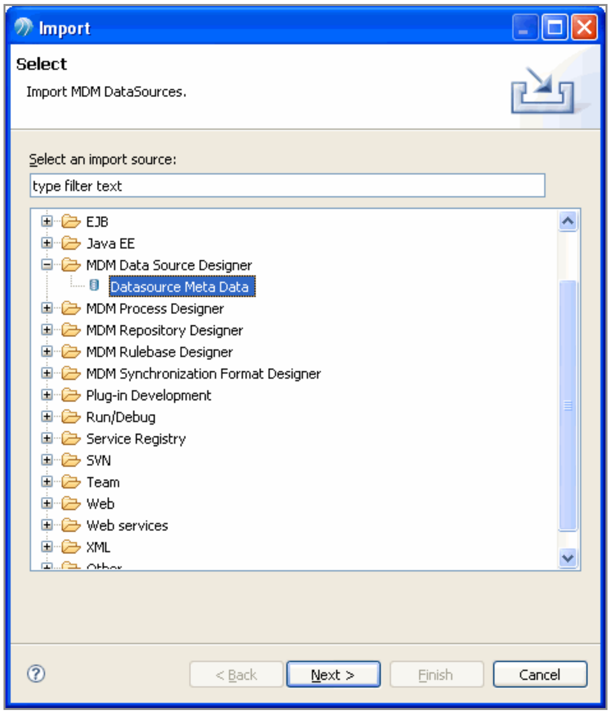
Using the Import Wizard you can Import Data Source model into TIBCO MDM Studio.

Procedure

1. Right-click on the **Datasources** folder in the Project Explorer and click **Import**.



2. The Import Wizard is displayed. Expand **MDM Data Source Designer**, select **Datasource Meta Data** and click **Next.**


Import

Select

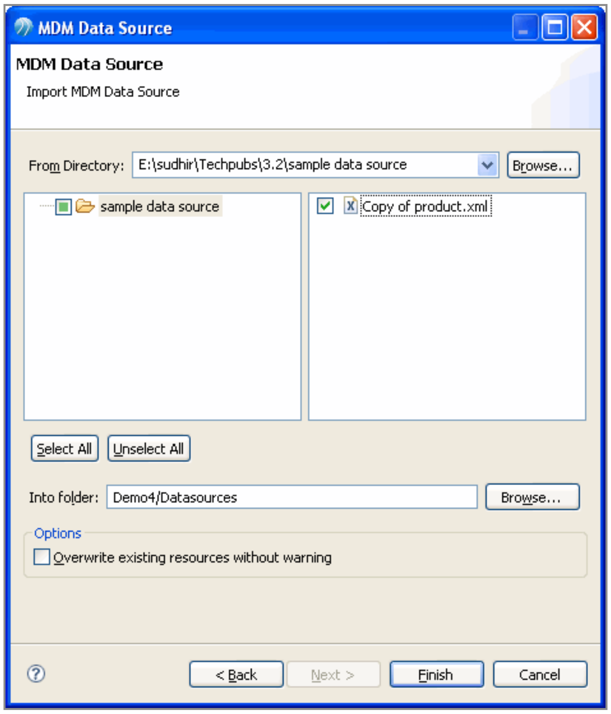
Import MDM DataSources.

Select an import source:

type filter text

3. Browse to the folder in which valid data source meta data xml file is saved. Select the data source xml file and click **Finish**.

Result



MDM Data Source

Import MDM Data Source

From Directory: E:\sudhir\Techpubs\3.2\sample data source

■ sample data source

☑ Copy of product.xml

Into folder: Demo4/Datasources

Options

□ Overwrite existing resources without warning

Note: Import Wizard supports only single data source import in data source meta data xml file.

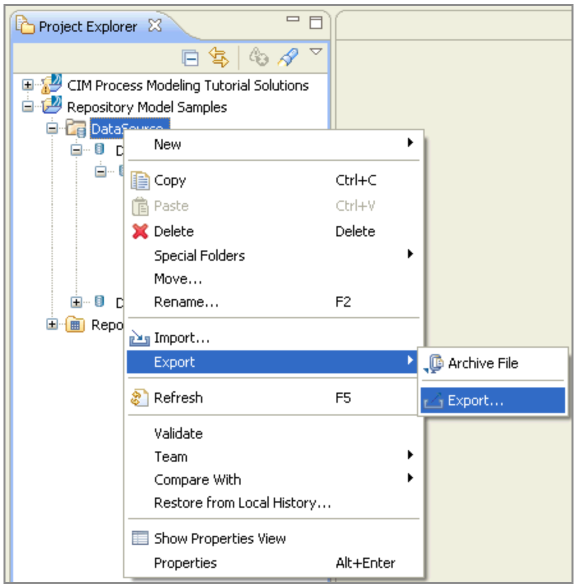
During Import you can import multiple data source model.

Exporting a Data Source

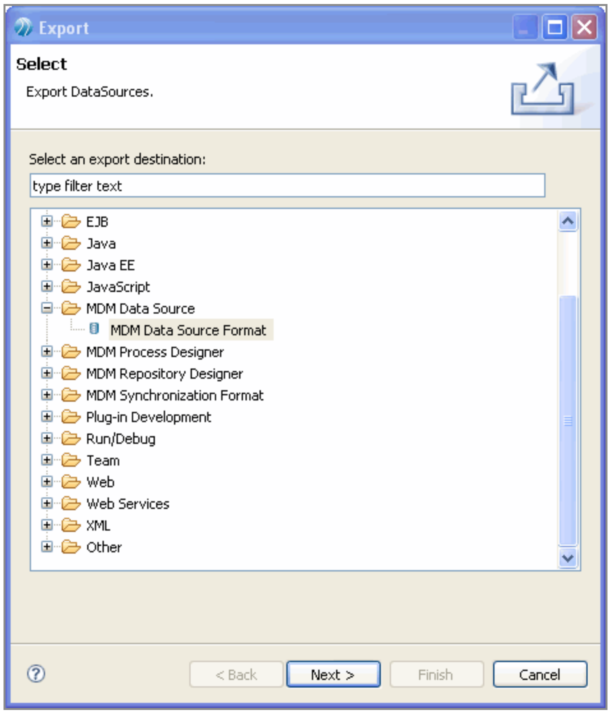
Using Export Wizard you can export Data Source model in TIBCO MDM Data Source format.

Procedure

1. Right-click on the **Datasources** folder in the Project Explorer and click **Export-- > Export**.

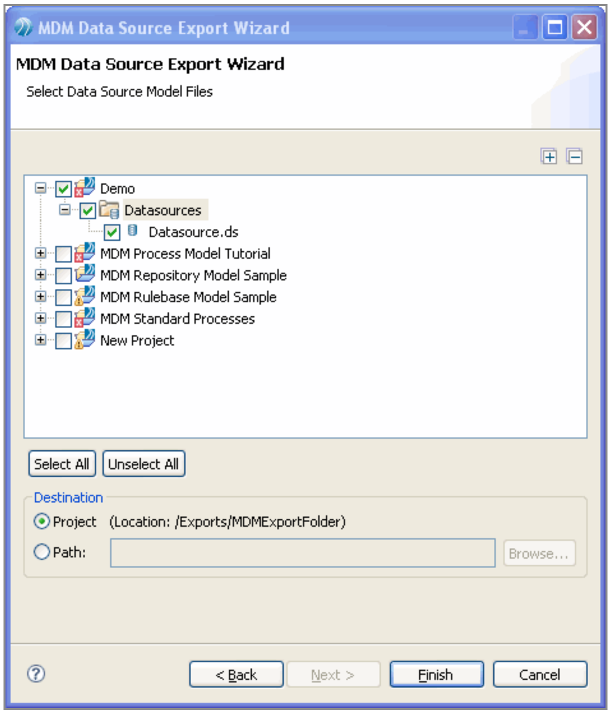


2. The Export Wizard is displayed. Expand **MDM Data Source** from export destination list, select **MDM Data Source Format** and click **Next.**



3. Select the Data Source Model which you want to export. By default Data Source Model is exported to Exports/MDMExportFolder directory which is in same project. You can change the location by specifying different destination path. Click **Path** and

browse to the folder in which you want to export the Data Source Model xml file and click **Finish.**



MDM Data Source Export Wizard

MDM Data Source Export Wizard

Select Data Source Model Files

- ☑ Demo
  - ☑ Datasources
    - ☑ Datasource.ds
- □ MDM Process Model Tutorial
- □ MDM Repository Model Sample
- □ MDM Rulebase Model Sample
- □ MDM Standard Processes
- □ New Project

Destination

● Project (Location: /Exports/MDMExportFolder)

○ Path:

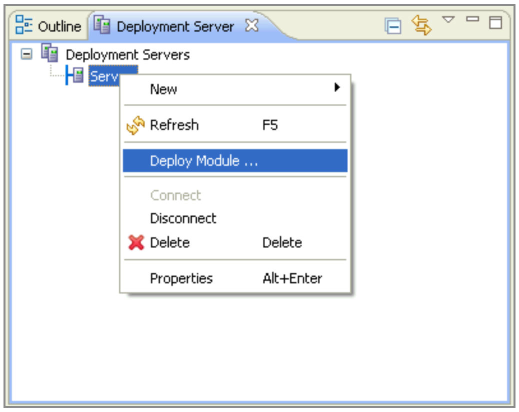
Deploying a Data Source

When a repository model (.rep file) is deployed, all the corresponding data source and its data file are also deployed.

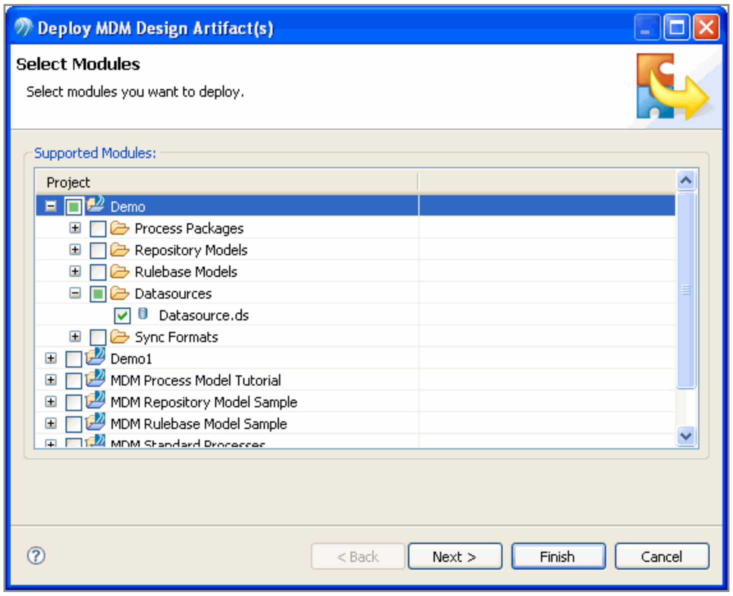
Note: While deploying data source ensure that the data source data file size is limited to 10 MB. If you want to upload data file more than 10 MB, change the preference property **Windows- > Preferences- > Repository Designer- > Data Source- > Data Source File Size Limit**. However, it is recommended to upload data files with less than 10 MB file size.

Procedure

1. In the Deployment pane, right click the <**MDM Server** and select **Deploy Module.**



2. Select the data source to deploy. Click **Next**.



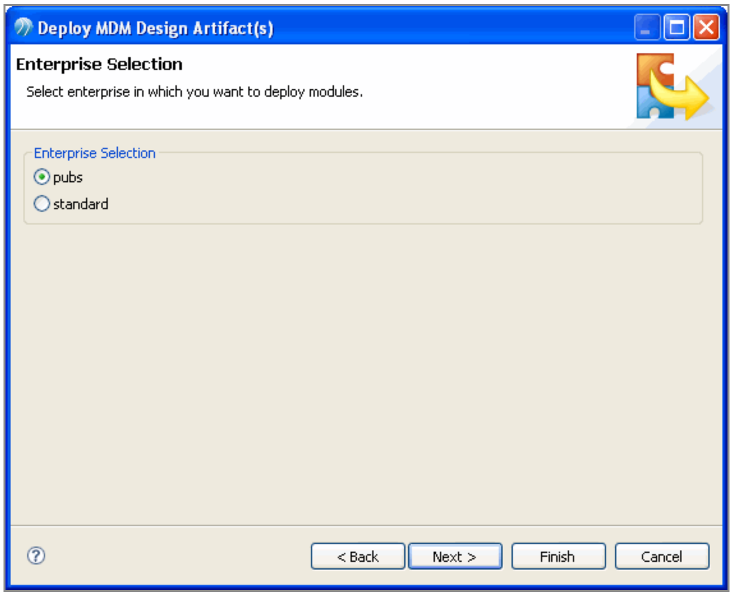
Select Modules

Select modules you want to deploy.

Supported Modules:

Project
⊟ ■ Demo
⊞ □ Process Packages
⊞ □ Repository Models
⊞ □ Rulebase Models
⊟ ■ Datasources
☑ Datasource.ds
⊞ □ Sync Formats
⊞ □ Demo1
⊞ □ MDM Process Model Tutorial
⊞ □ MDM Repository Model Sample
⊞ □ MDM Rulebase Model Sample
⊞ □ MDM Standard Processes

3. Select the enterprise to deploy the modules to (either the current enterprise or standard). Click **Next**. The selected modules is sent to the server for server side validations. The validation progress bar is shown at the bottom of the page.



Deploy MDM Design Artifact(s)

Enterprise Selection

Select enterprise in which you want to deploy modules.

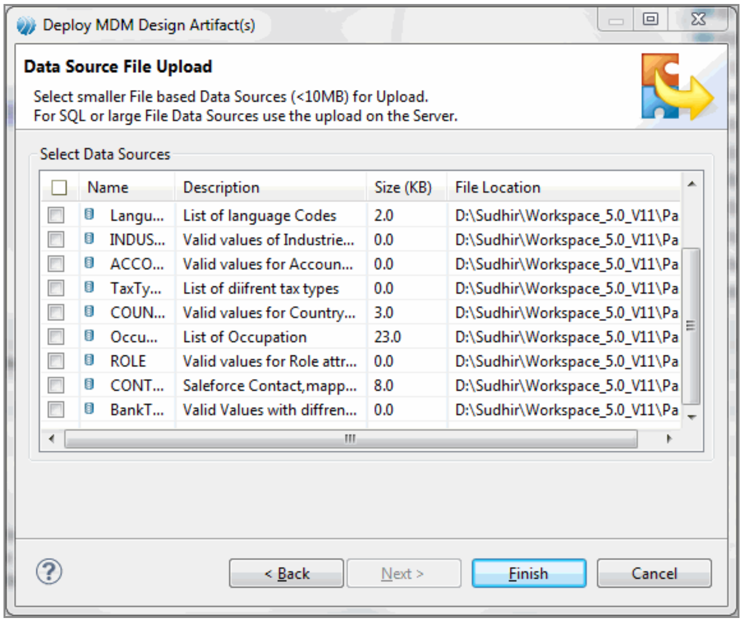
Enterprise Selection

● pubs

○ standard

4. The validation result page is displayed with the validation messages. The errors and information messages are shown in separate sections. If there are validation errors, the **Finish** button is disabled. Click **Next**.

5. Select the data source to upload, if the data source file contains the corresponding data files (*.csv, *.txt and so on) it will also be uploaded. Click **Finish**.



Deploy MDM Design Artifact(s)

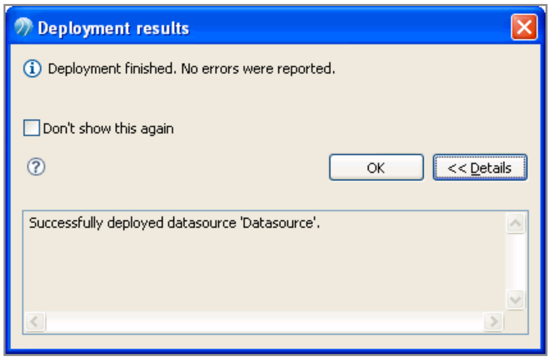
Data Source File Upload

Select smaller File based Data Sources (<10MB) for Upload.
For SQL or large File Data Sources use the upload on the Server.

Select Data Sources

□	Name	Description	Size (KB)	File Location
□	Langu...	List of language Codes	2.0	D:\Sudhir\Workspace_5.0_V11\Pa
□	INDUS...	Valid values of Industrie...	0.0	D:\Sudhir\Workspace_5.0_V11\Pa
□	ACCO...	Valid values for Accoun...	0.0	D:\Sudhir\Workspace_5.0_V11\Pa
□	TaxTy...	List of diifrent tax types	0.0	D:\Sudhir\Workspace_5.0_V11\Pa
□	COUN...	Valid values for Country...	3.0	D:\Sudhir\Workspace_5.0_V11\Pa
□	Occu...	List of Occupation	23.0	D:\Sudhir\Workspace_5.0_V11\Pa
□	ROLE	Valid values for Role attr...	0.0	D:\Sudhir\Workspace_5.0_V11\Pa
□	CONT...	Saleforce Contact,mapp...	8.0	D:\Sudhir\Workspace_5.0_V11\Pa
□	BankT...	Valid Values with diffren...	0.0	D:\Sudhir\Workspace_5.0_V11\Pa

6. The data source is successfully deployed.



Deployment finished. No errors were reported.

□ Don't show this again

Successfully deployed datasource 'Datasource'.

7. Log onto the TIBCO MDM Server and check if your data source have got included.

Result

For more information of deploying refer to, Deployment.

Synchronization Format Editor

This chapter describes how to design synchronization format in TIBCO MDM Studio.

TIBCO MDM Studio provides wizard to create Synchronization format and an editor that allows you to configure properties and define attribute groups and belonging attributes. These attributes are then mapped with the repository attributes in the output map editor.

Synchronization Formats Overview

Synchronization formats define the attributes and structure of the data required by the consumers of that data. The output map defines mappings of repository attributes to synchronization format. When a repository is synchronized, the synchronization format provides the structure for the output information.

The Synchronization format editor provides you with a User Interface that helps create Synchronization format which can then be referred in the Output Map during mappings.

While creating a new MDM Developer project, the synchronization special folder named Sync Format is created. If the Sync format folder is not created you can manually create the special folder. For more information, refer to Creating the Synchronization Format Special Folder.

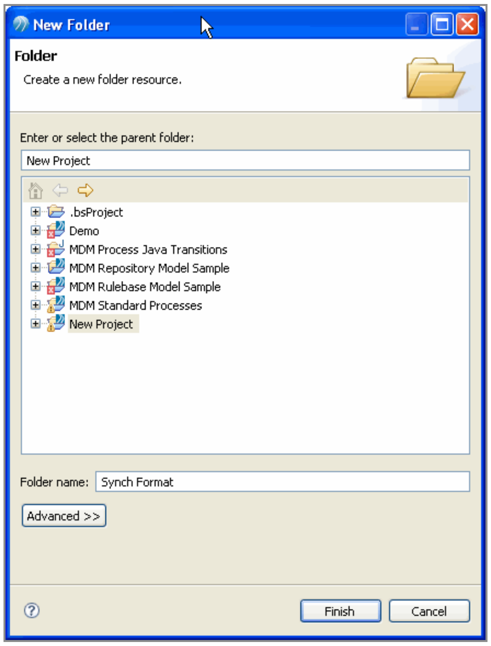
Creating the Synchronization Format Special Folder

While creating a new MDM Developer project, the synchronization special folder named Sync Format is created. If the Sync format folder is not created you can manually create the special folder.

Procedure

1. Right click the **MDM Developer Project**, point to **New** and click **Folder**. Enter a user

friendly name in the **Folder name** field. For example, Sync Format. Click **Finish**.



New Folder

Folder

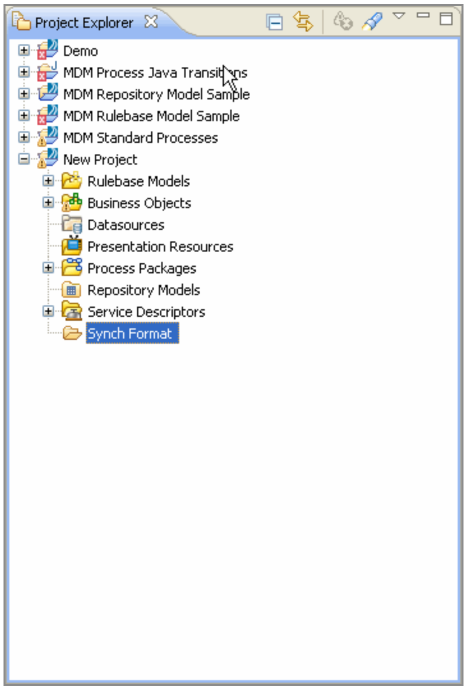
Create a new folder resource.

Enter or select the parent folder: New Project

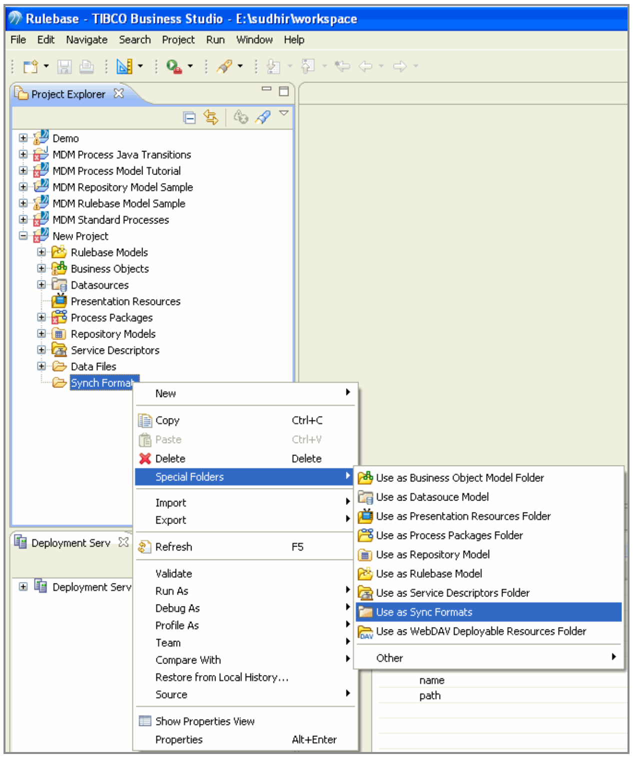
Folder name: Synch Format

.bsProject
Demo
MDM Process Java Transitions
MDM Repository Model Sample
MDM Rulebase Model Sample
MDM Standard Processes
New Project
Advanced >>
Finish
Cancel

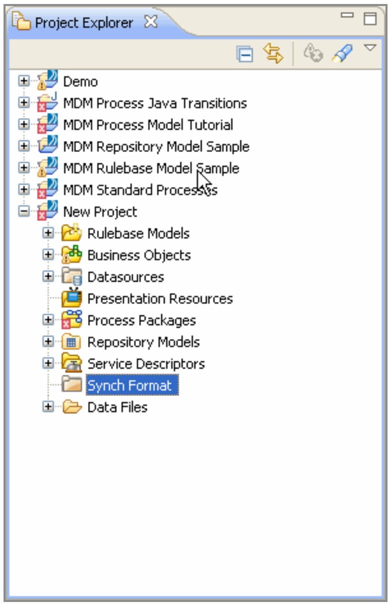
2. A folder by the name Sync Format is created in the MDM Developer Project.



3. To designate it as a special folder, right click newly created folder (Synch Format) and select **Special Folder** and click **Use as Synch Formats**.



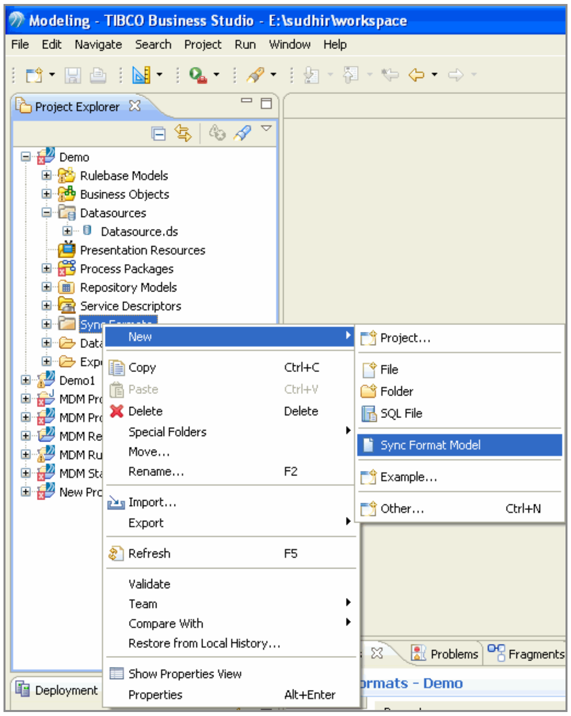
4. The newly created folder gets designated as Special Folder by the name Sync Folder.



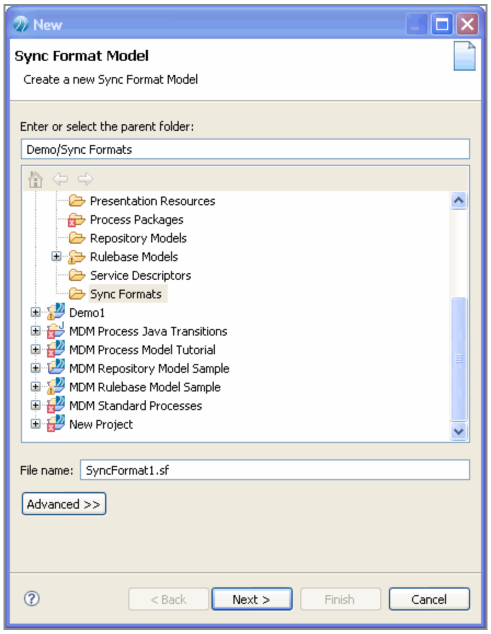
Creating a Synchronization Format

Procedure

1. Right click on the Sync Format Special folder, point to **New** and click **Sync Format Model**.



2. Enter the appropriate file name for the synchronization format in the **File name** field. The extension of the file name must be **.sf**.



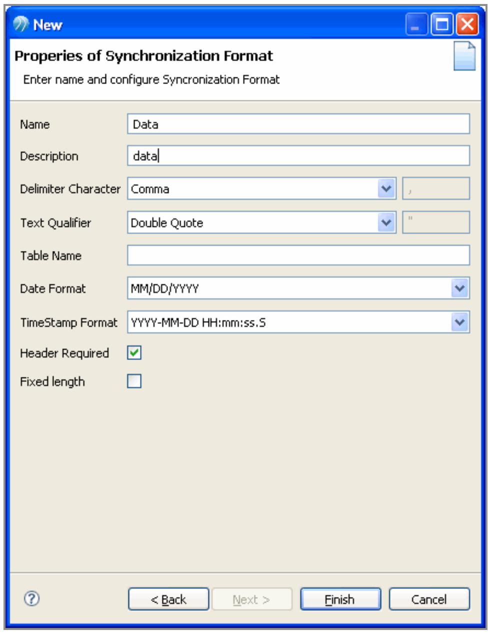
Sync Format Model

Create a new Sync Format Model

Enter or select the parent folder: Demo/Sync Formats

File name: SyncFormat1.sf

3. Click **Next**. The properties of the synchronization format are displayed.



Properies of Synchronization Format

Enter name and configure Syncronization Format

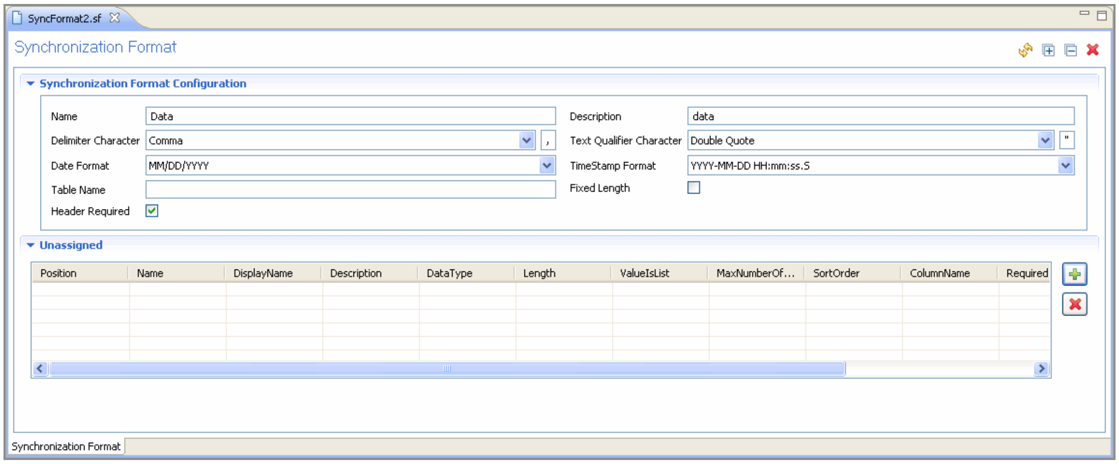
Name	Data
Description	data
Delimiter Character	Comma ,
Text Qualifier	Double Quote "
Table Name	
Date Format	MM/DD/YYYY
TimeStamp Format	YYYY-MM-DD HH:mm:ss.S
Header Required	☑
Fixed length	☐

4. Enter or select the appropriate values in the respective fields.

Field	Description
Name	The synchronization name must be unique and can contain A-Z, 0-9, all special characters (except a space), and characters from other languages.
Description	The synchronization format description. The description field does not allow more than 80 characters. However, white spaces are allowed in the description.
Delimiter Character	The delimiter character can be a Comma, Pipe, Semicolon, Space, Tab, or Other character. The backslash (\) is not supported as delimiter.
Text Qualifier Character	The text qualifier can be a double quote, Single quote, or any other qualifier. Data field may contain Delimiter character. In such case data field should be enclosed with Text Qualifier.
Date Format	The date format must match the actual date format. If specified incorrectly, data may be rejected or loaded incorrectly. For example, if the date format is specified as MM/DD/YY, and the actual data is specified in DD/MM/YY, then 02/01/12 may be imported as Feb-01-2012 instead of Jan-02-20120.
TimeStamp Format	The timestamp format must match the timestamp format used in the actual data. If you specify incorrect timestamp format, data may be rejected or loaded incorrectly. By default, YYYY-MM-DD HH:mm:ss.S timestamp format is displayed. If the actual data contains a different format, you can change it.

Field	Description
Table Name	The table name where the synchronized data is stored.
Header Required	Select the checkbox if you want the header to appear in the synchronized output file.
Fixed Length	Select the checkbox if you want the synchronized file to be of fixed length.

5. Click **Finish.** The synchronization format editor for the newly created synchronization format is displayed.



Name	Data
Delimiter Character	Comma ,
Date Format	MM/DD/YYYY
Table Name	
Header Required	☑
Description	data
Text Qualifier Character	Double Quote "
TimeStamp Format	YYYY-MM-DD HH:mm:ss.S
Fixed Length	□

Position	Name	DisplayName	Description	DataType	Length	ValueIsList	MaxNumberOf...	SortOrder	ColumnName	Required

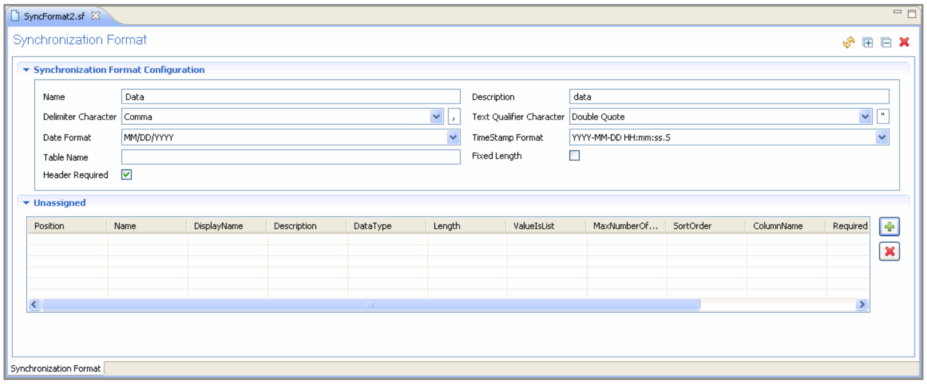
Modify Existing Synchronization Format

Using the Synchronization Format Editor you can modify the existing Synchronization Format in the project. The Synchronization Format Editor has two sections.

> **Note:** The default synchronization formats cannot be modified. Only the synchronization formats that you create can be modified. Also, you cannot modify the synchronization format that is in use for Output Map or Export.

- **Synchronization Format Configuration**: Here you can modify the basic details, such as name of the data synchronization format, description, delimiter character, text qualifier character, data format, timestamp format, table name, and other details.

- **Attributes**: All the attributes of synchronization format can be viewed and modified in this section. Click ✚ to add new attributes.



Name	Data
Description	data
Delimiter Character	Comma ,
Text Qualifier Character	Double Quote "
Date Format	MM/DD/YYYY
TimeStamp Format	YYYY-MM-DD HH:mm:ss.S
Table Name	
Fixed Length	□
Header Required	☑

Position	Name	DisplayName	Description	DataType	Length	ValueIsList	MaxNumberOf...	SortOrder	ColumnName	Required

Field	Description
Position	The position of the of synchronization format attribute.
Name	Name of the of synchronization format attribute.
DisplayName	The display name of the synchronization format attribute.
Description	The description of the synchronization format attribute.
DataType	The data type of the synchronization format attribute. The supported data types are String, File, Amount, Boolean, Integer, Date, Decimal, Custom Decimal, Timestamp, Long, and URL.
Length	The length of the synchronization format attribute.
ValueIsList	Specify whether the synchronization format

Field	Description
	attribute is a list value. The possible value is true or false. It is a toggle field.
MaxNumberOfValues	Specify the maximum number of values to be displayed after the synchronization format. By default it takes ALL (-1).
SortOrder	Specify the sort order of the synchronization format.
ColumnName	Specify the column name of the synchronization format attribute.
Required	Specify whether the synchronization format attribute is a required field. The possible value are true or false. It is a toggle field.
Searchable	Specify whether the synchronization format attribute should be a searchable value. The possible value are true or false. It is a toggle field.
DisplayInRecordList	Specify whether the synchronization format attribute should be displayed during the record search. The possible value are true or false. It is a toggle field.

- To delete an attribute select the checkbox corresponding to the attribute and click ✖.

Manage Attribute Groups

Using the Synchronization Format Editor you can create and manage the attribute groups for a synchronization format.

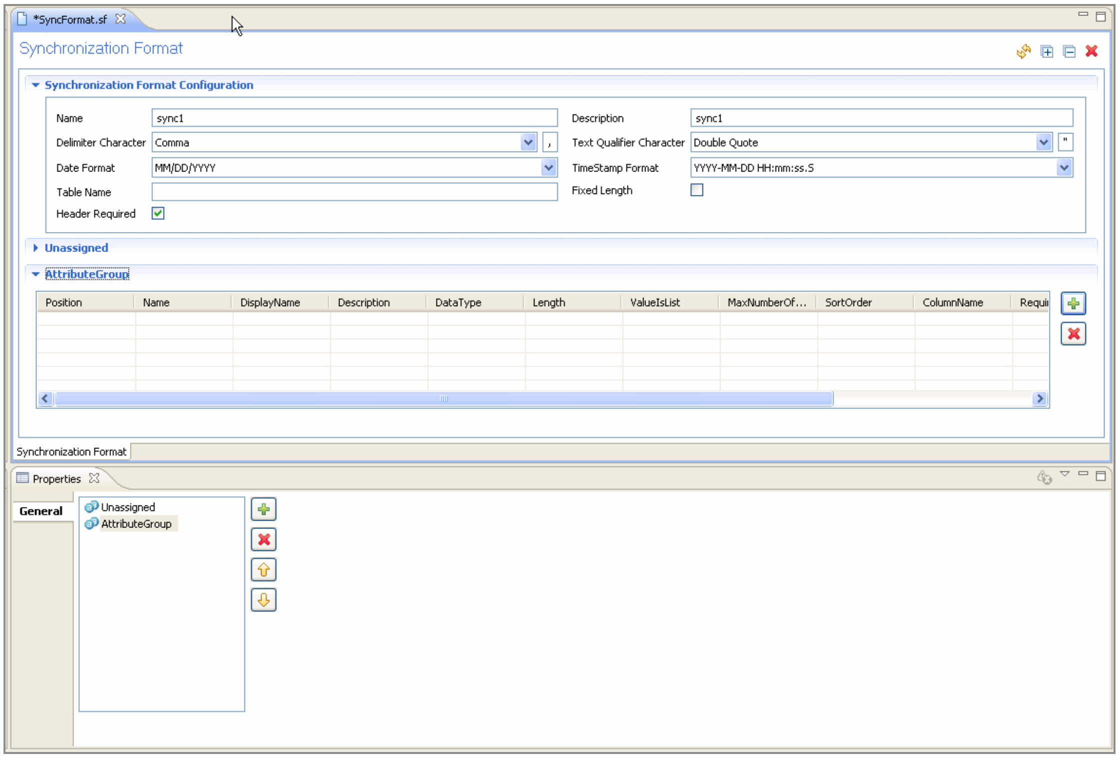
Creating a New Attribute Group

The attribute property section allows you to create a new attribute group.

Procedure

1. On the property section, click ➕.

   The new attribute group is created in the synchronization format editor. To create attributes in the attribute group, see Modify Existing Synchronization Format.



Synchronization Format Configuration

Name	sync1	Description	sync1
Delimiter Character	Comma ,	Text Qualifier Character	Double Quote "
Date Format	MM/DD/YYYY	TimeStamp Format	YYYY-MM-DD HH:mm:ss.S
Table Name		Fixed Length	□
Header Required	☑		

Unassigned

AttributeGroup

Position	Name	DisplayName	Description	DataType	Length	ValueIsList	MaxNumberOf...	SortOrder	ColumnName	Requi

Unassigned
AttributeGroup

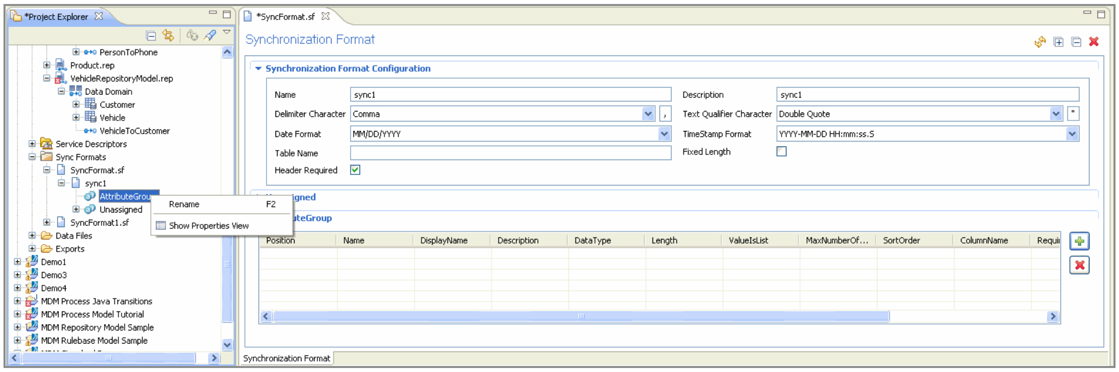
2. To change the order of the attribute groups use ⇧ or ⇩.
3. To delete a attribute group select the attribute group and click ✖.

Renaming the Attribute Group

You can change the default attribute group name.

Procedure

1. On the project explorer, select the <Attribute Group> which you want to rename.
2. Right-click and select **Rename**.



Name	sync1	Description	sync1
Delimiter Character	Comma	Text Qualifier Character	Double Quote
Date Format	MM/DD/YYYY	TimeStamp Format	YYYY-MM-DD HH:mm:ss.S
Table Name		Fixed Length	□
Header Required	☑		

Position	Name	DisplayName	Description	DataType	Length	ValueIsList	MaxNumberOf...	SortOrder	ColumnName	Requi

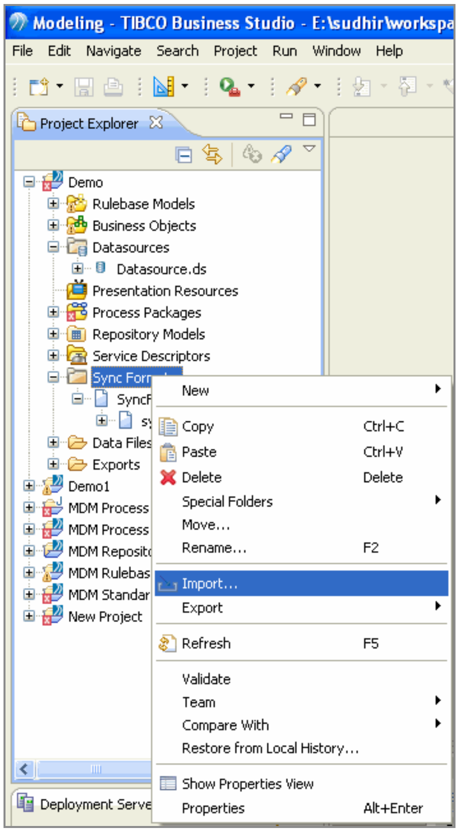
3. Enter the new attribute group name.

Importing a Synchronization Format

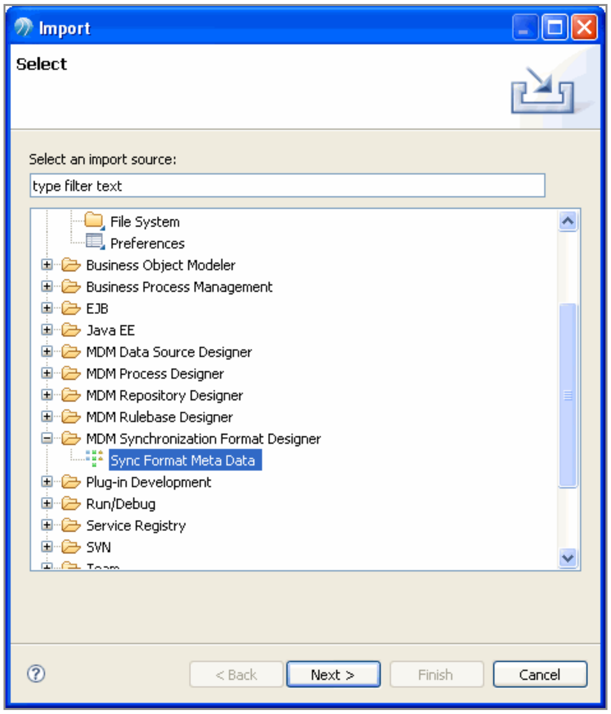
Using the Import Wizard you can import Synchronization Format into TIBCO MDM Studio.

Procedure

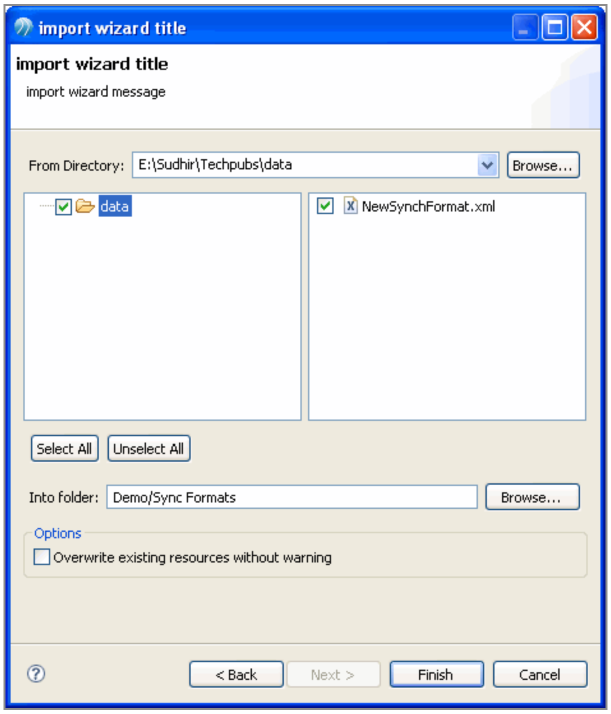
1. Right-click on the **Sync Formats** folder in the Project Explorer and click **Import**.



2. The Import Wizard is displayed. Expand **MDM Synchronization Format Designer**, select **Sync Format Meta Data** and click **Next.**



3. Browse to the folder in which valid sync format meta data xml file saved. Select the sync format xml file and click **Finish.**



From Directory: E:\Sudhir\Techpubs\data

☑ data

☑ NewSynchFormat.xml

Into folder: Demo/Sync Formats

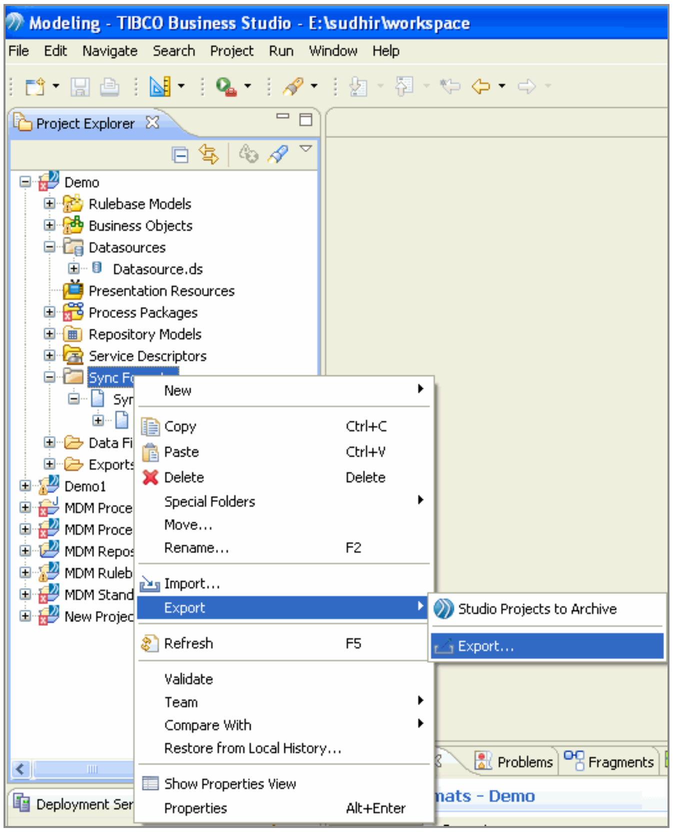
□ Overwrite existing resources without warning

Exporting a Synchronization Format

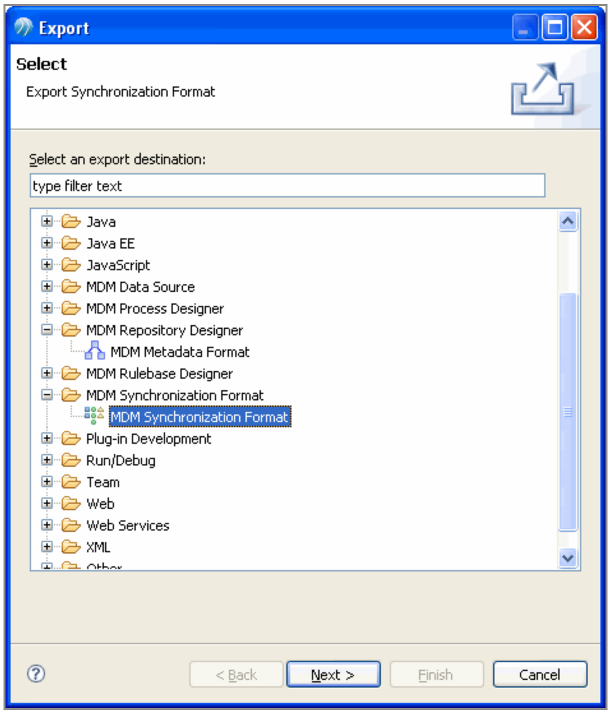
Using Export Wizard you can export a Synchronization Format in to TIBCO MDM metadata xml format.

Procedure

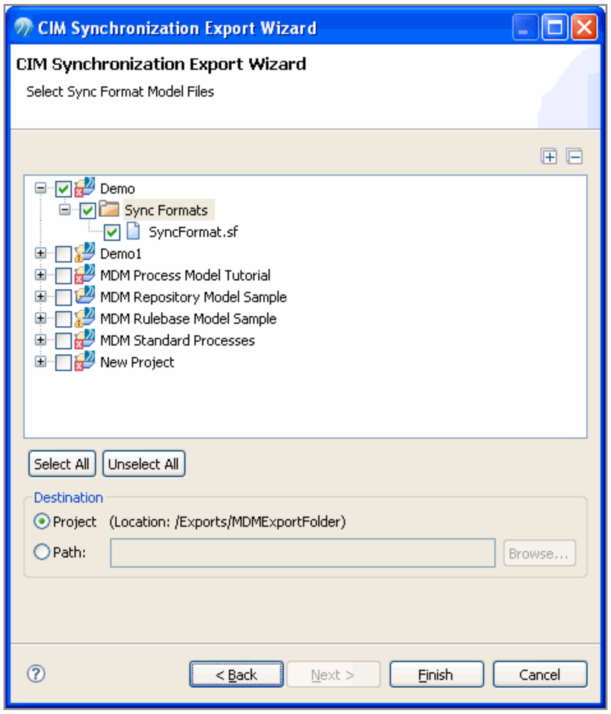
1. Right-click on the **Sync Formats** folder in the Project Explorer and click **Export-- > Export**.



2. The Export Wizard is displayed. Expand **MDM Synchronization Format** from export destination list, select **MDM Synchronization Format** and click **Next.**



3. Select the synchronization format which you want to export. By default Synchronization Format is exported to Exports/MDMExportFolder directory which is in same project. You can change the location by specifying different destination path. Click **Path** and browse to the folder in which you want to export the synchronization format xml file and click **Finish.**



CIM Synchronization Export Wizard

Select Sync Format Model Files

- ☑ Demo
  - ☑ Sync Formats
    - ☑ SyncFormat.sf
- □ Demo1
- □ MDM Process Model Tutorial
- □ MDM Repository Model Sample
- □ MDM Rulebase Model Sample
- □ MDM Standard Processes
- □ New Project

Destination

● Project (Location: /Exports/MDMExportFolder)

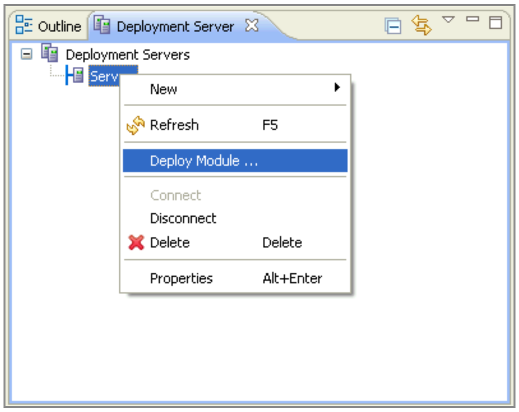
○ Path:

Deploying a Synchronization Format

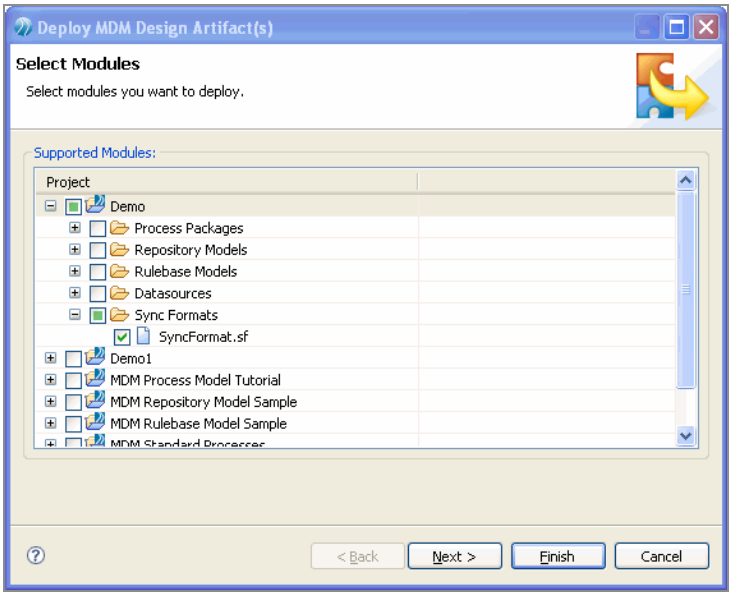
In deployment wizard, sync formats are listed under Sync Formats folder and can be deployed independent of repository model. When a repository model (.rep file) is deployed, all the corresponding synchronization formats are also deployed.

Procedure

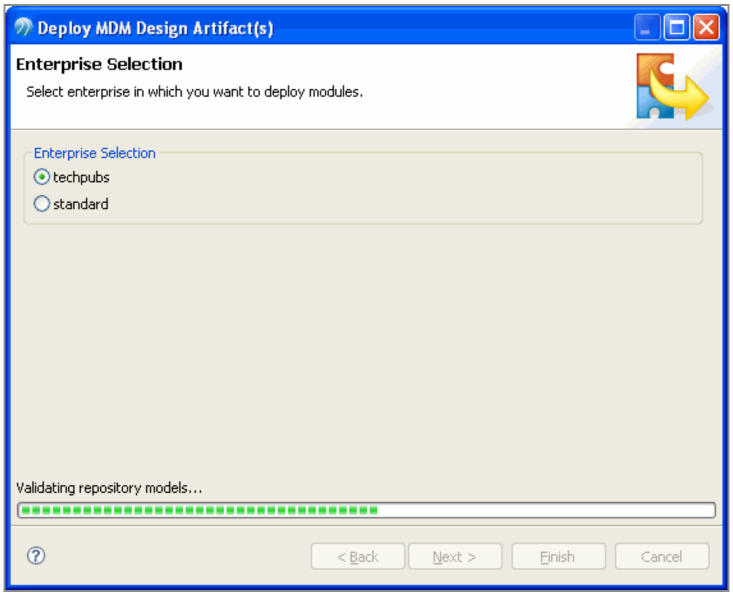
1. In the Deployment pane, right click the <**MDM Server** and select **Deploy Module.**



2. Select the synchronization format to deploy. Click **Next**.



3. Select the synchronization format to deploy and click **Next**.



Select enterprise in which you want to deploy modules.

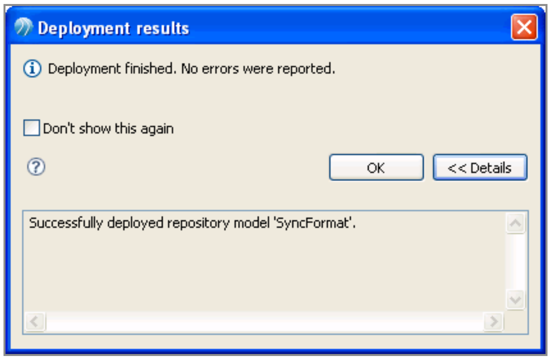
Enterprise Selection

● techpubs

○ standard

Validating repository models...

4. Click **Finish**.
5. The synchronization format is successfully deployed.



Deployment finished. No errors were reported.

□ Don't show this again

Successfully deployed repository model 'SyncFormat'.

6. Log onto the TIBCO MDM Server and check if your synchronization format have got included.

Result

For more information of deploying refer to, Deployment.

Classification Scheme Editor

This chapter introduces you to the concept of Category Specific Attributes in TIBCO MDM Studio and explains how to design and deploy the Classification Scheme using Classification Scheme Editor.

Overview

The Classification Scheme allows you to arrange the various categories of the products in a particular hierarchy. The Classification Scheme Editor allows to define the Category Code hierarchy. Some of the features supported by Classification Editor are as follows:

Editing Category Tree

- Allows adding child and sibling Classification Code.
- Allows Inline editing of the Classification Code name on the tree node and Classification Code Property section allows editing of the code, name, and description.
- Allows linking and unlinking of Category Specific Attributes to Classification Code.

Browsing Category Tree

- Allows browsing and searching of Classification Code within the tree based on the name.
- Allows viewing of linked attributes and the inherited attributes.

The Classification Scheme which are defined using the Classification Scheme Editor are deployed using the TIBCO MDM Studio. Since the definition of Classification Scheme is managed in TIBCO MDM Studio, it is necessary to deploy all the artifacts to the TIBCO MDM Server.

For example, in large departmental stores we find all the products we need in our day to day life. It is well organized and arranged such that customers can easily find the items in their shopping list. From the retailer's point of view, managing such huge products and stock taking is a challenge. Hence the products are grouped and sub-grouped, categorized and sub-categorized depending on the nature of product. Thus allowing the retailer to capture the various details of a product based on its group and category.

Create Classification Schemes

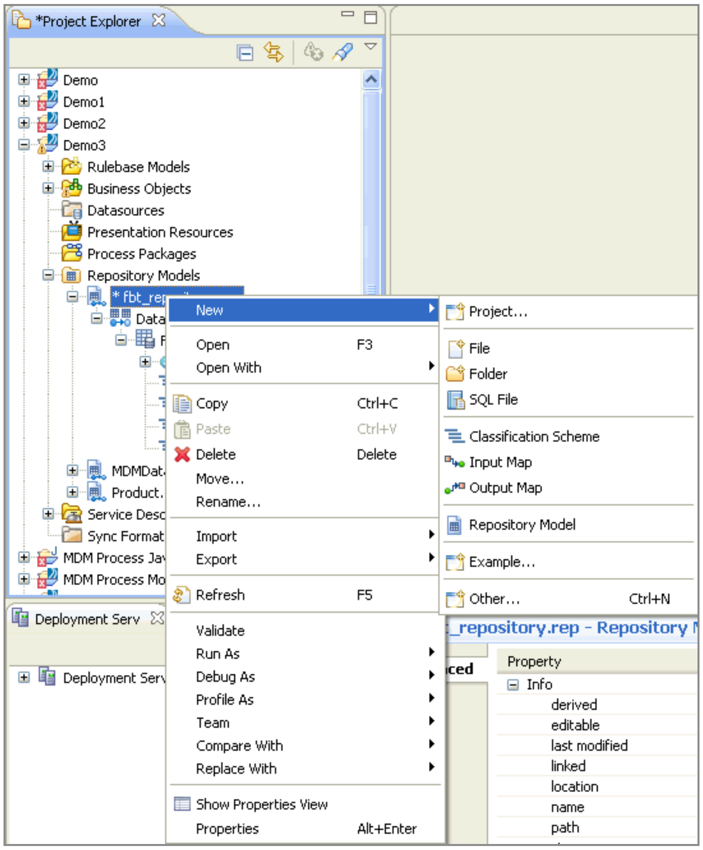
The Classification Schemes are created using the following two ways:

- Classification Scheme wizard.
- Classification Scheme property section.

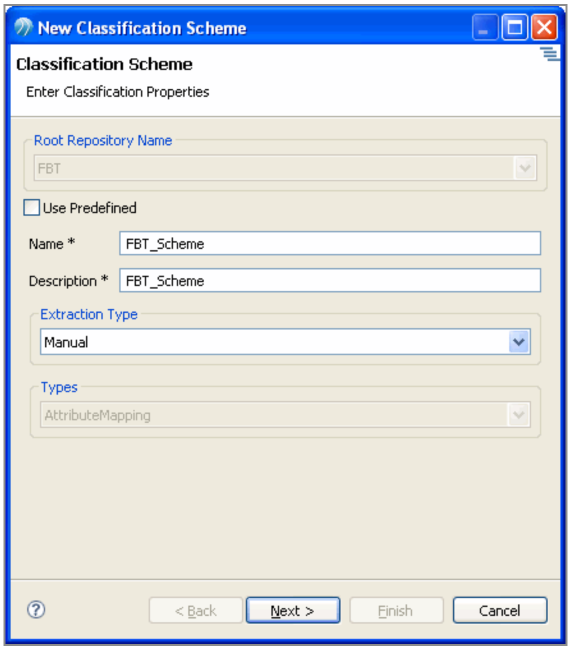
Using Classification Scheme Wizard

Procedure

1. Select the Repository Model for which you want to create the classification and right-click on **New > Classification Scheme**.



2. The Classification Scheme wizard is displayed.



Classification Scheme

Enter Classification Properties

Root Repository Name: FBT

☐ Use Predefined

Name *: FBT_Scheme

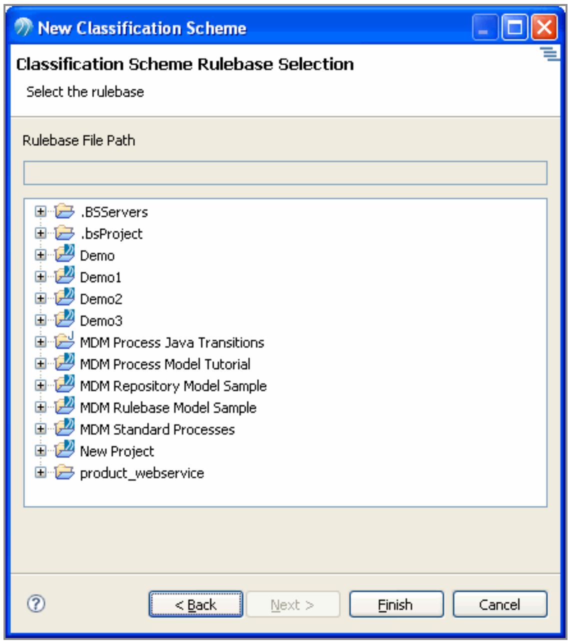
Description *: FBT_Scheme

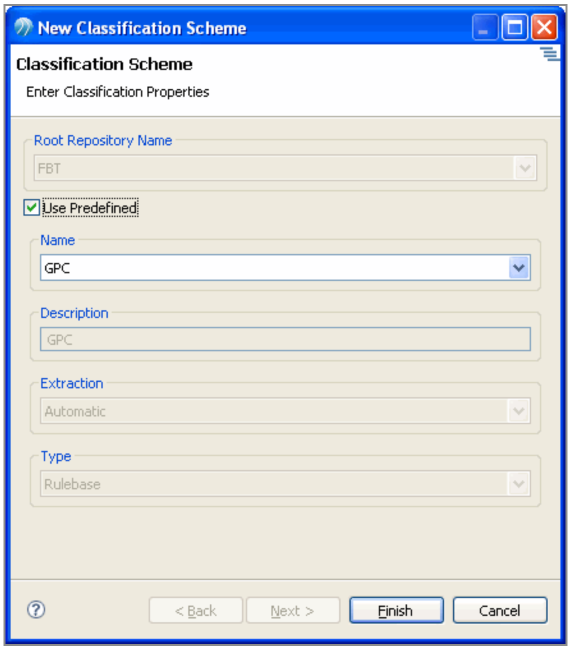
Extraction Type: Manual

Types: AttributeMapping

Field	Description
Root Repository Name	The repository name for which the classification scheme is created. The Root repository name is view only field if we select the particular repository. However, if we select repository model file having more than one repository then we can change the repository.
Name	Name of the Classification Scheme. The Classification Scheme Name cannot exceed 78 characters. The Classification Scheme name is mandatory.
Description	The description for the Classification Scheme. The classification description is mandatory.
Extraction Type	Select the appropriate extraction type from the drop-down. The available options are **Manual**, **Automatic**, and **Mixed**. By default, **Manual** is displayed. **Manual**: Use this option to manually define Classification Code hierarchy. When record is added, it will not be automatically classified. You have to click on the **open** link to open to classify record in the Classification Code editor and define the Classification Code. **Automatic:** Use this option to create Classification Code hierarchy automatically based on repository attribute hierarchy. When the Classification Scheme type **AttributeMapping** is selected and extraction type is **Automatic**, the **Classification Scheme Attribute** section is displayed and you must map the repository defined attributes for each hierarchical position. As attribute definition

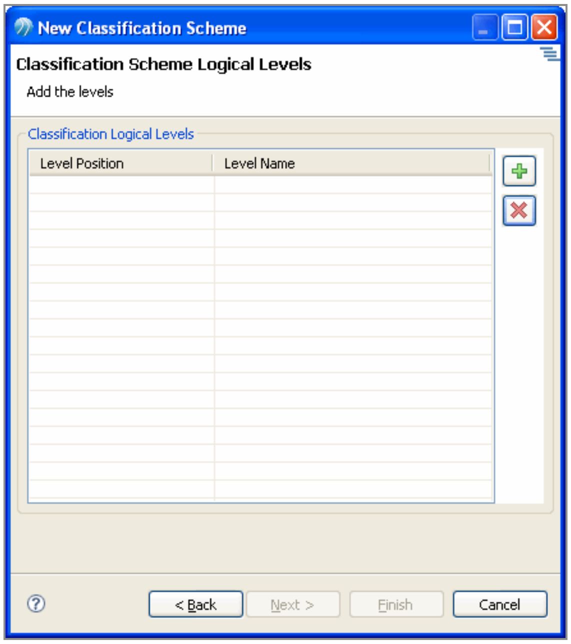
Field	Description
	defines how and where to extract data for forming category hierarchy as well as classifying record. **Mixed:** Use this option if you want to first classify the records using **Automatic** extraction type and later on use the **Manual** extraction type to classify the record manually. On selecting the **Mixed** extraction type, the **Classification Scheme Logical Level** section is followed by the **Classification Scheme** Attribute section.
Types	Select the classification type from the drop-down. The available options are **AttributeMapping** and **Rulebase**. This field is enabled only for Automatic and Mixed Classification Scheme. **AttributeMapping:** The AttributeMapping type classification scheme support all the three extraction types Automatic, Manual and Mixed. **Rulebase:** The Rulebase classification type supports Automatic and Mixed extraction types.

Field	Description
	 New Classification Scheme Classification Scheme Rulebase Selection Select the rulebase Rulebase File Path .BSServers .bsProject Demo Demo1 Demo2 Demo3 MDM Process Java Transitions MDM Process Model Tutorial MDM Repository Model Sample MDM Rulebase Model Sample MDM Standard Processes New Project product_webservice
Use Predefined	Select the **Use Predefined** check box to select the predefined classification scheme. The **Name** field is enabled and all the other field are read-only. The available options are **GPC** and **UDEX**. If you select Use Predefined checkbox, classification attributes are added to the repository as per GPC or UDEX. Select the predefined classification schemes which you want to associate with repository from the available options and click **Finish**.

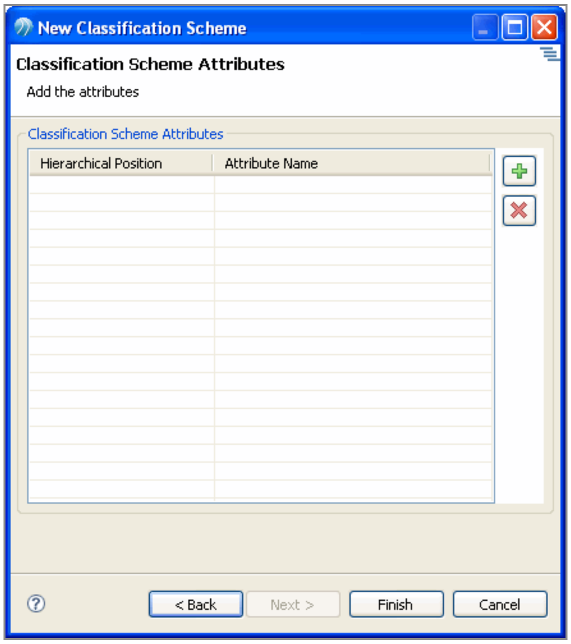
Field	Description
	 New Classification Scheme Classification Scheme Enter Classification Properties Root Repository Name: FBT ☑ Use Predefined Name: GPC Description: GPC Extraction: Automatic Type: Rulebase The **Classification Scheme Editor** is displayed with predefined classification scheme.

3. Click **Next**.

    - On selecting the Manual, Mixed or Automatic extraction types, Classification Scheme Logical Levels dialog box is displayed.


New Classification Scheme
Classification Scheme Logical Levels
Add the levels
Classification Logical Levels
Level Position
Level Name
< Back
Next >
Finish
Cancel

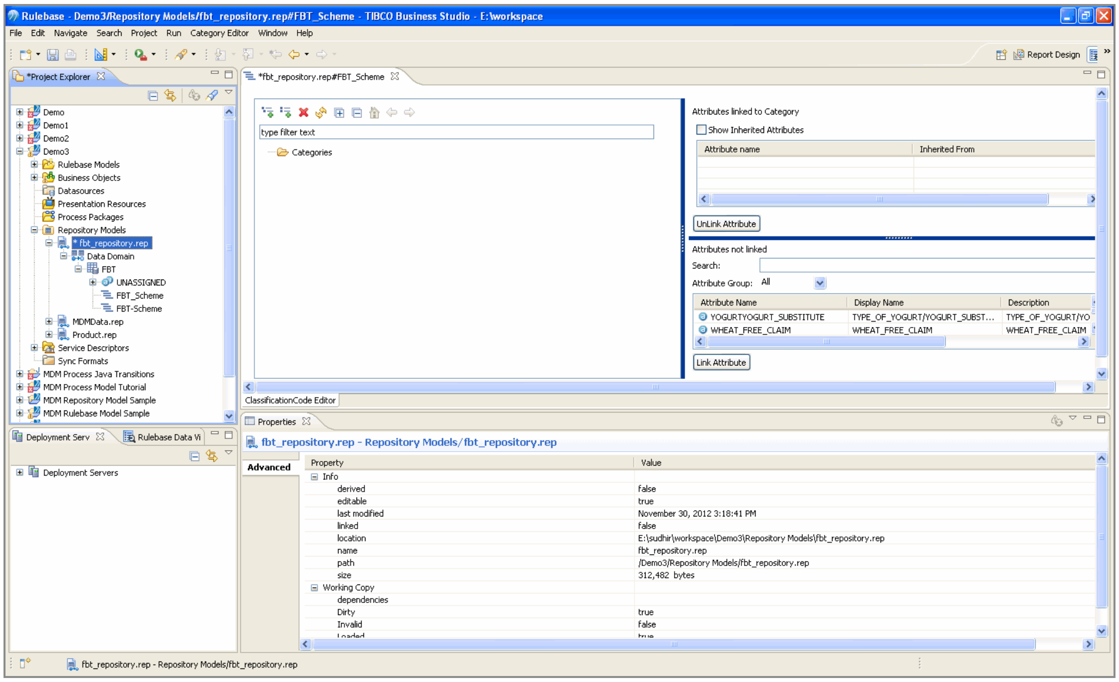
- Define the logical level for the Classification Scheme. Click ✚ to add a logical level. It is mandatory to create minimum one logical level.
- On selecting the Automatic and Mixed extraction types and type as Attribute Mapping, the Classification Scheme Attributes dialog box is displayed and if Rulebase type is selected then **Rulebase selection** page displayed.


New Classification Scheme
Classification Scheme Attributes
Add the attributes
Classification Scheme Attributes
Hierarchical Position
Attribute Name
< Back
Next >
Finish
Cancel

- Click ✚ to add the repository defined attributes for each hierarchical position and click **Finish**.

4. On selecting the **Automatic** and **Mixed** extraction types and type as **Attribute Mapping**, the Classification Scheme Attributes dialog box is displayed and if **Rulebase** type is selected then **Rulebase selection** page displayed and click **Finish**.

5. The **Classification Scheme Editor** is displayed. The classification scheme editor is displayed for manual, mixed and automatic Classification extraction types.



Attributes linked to Category

□ Show Inherited Attributes

Attribute name	Inherited From

Attributes not linked

Search:

Attribute Group: All

Attribute Name	Display Name	Description
YOGURTYOGURT_SUBSTITUTE	TYPE_OF_YOGURT/YOGURT_SUBST...	TYPE_OF_YOGURT/YO
WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM

fbt_repository.rep - Repository Models/fbt_repository.rep

Property	Value
Info	
derived	false
editable	true
last modified	November 30, 2012 3:18:41 PM
linked	false
location	E:\sudhir\workspace\Demo3\Repository Models\fbt_repository.rep
name	fbt_repository.rep
path	/Demo3/Repository Models/fbt_repository.rep
size	312,482 bytes
Working Copy	
dependencies	
Dirty	true
Invalid	false
Loaded	true

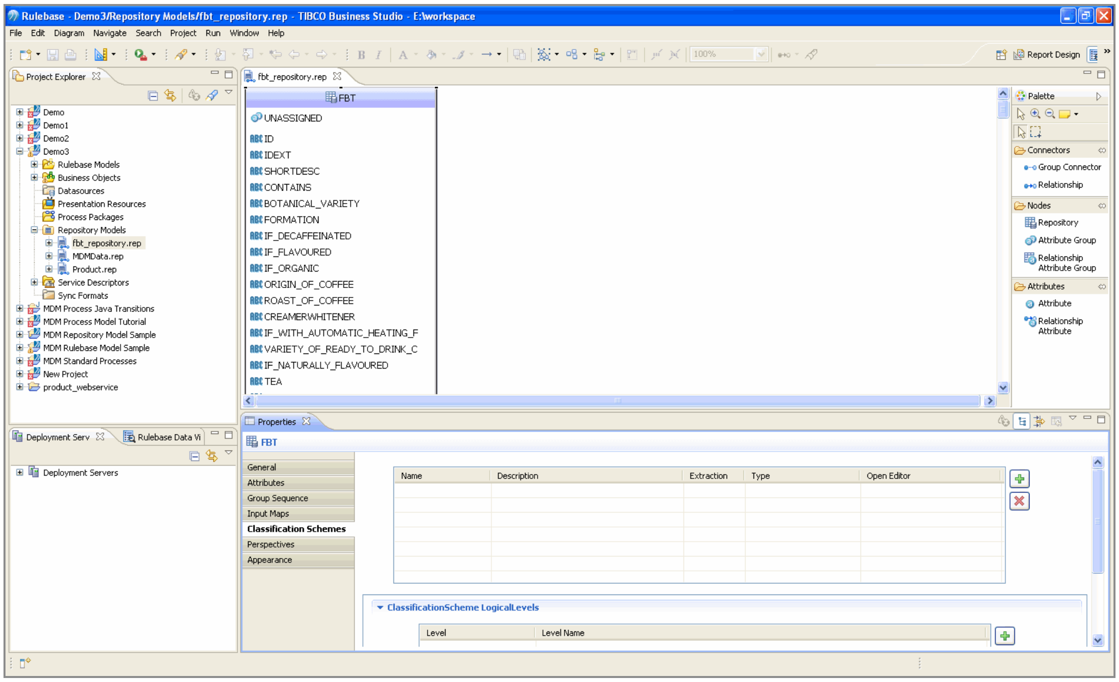
fbt_repository.rep - Repository Models/fbt_repository.rep

6. Using the Classification Scheme Editor you can create the classification code hierarchy. For more information on creating Classification Code refer to Creating Classification Code Hierarchy.

Using Classification Scheme Property Section

Procedure

1. Select the Repository for which you want to create the Classification Scheme and click the **Classification Schemes** tab.



2. The editable table with the classification parameters is displayed. Click + to add new classification scheme. Enter the appropriate Classification Scheme name in the **Name** column.

3. Enter the appropriate description for the Classification Scheme in the **Description** column.

4. Select the appropriate extraction type from the drop-down. The available options are **Manual**, **Automatic**, and **Mixed**.

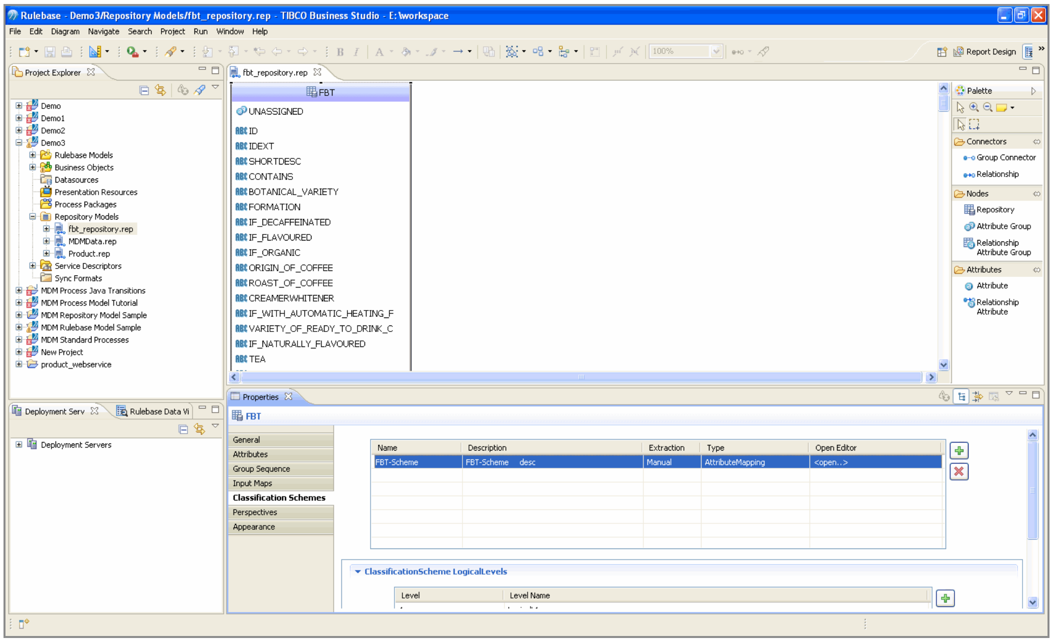
   - **Manual**: The manual extraction type supports Classification Scheme Logical Levels.

   - **Automatic/Mixed**: The automatic and mixed extraction types supports Classification Scheme logical levels and Classification Scheme attribute sections.

5. Select the classification type from the drop-down. The available options are **AttributeMapping** and **Rulebase**.

   - **AttributeMapping**: The attributemapping type classification scheme support all the three extraction types Automatic, Manual and Mixed.

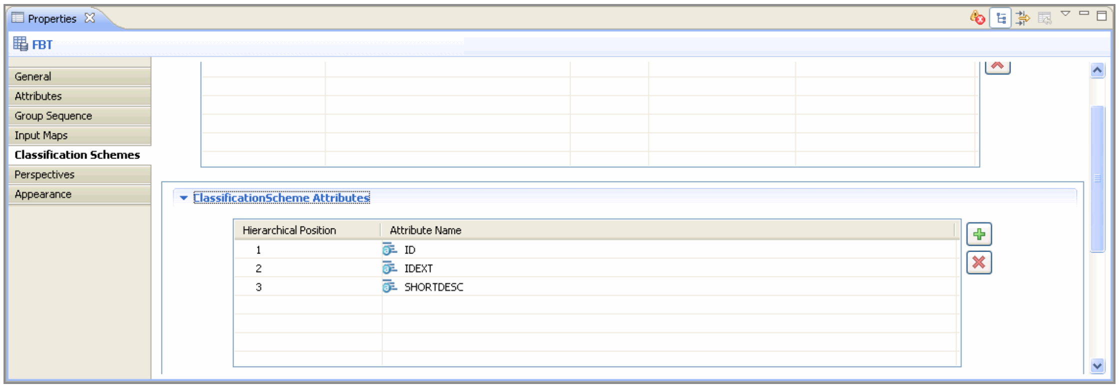
   - **Rulebase**: This Rulebase classification type supports automatic and mixed extraction type. On selecting the Rulebase classification type, click the Browse button and select the already defined rulebase. The selected rulebase is displayed.



Name	Description	Extraction	Type	Open Editor
FBT-Scheme	FBT-Scheme desc	Manual	AttributeMapping	<open..>

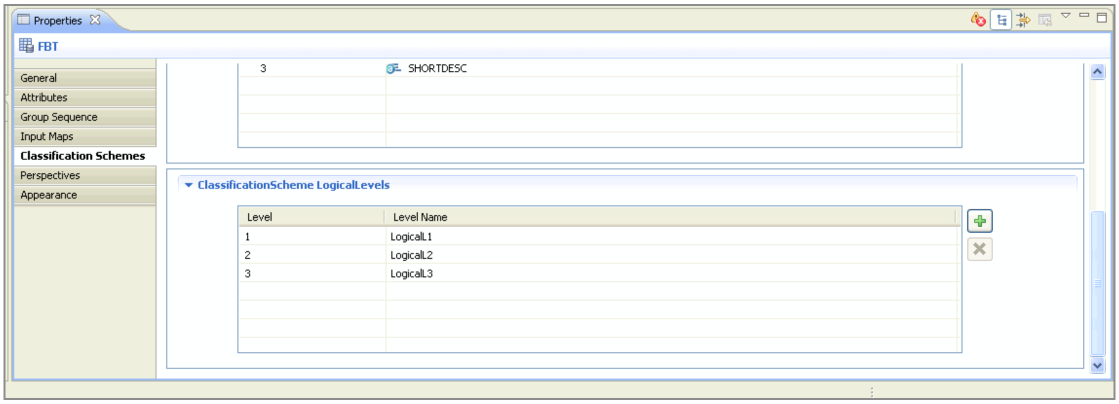
Level	Level Name

6. Create the Classification Scheme Attributes. Click ✚ to create Category Specific Attributes. By default, the Hierarchical position of the Classification Attribute are displayed. Select the Category Specific Attribute from the drop-down list.



Hierarchical Position	Attribute Name
1	ID
2	IDEXT
3	SHORTDESC

7. Create the Classification Scheme Logical Levels. Click ✚ to create logical levels. By default, the logical levels and Logical Level names are displayed. You can modify the Logical level names.

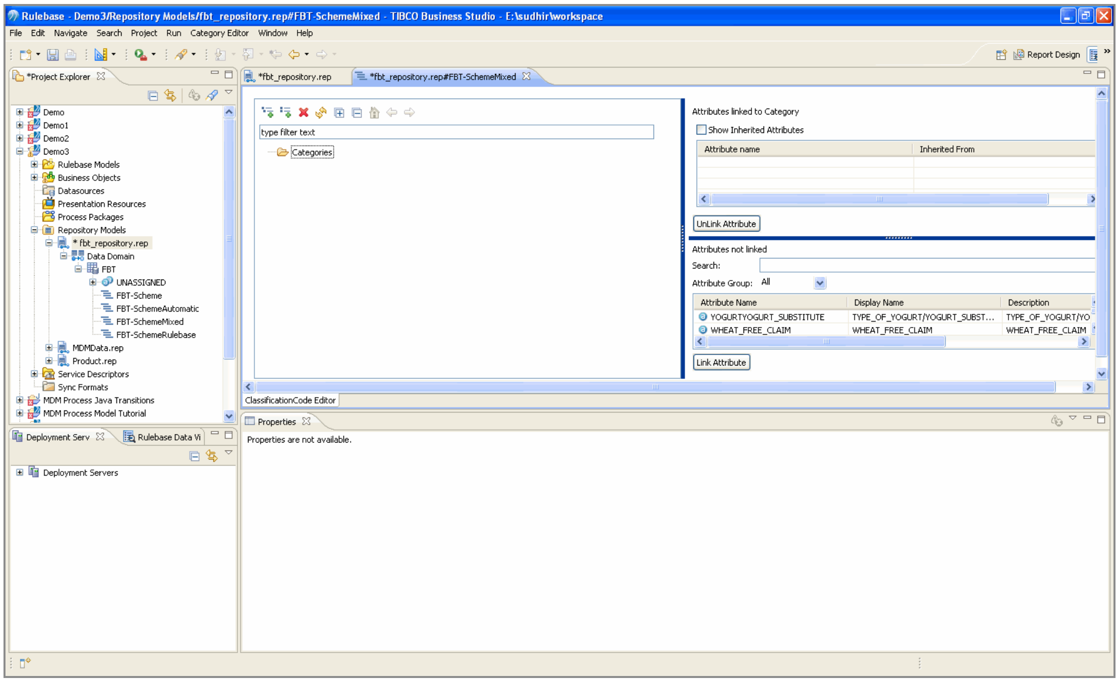


Level	Level Name
1	LogicalL1
2	LogicalL2
3	LogicalL3

8. To delete the attributes select the attributes and click ✖ .

9. Click **Save**.

10. Click **<open..>** in the **Open Editor** column. The newly created Classification Scheme is displayed in the Classification Scheme Editor. The Classification Scheme Editor is displayed on click open link.



Attribute name	Inherited From

Attribute Name	Display Name	Description
YOGURTYOGURT_SUBSTITUTE	TYPE_OF_YOGURT/YOGURT_SUBST...	TYPE_OF_YOGURT/YO
WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM

11. Using the Classification Scheme Editor you can create the classification code hierarchy. For more information on creating Classification Code, refer to Creating Classification Code Hierarchy.

Deleting Classification Scheme

Procedure

1. On Project Explorer, select the Classification Scheme and click ✖.

Classification Scheme Editor

The Classification Scheme is divided in three section:

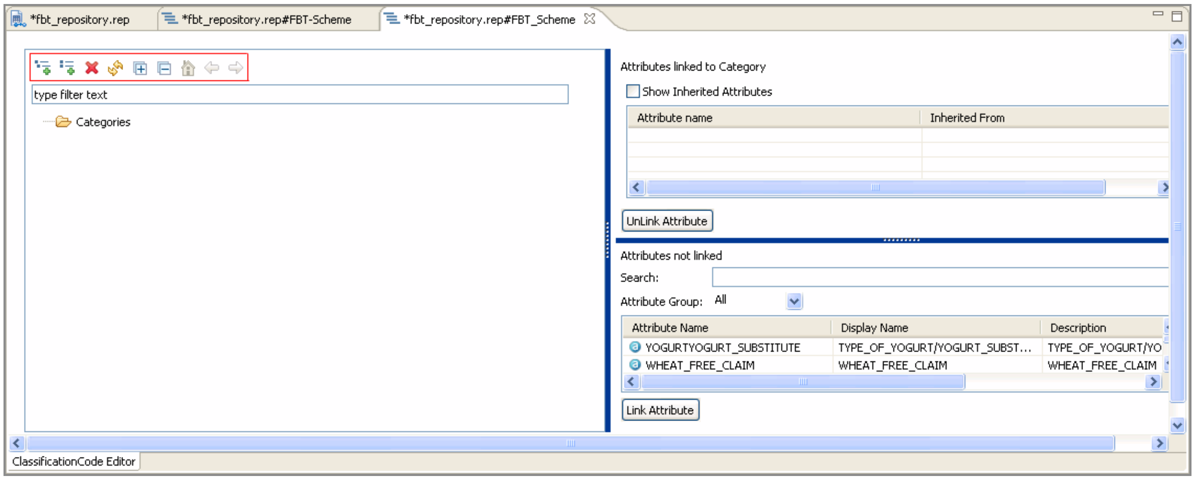
- Category Code Editor
- Linking/Unlinking Category Specific Attribute section
- Property Section

The Category Code Editor allows

- Adding child and sibling Classification Code.
- Importing of the Classification Code hierarchy from a CSV file.
- Inline editing of the Classification Code name on the tree node and Classification Code Property section allows editing of the code, name, and description.
- Linking and unlinking of Category Specific Attributes to Classification Code.
- Browsing and searching of Classification Code within the tree based on the name.
- Viewing of attached attributes and the inherited attributes.

Creating Classification Code Hierarchy

Using Classification Code tree controls you can create Classification Code Hierarchy.


*fbt_repository.rep
*fbt_repository.rep#FBT-Scheme
*fbt_repository.rep#FBT_Scheme
type filter text
Categories
Attributes linked to Category
☐ Show Inherited Attributes
UnLink Attribute
Attributes not linked
Search:
Attribute Group: All
Link Attribute
ClassificationCode Editor

Attribute name	Inherited From

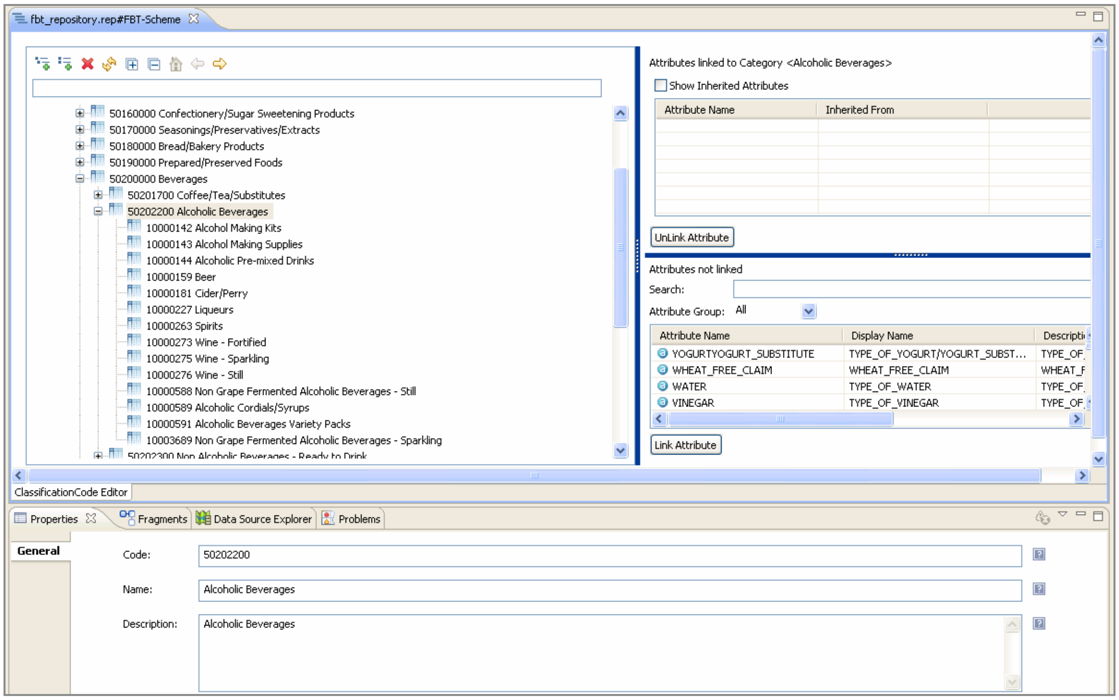
Attribute Name	Display Name	Description
YOGURTYOGURT_SUBSTITUTE	TYPE_OF_YOGURT/YOGURT_SUBST...	TYPE_OF_YOGURT/YO
WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM

Tree Controls	
Editing Controls	
⊕	Create a new child Classification Code.
⊕	Create a new sibling Classification Code.
✖	Delete a Classification Code.
Navigating Controls	
⟳	Refresh the Classification Code.
⊞	Expand to view all the Classification Code.
⊟	Collapse all the Classification Code.
⌂	Home Classification Code. If you are inside a child or sibling Classification Code and want to return the main Classification Code.
⇦	Back Classification Code. If you are inside a

	child or sibling Classification Code click and want to return back.
⇨ ⇨	GoInto Classification Code if you want view the child or sibling Classification Code.

Procedure

1. Click to add a child Classification Code. Enter an appropriate name for the child Classification Code.



fbt_repository.rep#FBT-Scheme

```
⊞ 50160000 Confectionery/Sugar Sweetening Products
⊞ 50170000 Seasonings/Preservatives/Extracts
⊞ 50180000 Bread/Bakery Products
⊞ 50190000 Prepared/Preserved Foods
⊟ 50200000 Beverages
    ⊞ 50201700 Coffee/Tea/Substitutes
    ⊟ 50202200 Alcoholic Beverages
        10000142 Alcohol Making Kits
        10000143 Alcohol Making Supplies
        10000144 Alcoholic Pre-mixed Drinks
        10000159 Beer
        10000181 Cider/Perry
        10000227 Liqueurs
        10000263 Spirits
        10000273 Wine - Fortified
        10000275 Wine - Sparkling
        10000276 Wine - Still
        10000588 Non Grape Fermented Alcoholic Beverages - Still
        10000589 Alcoholic Cordials/Syrups
        10000591 Alcoholic Beverages Variety Packs
        10003689 Non Grape Fermented Alcoholic Beverages - Sparkling
    ⊞ 50202300 Non Alcoholic Beverages - Ready to Drink
```

Attributes linked to Category <Alcoholic Beverages>

□ Show Inherited Attributes

Attribute Name	Inherited From

UnLink Attribute

Attributes not linked

Search:

Attribute Group: All

Attribute Name	Display Name	Descripti
YOGURTYOGURT_SUBSTITUTE	TYPE_OF_YOGURT/YOGURT_SUBST...	TYPE_OF
WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM	WHEAT_F
WATER	TYPE_OF_WATER	TYPE_OF
VINEGAR	TYPE_OF_VINEGAR	TYPE_OF

Link Attribute

ClassificationCode Editor

Code: 50202200

Name: Alcoholic Beverages

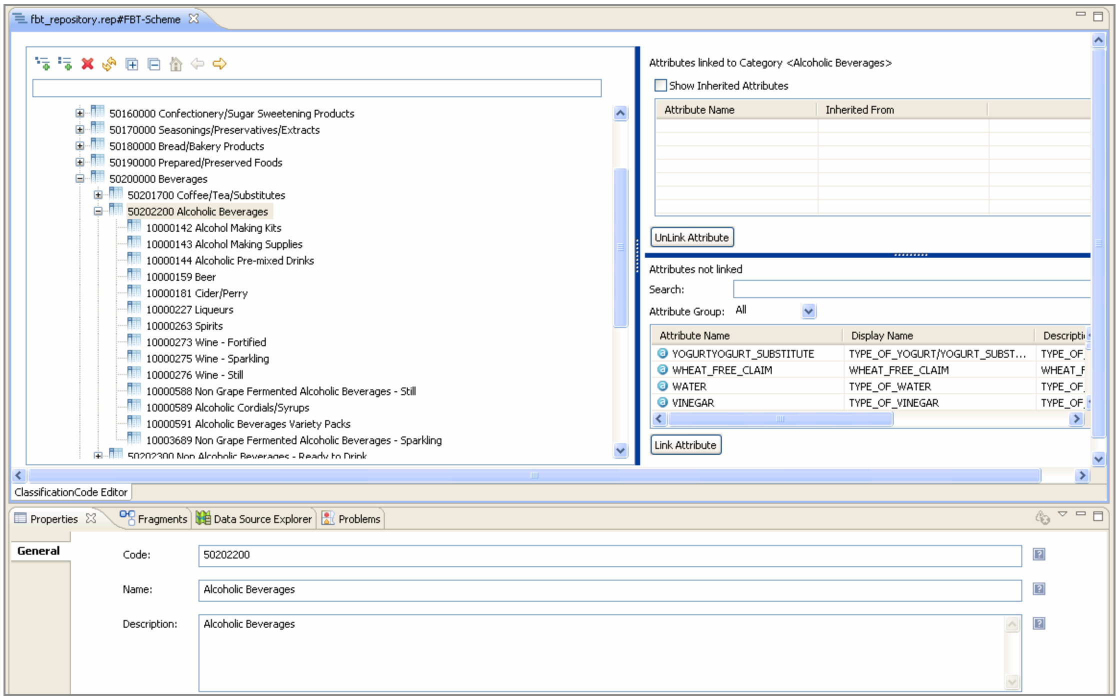
Description: Alcoholic Beverages

2. The properties for child Classification Code are displayed. Enter the **Code**, **Name**, and **Description** in the respective fields. You can add multiple child Classification Code.

3. Similarly click to add sibling Classification Code. Enter the **Code**, **Name**, and **Description** in the respective fields. You can create multiple siblings of an existing child.

4. Use the tree controls specified in the control table to Expand, Collapse, GoInto and so on.

5. The newly created Classification Code is displayed.



- 50160000 Confectionery/Sugar Sweetening Products
- 50170000 Seasonings/Preservatives/Extracts
- 50180000 Bread/Bakery Products
- 50190000 Prepared/Preserved Foods
- 50200000 Beverages
  - 50201700 Coffee/Tea/Substitutes
  - 50202200 Alcoholic Beverages
    - 10000142 Alcohol Making Kits
    - 10000143 Alcohol Making Supplies
    - 10000144 Alcoholic Pre-mixed Drinks
    - 10000159 Beer
    - 10000181 Cider/Perry
    - 10000227 Liqueurs
    - 10000263 Spirits
    - 10000273 Wine - Fortified
    - 10000275 Wine - Sparkling
    - 10000276 Wine - Still
    - 10000588 Non Grape Fermented Alcoholic Beverages - Still
    - 10000589 Alcoholic Cordials/Syrups
    - 10000591 Alcoholic Beverages Variety Packs
    - 10003689 Non Grape Fermented Alcoholic Beverages - Sparkling
  - 50202300 Non Alcoholic Beverages - Ready to Drink

Attributes linked to Category <Alcoholic Beverages>

□ Show Inherited Attributes

Attribute Name	Inherited From

Attributes not linked

Search:

Attribute Group: All

Attribute Name	Display Name	Descripti
YOGURTYOGURT_SUBSTITUTE	TYPE_OF_YOGURT/YOGURT_SUBST...	TYPE_OF_
WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM	WHEAT_F
WATER	TYPE_OF_WATER	TYPE_OF_
VINEGAR	TYPE_OF_VINEGAR	TYPE_OF_

General

Code: 50202200

Name: Alcoholic Beverages

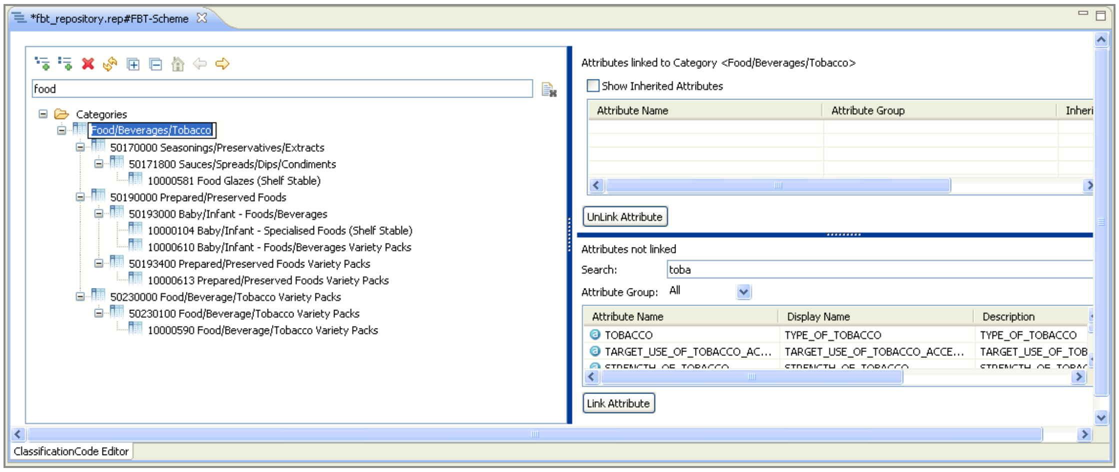
Description: Alcoholic Beverages

Editing Classification Code

Procedure

1. Select the Classification Code which you want to edit.

2. Double-click on the Classification Code name or use the F2 key. The inline text edit control allows you to modify the Classification Code name.



food

- Categories
  - Food/Beverages/Tobacco
    - 50170000 Seasonings/Preservatives/Extracts
      - 50171800 Sauces/Spreads/Dips/Condiments
        - 10000581 Food Glazes (Shelf Stable)
    - 50190000 Prepared/Preserved Foods
      - 50193000 Baby/Infant - Foods/Beverages
        - 10000104 Baby/Infant - Specialised Foods (Shelf Stable)
        - 10000610 Baby/Infant - Foods/Beverages Variety Packs
      - 50193400 Prepared/Preserved Foods Variety Packs
        - 10000613 Prepared/Preserved Foods Variety Packs
    - 50230000 Food/Beverage/Tobacco Variety Packs
      - 50230100 Food/Beverage/Tobacco Variety Packs
        - 10000590 Food/Beverage/Tobacco Variety Packs

Attributes linked to Category <Food/Beverages/Tobacco>

□ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri

Attributes not linked

Search: toba

Attribute Group: All

Attribute Name	Display Name	Description
TOBACCO	TYPE_OF_TOBACCO	TYPE_OF_TOBACCO
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC

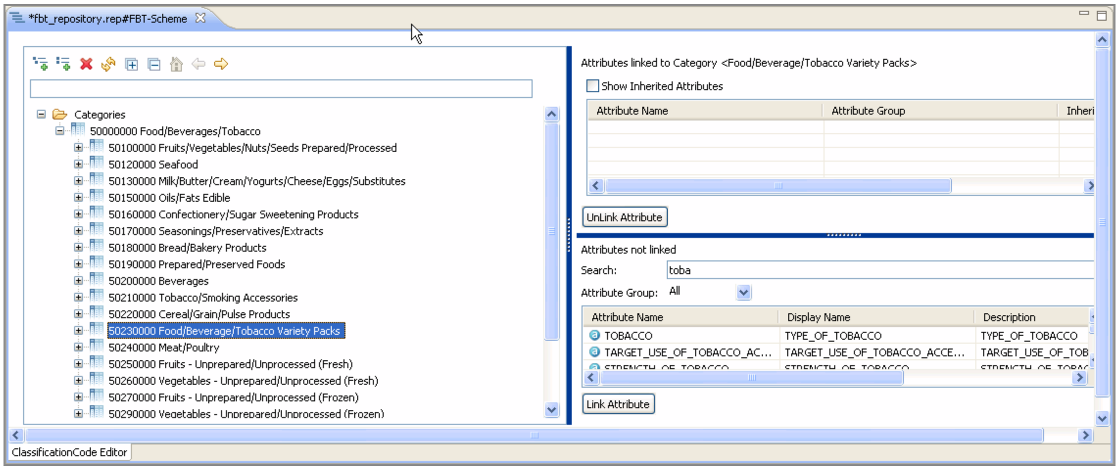
Dragging and Dropping of Classification Code Hierarchy

You can drag & drop the select Classification Codes to a Classification Code node in the tree.

Procedure

1. Select the Classification Code which you want to drag from the existing hierarchy and drop it to the target Classification Code. For example, drag the **Food/Beverage/Tobacco Variety Packs** Classification Code and drop it to another Classification Code.

   > **Note:** You cannot drag a parent node on the child node. If done, occasionally the dragged items become invisible. To restore the hierarchy, click
   >
   > ⟳.

   *fbt_repository.rep#FBT-Scheme

   Categories
   - 50000000 Food/Beverages/Tobacco
     - 50100000 Fruits/Vegetables/Nuts/Seeds Prepared/Processed
     - 50120000 Seafood
     - 50130000 Milk/Butter/Cream/Yogurts/Cheese/Eggs/Substitutes
     - 50150000 Oils/Fats Edible
     - 50160000 Confectionery/Sugar Sweetening Products
     - 50170000 Seasonings/Preservatives/Extracts
     - 50180000 Bread/Bakery Products
     - 50190000 Prepared/Preserved Foods
     - 50200000 Beverages
     - 50210000 Tobacco/Smoking Accessories
     - 50220000 Cereal/Grain/Pulse Products
     - 50230000 Food/Beverage/Tobacco Variety Packs
     - 50240000 Meat/Poultry
     - 50250000 Fruits - Unprepared/Unprocessed (Fresh)
     - 50260000 Vegetables - Unprepared/Unprocessed (Fresh)
     - 50270000 Fruits - Unprepared/Unprocessed (Frozen)
     - 50290000 Vegetables - Unprepared/Unprocessed (Frozen)

   Attributes linked to Category <Food/Beverage/Tobacco Variety Packs>

   □ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri

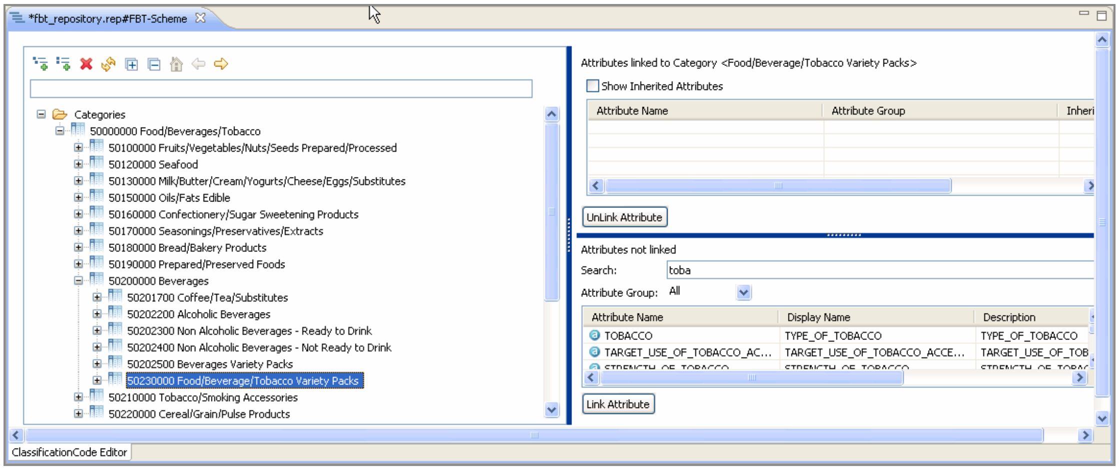
   Attributes not linked

   Search: toba

   Attribute Group: All

Attribute Name	Display Name	Description
TOBACCO	TYPE_OF_TOBACCO	TYPE_OF_TOBACCO
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC

2. Drop the **Food/Beverage/Tobacco Variety Packs** under the **Beverages** Classification Code. The whole branch is removed from the existing hierarchy and dragged hierarchy is formed under the target Classification Code.


*fbt_repository.rep#FBT-Scheme
Categories
50000000 Food/Beverages/Tobacco
50100000 Fruits/Vegetables/Nuts/Seeds Prepared/Processed
50120000 Seafood
50130000 Milk/Butter/Cream/Yogurts/Cheese/Eggs/Substitutes
50150000 Oils/Fats Edible
50160000 Confectionery/Sugar Sweetening Products
50170000 Seasonings/Preservatives/Extracts
50180000 Bread/Bakery Products
50190000 Prepared/Preserved Foods
50200000 Beverages
50201700 Coffee/Tea/Substitutes
50202200 Alcoholic Beverages
50202300 Non Alcoholic Beverages - Ready to Drink
50202400 Non Alcoholic Beverages - Not Ready to Drink
50202500 Beverages Variety Packs
50230000 Food/Beverage/Tobacco Variety Packs
50210000 Tobacco/Smoking Accessories
50220000 Cereal/Grain/Pulse Products
Attributes linked to Category <Food/Beverage/Tobacco Variety Packs>
Show Inherited Attributes
Attribute Name
Attribute Group
Inheri
UnLink Attribute
Attributes not linked
Link Attribute
ClassificationCode Editor

Search: toba

Attribute Group: All

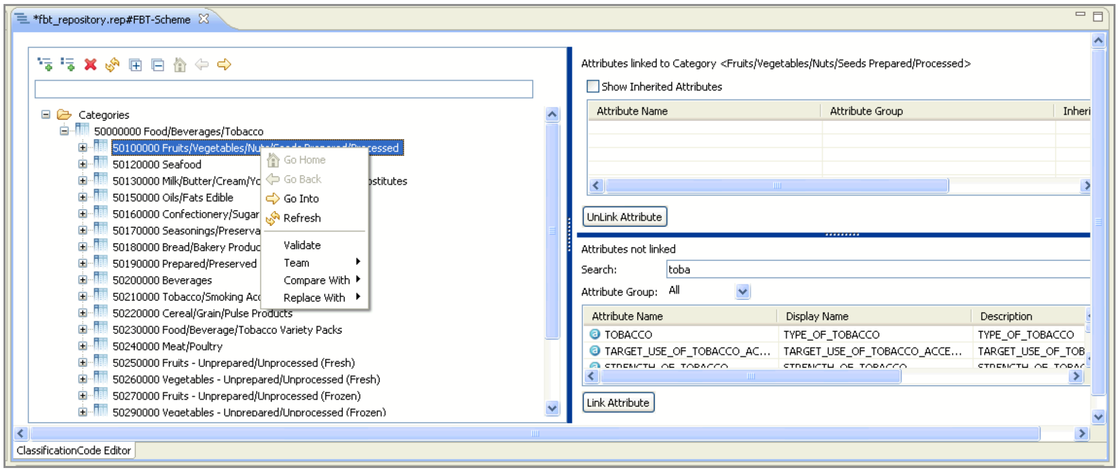
Attribute Name	Display Name	Description
TOBACCO	TYPE_OF_TOBACCO	TYPE_OF_TOBACCO
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC

Browsing Classification Code hierarchy

If you have many Classification Code, navigating the Classification Code is made easy by browsing a section of the Classification Code. Only the selected section or branch of the Classification Code is displayed.

Procedure

1. Select the Classification Code which you want to browse. For example, select **Fruits/Vegetables/Nuts/Seeds Prepared/Processed** Classification Code and right-click. You can also use the navigation controls on the tree control panel to navigate the Classification Code Hierarchy.

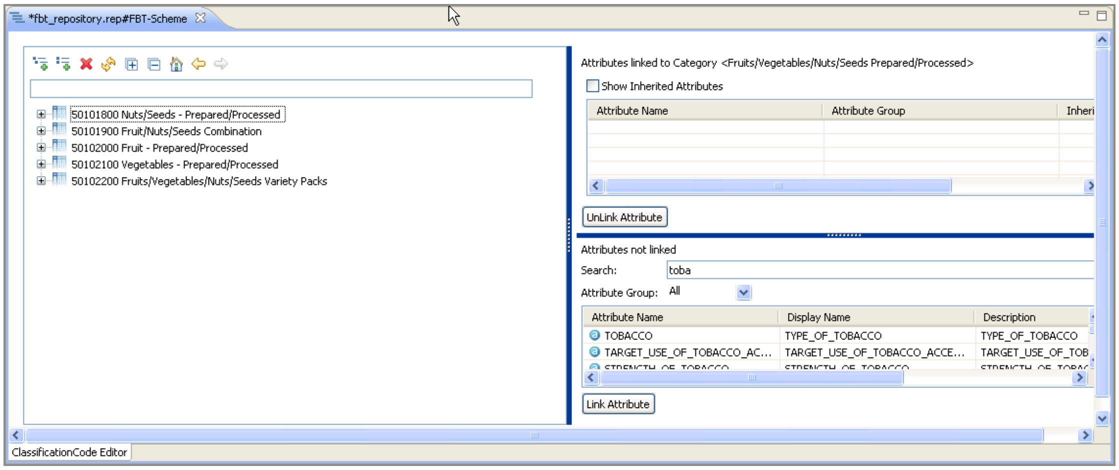

*fbt_repository.rep#FBT-Scheme
Categories
50000000 Food/Beverages/Tobacco
50100000 Fruits/Vegetables/Nuts/Seeds Prepared/Processed
50120000 Seafood
50130000 Milk/Butter/Cream/Yogurts/Cheese/Eggs/Substitutes
50150000 Oils/Fats Edible
50160000 Confectionery/Sugar
50170000 Seasonings/Preserva
50180000 Bread/Bakery Produc
50190000 Prepared/Preserved
50200000 Beverages
50210000 Tobacco/Smoking Acc
50220000 Cereal/Grain/Pulse Products
50230000 Food/Beverage/Tobacco Variety Packs
50240000 Meat/Poultry
50250000 Fruits - Unprepared/Unprocessed (Fresh)
50260000 Vegetables - Unprepared/Unprocessed (Fresh)
50270000 Fruits - Unprepared/Unprocessed (Frozen)
50290000 Vegetables - Unprepared/Unprocessed (Frozen)
Go Home
Go Back
Go Into
Refresh
Validate
Team
Compare With
Replace With
Attributes linked to Category <Fruits/Vegetables/Nuts/Seeds Prepared/Processed>
Show Inherited Attributes
Attribute Name
Attribute Group
Inheri
UnLink Attribute
Attributes not linked
Link Attribute
ClassificationCode Editor

Search: toba

Attribute Group: All

Attribute Name	Display Name	Description
TOBACCO	TYPE_OF_TOBACCO	TYPE_OF_TOBACCO
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC

2. Select **Go Into** from the right menu. Only the selected branch is displayed.



*fbt_repository.rep#FBT-Scheme

- 50101800 Nuts/Seeds - Prepared/Processed
- 50101900 Fruit/Nuts/Seeds Combination
- 50102000 Fruit - Prepared/Processed
- 50102100 Vegetables - Prepared/Processed
- 50102200 Fruits/Vegetables/Nuts/Seeds Variety Packs

Attributes linked to Category <Fruits/Vegetables/Nuts/Seeds Prepared/Processed>

□ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri

UnLink Attribute

Attributes not linked

Search: toba

Attribute Group: All

Attribute Name	Display Name	Description
TOBACCO	TYPE_OF_TOBACCO	TYPE_OF_TOBACCO
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC

Link Attribute

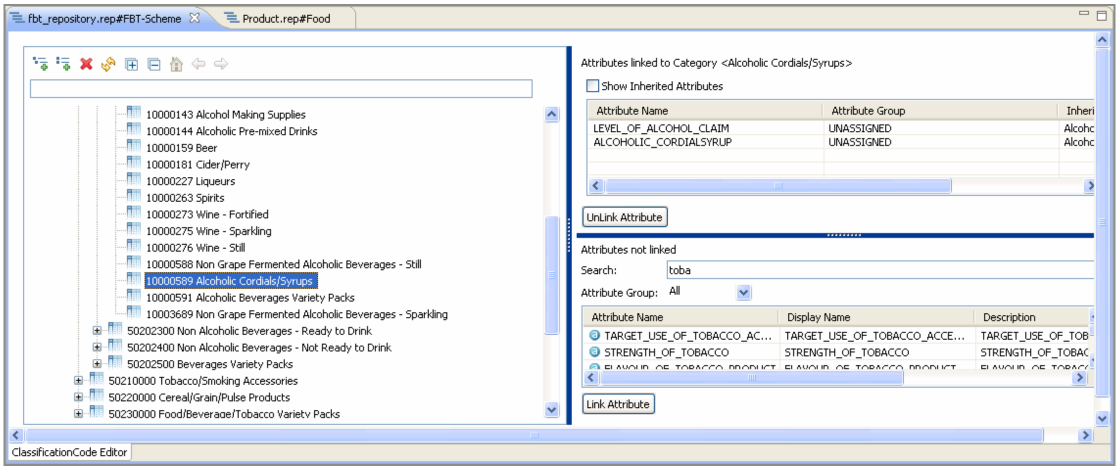
ClassificationCode Editor

Viewing Inherited Classification Code Attributes

You can view the inherited Classification Code attributes. The Shown Inherit Attribute check box allows to inherit all the Classification Code Attributes of its parent Classification Code.

Procedure

1. Select the Classification Code for which you want to view the inherited Classification Code. For example, select **Alcoholic Cordials/Syrups** the inherited Classification Code is displayed.



fbt_repository.rep#FBT-Scheme | Product.rep#Food

- 10000143 Alcohol Making Supplies
- 10000144 Alcoholic Pre-mixed Drinks
- 10000159 Beer
- 10000181 Cider/Perry
- 10000227 Liqueurs
- 10000263 Spirits
- 10000273 Wine - Fortified
- 10000275 Wine - Sparkling
- 10000276 Wine - Still
- 10000588 Non Grape Fermented Alcoholic Beverages - Still
- 10000589 Alcoholic Cordials/Syrups
- 10000591 Alcoholic Beverages Variety Packs
- 10003689 Non Grape Fermented Alcoholic Beverages - Sparkling
- 50202300 Non Alcoholic Beverages - Ready to Drink
- 50202400 Non Alcoholic Beverages - Not Ready to Drink
- 50202500 Beverages Variety Packs
- 50210000 Tobacco/Smoking Accessories
- 50220000 Cereal/Grain/Pulse Products
- 50230000 Food/Beverage/Tobacco Variety Packs

Attributes linked to Category <Alcoholic Cordials/Syrups>

□ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri
LEVEL_OF_ALCOHOL_CLAIM	UNASSIGNED	Alcohc
ALCOHOLIC_CORDIALSYRUP	UNASSIGNED	Alcohc

UnLink Attribute

Attributes not linked

Search: toba

Attribute Group: All

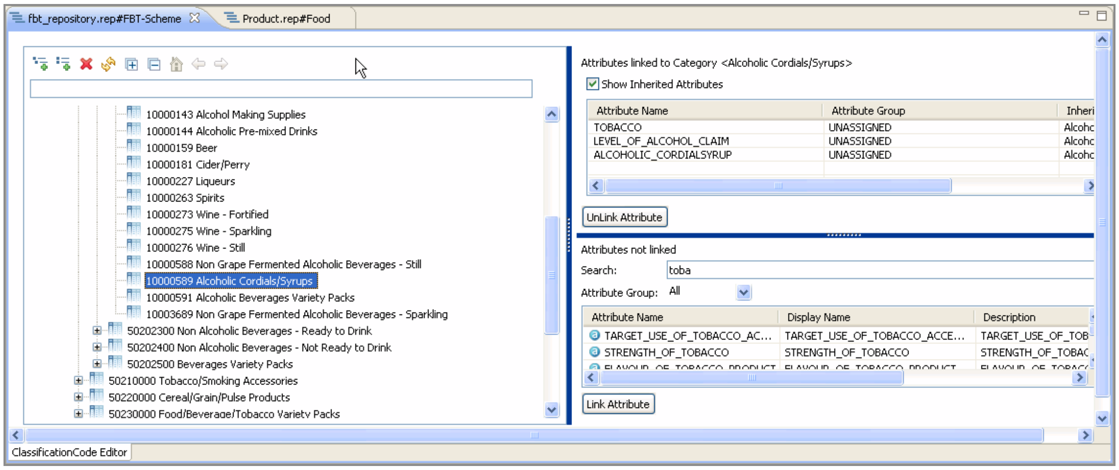
Attribute Name	Display Name	Description
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC
FLAVOUR_OF_TOBACCO_PRODUCT	FLAVOUR_OF_TOBACCO_PRODUCT	FLAVOUR_OF_TOBACC

Link Attribute

ClassificationCode Editor

2. Select the check box corresponding to **Shown Inherited Attributed**. The inherited attributes along with list Category Specific Attributes associated with the Classification Code (**Food/Beverages/Tobacco**) is displayed. The inherited attribute is

displayed in the Attribute column and you can see the inheritedFrom column to ascertain from where the attribute is inherited.



- 10000143 Alcohol Making Supplies
- 10000144 Alcoholic Pre-mixed Drinks
- 10000159 Beer
- 10000181 Cider/Perry
- 10000227 Liqueurs
- 10000263 Spirits
- 10000273 Wine - Fortified
- 10000275 Wine - Sparkling
- 10000276 Wine - Still
- 10000588 Non Grape Fermented Alcoholic Beverages - Still
- 10000589 Alcoholic Cordials/Syrups
- 10000591 Alcoholic Beverages Variety Packs
- 10003689 Non Grape Fermented Alcoholic Beverages - Sparkling
- 50202300 Non Alcoholic Beverages - Ready to Drink
- 50202400 Non Alcoholic Beverages - Not Ready to Drink
- 50202500 Beverages Variety Packs
- 50210000 Tobacco/Smoking Accessories
- 50220000 Cereal/Grain/Pulse Products
- 50230000 Food/Beverage/Tobacco Variety Packs

Attributes linked to Category <Alcoholic Cordials/Syrups>

☑ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri
TOBACCO	UNASSIGNED	Alcohc
LEVEL_OF_ALCOHOL_CLAIM	UNASSIGNED	Alcohc
ALCOHOLIC_CORDIALSYRUP	UNASSIGNED	Alcohc

Attributes not linked

Search: toba

Attribute Group: All

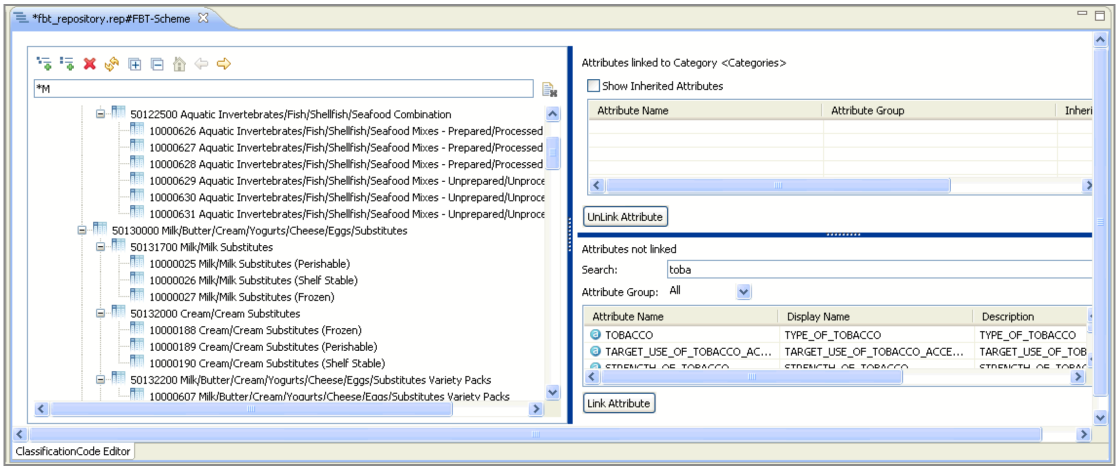
Attribute Name	Display Name	Description
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC
FLAVOUR_OF_TOBACCO_PRODUCT	FLAVOUR_OF_TOBACCO_PRODUCT	FLAVOUR_OF_TOBACC

Searching Classification Code Hierarchy

You can search for a Classification Code using the pattern matching string as the search criteria.

Procedure

1. Type the pattern matching string in the search field. For example, on typing *M displays Classification Code names starting with anything and having a letter M.
2. To clear the search criteria click .



*M

- 50122500 Aquatic Invertebrates/Fish/Shellfish/Seafood Combination
  - 10000626 Aquatic Invertebrates/Fish/Shellfish/Seafood Mixes - Prepared/Processed
  - 10000627 Aquatic Invertebrates/Fish/Shellfish/Seafood Mixes - Prepared/Processed
  - 10000628 Aquatic Invertebrates/Fish/Shellfish/Seafood Mixes - Prepared/Processed
  - 10000629 Aquatic Invertebrates/Fish/Shellfish/Seafood Mixes - Unprepared/Unproce
  - 10000630 Aquatic Invertebrates/Fish/Shellfish/Seafood Mixes - Unprepared/Unproce
  - 10000631 Aquatic Invertebrates/Fish/Shellfish/Seafood Mixes - Unprepared/Unproce
- 50130000 Milk/Butter/Cream/Yogurts/Cheese/Eggs/Substitutes
  - 50131700 Milk/Milk Substitutes
    - 10000025 Milk/Milk Substitutes (Perishable)
    - 10000026 Milk/Milk Substitutes (Shelf Stable)
    - 10000027 Milk/Milk Substitutes (Frozen)
  - 50132000 Cream/Cream Substitutes
    - 10000188 Cream/Cream Substitutes (Frozen)
    - 10000189 Cream/Cream Substitutes (Perishable)
    - 10000190 Cream/Cream Substitutes (Shelf Stable)
  - 50132200 Milk/Butter/Cream/Yogurts/Cheese/Eggs/Substitutes Variety Packs
    - 10000607 Milk/Butter/Cream/Yogurts/Cheese/Eggs/Substitutes Variety Packs

Attributes linked to Category <Categories>

☐ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri

Attributes not linked

Search: toba

Attribute Group: All

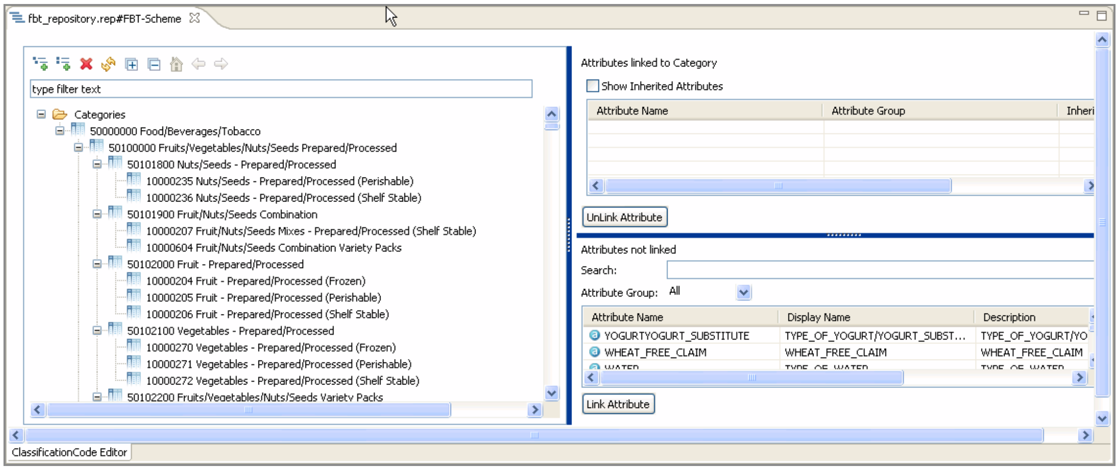
Attribute Name	Display Name	Description
TOBACCO	TYPE_OF_TOBACCO	TYPE_OF_TOBACCO
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC

Linking Category Specific Attributes with Classification Code

Using this option you can link defined Category Specific Attributes to category code.

Procedure

1. Select the Category to link the attributes.


fbt_repository.rep#FBT-Scheme
type filter text
Attributes linked to Category
☐ Show Inherited Attributes
UnLink Attribute
Attributes not linked
Link Attribute
ClassificationCode Editor

```
Categories
  50000000 Food/Beverages/Tobacco
    50100000 Fruits/Vegetables/Nuts/Seeds Prepared/Processed
      50101800 Nuts/Seeds - Prepared/Processed
        10000235 Nuts/Seeds - Prepared/Processed (Perishable)
        10000236 Nuts/Seeds - Prepared/Processed (Shelf Stable)
      50101900 Fruit/Nuts/Seeds Combination
        10000207 Fruit/Nuts/Seeds Mixes - Prepared/Processed (Shelf Stable)
        10000604 Fruit/Nuts/Seeds Combination Variety Packs
      50102000 Fruit - Prepared/Processed
        10000204 Fruit - Prepared/Processed (Frozen)
        10000205 Fruit - Prepared/Processed (Perishable)
        10000206 Fruit - Prepared/Processed (Shelf Stable)
      50102100 Vegetables - Prepared/Processed
        10000270 Vegetables - Prepared/Processed (Frozen)
        10000271 Vegetables - Prepared/Processed (Perishable)
        10000272 Vegetables - Prepared/Processed (Shelf Stable)
      50102200 Fruits/Vegetables/Nuts/Seeds Variety Packs
```

Attribute Name	Attribute Group	Inheri

Search:

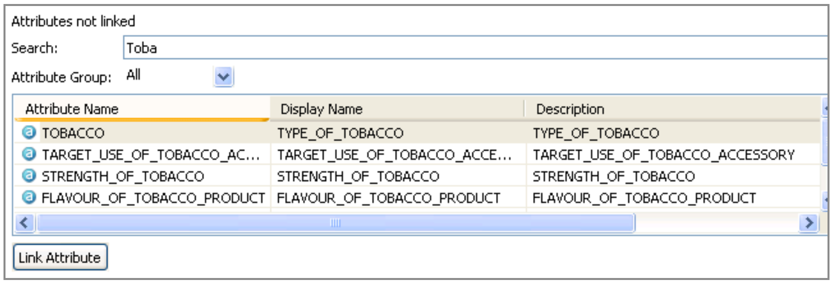
Attribute Group: All

Attribute Name	Display Name	Description
YOGURTYOGURT_SUBSTITUTE	TYPE_OF_YOGURT/YOGURT_SUBST...	TYPE_OF_YOGURT/YO
WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM
WATER	TYPE_OF_WATER	TYPE_OF_WATER

2. Select the Classification Code to which you want to link the Category Specific Attribute. The table displays only the Category Specific Attributes, normal attributes are not displayed in the table.

3. In the **Attributes not linked** section, select **All** from the **Attribute Group** drop-down. This is used as a filter, all the Category Specific Attributes in the repository are displayed.

4. Search for the category specific attribute by specifying the pattern based search in the **Search** field. For example, if you type "Toba" in the search field all the attributes containing the text "Toba" is displayed.

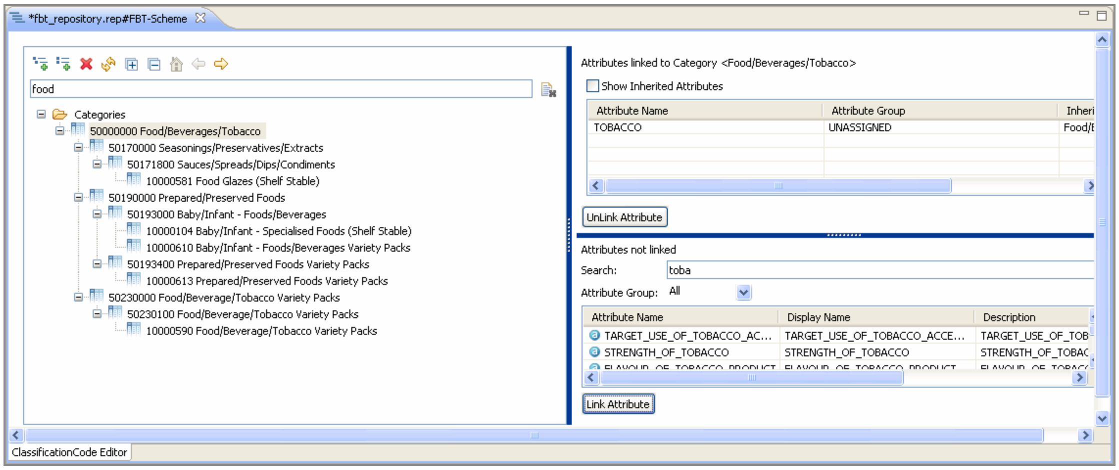


Search: Toba

Attribute Group: All

Attribute Name	Display Name	Description
TOBACCO	TYPE_OF_TOBACCO	TYPE_OF_TOBACCO
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOBACCO_ACCESSORY
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO
FLAVOUR_OF_TOBACCO_PRODUCT	FLAVOUR_OF_TOBACCO_PRODUCT	FLAVOUR_OF_TOBACCO_PRODUCT

5. Select the Category Specific Attribute and click Link Attribute.
6. You can select one or more than one CSA attributes to link. For example, Tobacco Category Specific Attribute is linked to Food/Beverages/Tobacco Classification Code. Continue to search for the categories and link it the Classification Code.



- Categories
  - 50000000 Food/Beverages/Tobacco
    - 50170000 Seasonings/Preservatives/Extracts
      - 50171800 Sauces/Spreads/Dips/Condiments
        - 10000581 Food Glazes (Shelf Stable)
    - 50190000 Prepared/Preserved Foods
      - 50193000 Baby/Infant - Foods/Beverages
        - 10000104 Baby/Infant - Specialised Foods (Shelf Stable)
        - 10000610 Baby/Infant - Foods/Beverages Variety Packs
      - 50193400 Prepared/Preserved Foods Variety Packs
        - 10000613 Prepared/Preserved Foods Variety Packs
    - 50230000 Food/Beverage/Tobacco Variety Packs
      - 50230100 Food/Beverage/Tobacco Variety Packs
        - 10000590 Food/Beverage/Tobacco Variety Packs

Attributes linked to Category <Food/Beverages/Tobacco>

□ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri
TOBACCO	UNASSIGNED	Food/E

Attributes not linked

Search: toba

Attribute Group: All

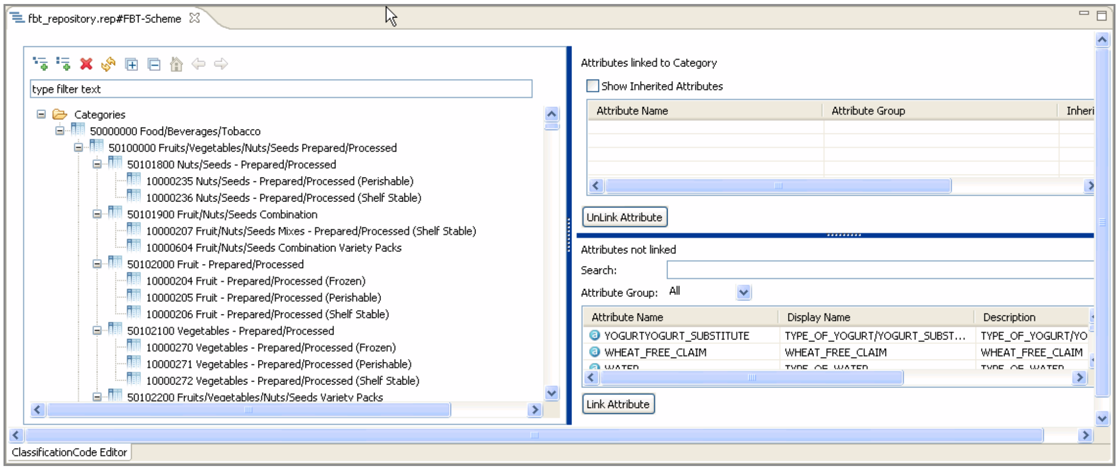
Attribute Name	Display Name	Description
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC
FLAVOUR_OF_TOBACCO_PRODUCT	FLAVOUR_OF_TOBACCO_PRODUCT	FLAVOUR_OF_TOBACC

Unlinking Category Specific Attributes from Classification Code

Using this option you can unlink the Category Specific Attributes from the Classification Code.

Procedure

1. Select the Category to unlink the Category Specific Attribute.



- Categories
  - 50000000 Food/Beverages/Tobacco
    - 50100000 Fruits/Vegetables/Nuts/Seeds Prepared/Processed
      - 50101800 Nuts/Seeds - Prepared/Processed
        - 10000235 Nuts/Seeds - Prepared/Processed (Perishable)
        - 10000236 Nuts/Seeds - Prepared/Processed (Shelf Stable)
      - 50101900 Fruit/Nuts/Seeds Combination
        - 10000207 Fruit/Nuts/Seeds Mixes - Prepared/Processed (Shelf Stable)
        - 10000604 Fruit/Nuts/Seeds Combination Variety Packs
      - 50102000 Fruit - Prepared/Processed
        - 10000204 Fruit - Prepared/Processed (Frozen)
        - 10000205 Fruit - Prepared/Processed (Perishable)
        - 10000206 Fruit - Prepared/Processed (Shelf Stable)
      - 50102100 Vegetables - Prepared/Processed
        - 10000270 Vegetables - Prepared/Processed (Frozen)
        - 10000271 Vegetables - Prepared/Processed (Perishable)
        - 10000272 Vegetables - Prepared/Processed (Shelf Stable)
      - 50102200 Fruits/Vegetables/Nuts/Seeds Variety Packs

Attributes linked to Category

□ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri

Attributes not linked

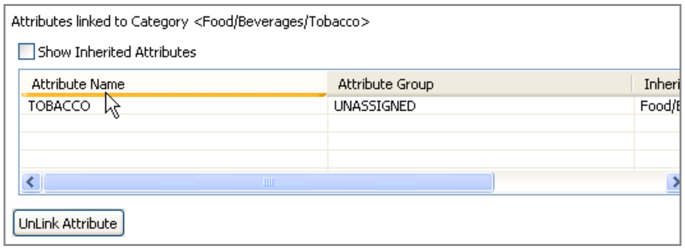
Search:

Attribute Group: All

Attribute Name	Display Name	Description
YOGURTYOGURT_SUBSTITUTE	TYPE_OF_YOGURT/YOGURT_SUBST...	TYPE_OF_YOGURT/YO
WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM	WHEAT_FREE_CLAIM
WATER	TYPE_OF_WATER	TYPE_OF_WATER

2. Select the Classification Code from which you want to unlink the Category Specific Attribute. For example, Tobacco Category Specific Attribute is unlinked from Food/Beverages/Tobacco Classification Code.

3. In the **Attributes linked to Category** section, the Category Specific Attributes linked to Classification Code is displayed.



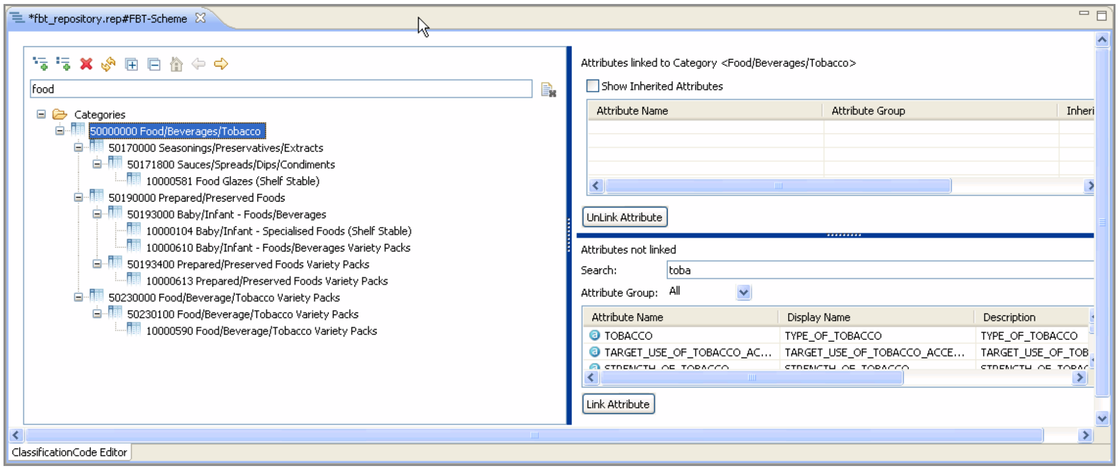
Attributes linked to Category <Food/Beverages/Tobacco>

□ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri
TOBACCO	UNASSIGNED	Food/E

4. Select the Category Specific Attribute and click UnLink Attribute.

5. You can select one or more than one CSA attributes to unlink. You cannot unlink inherited attributes, you have to select the inherited Classification Code and remove the Linked Attribute.



food

- Categories
  - 50000000 Food/Beverages/Tobacco
    - 50170000 Seasonings/Preservatives/Extracts
      - 50171800 Sauces/Spreads/Dips/Condiments
        - 10000581 Food Glazes (Shelf Stable)
    - 50190000 Prepared/Preserved Foods
      - 50193000 Baby/Infant - Foods/Beverages
        - 10000104 Baby/Infant - Specialised Foods (Shelf Stable)
        - 10000610 Baby/Infant - Foods/Beverages Variety Packs
      - 50193400 Prepared/Preserved Foods Variety Packs
        - 10000613 Prepared/Preserved Foods Variety Packs
    - 50230000 Food/Beverage/Tobacco Variety Packs
      - 50230100 Food/Beverage/Tobacco Variety Packs
        - 10000590 Food/Beverage/Tobacco Variety Packs

Attributes linked to Category <Food/Beverages/Tobacco>

□ Show Inherited Attributes

Attribute Name	Attribute Group	Inheri

Attributes not linked

Search: toba

Attribute Group: All

Attribute Name	Display Name	Description
TOBACCO	TYPE_OF_TOBACCO	TYPE_OF_TOBACCO
TARGET_USE_OF_TOBACCO_AC...	TARGET_USE_OF_TOBACCO_ACCE...	TARGET_USE_OF_TOB
STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBACCO	STRENGTH_OF_TOBAC

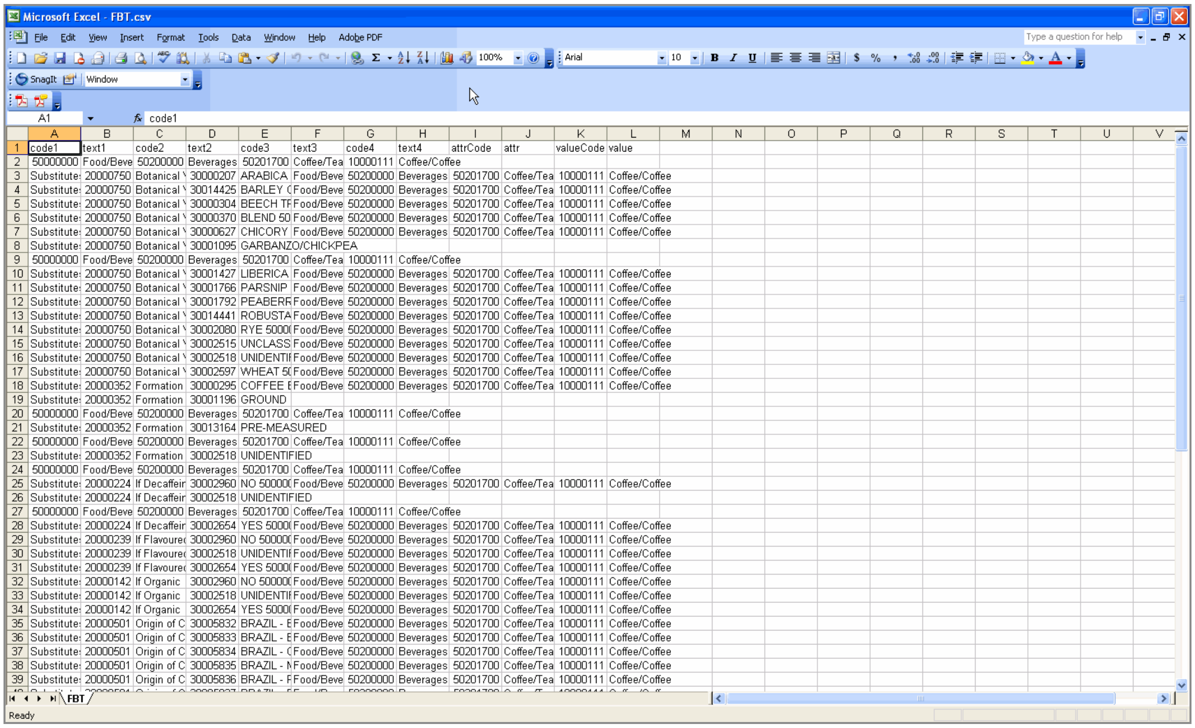
6. The unlinked Category Specific Attribute is removed from the **Attributes > linked to Category** section. The unlinked Category Specific Attribute is displayed in table in the **Attribute not linked**.

Importing Classification Code Hierarchy

In addition to creating the Classification Code using the Classification Scheme tree control, you can also import the existing Classification Code hierarchy from the CSV files into the Classification Scheme.

The CSV file format should contain the hierarchy level along with their code and description. The CSV file should be of specific format. The headers should be grouped as Columnsets. The columnset can have a value either 2 or 3. For example, code name desc or code, name. The Classification-code hierarchy is defined by adding the parent and child relationship of the ClassificationCode in the CSV file specified as a columnset. The Classification-code attribute can be specified after the Classification-code columns e.g. code1, name1, desc1, code2, name2, desc2, attributeName.

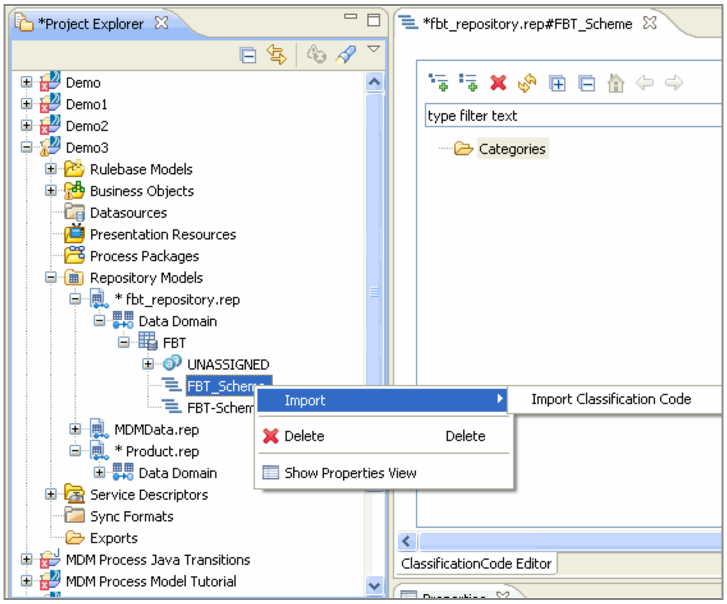
Below is a sample CSV format with column set as 2.


Microsoft Excel - FBT.csv
File Edit View Insert Format Tools Data Window Help Adobe PDF
100%
Arial
10
Type a question for help
SnagIt
Window
A1
code1
FBT
Ready

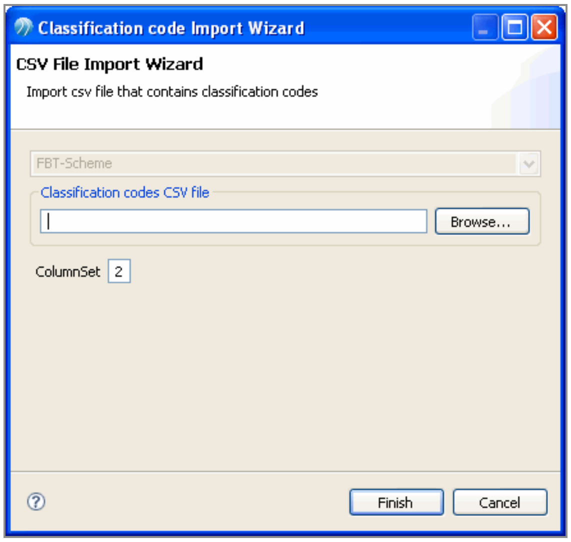
	A	B	C	D	E	F	G	H	I	J	K	L
1	code1	text1	code2	text2	code3	text3	code4	text4	attrCode	attr	valueCode	value
2	50000000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee				
3	Substitute	20000750	Botanical	30000207	ARABICA	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
4	Substitute	20000750	Botanical	30014425	BARLEY (	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
5	Substitute	20000750	Botanical	30000304	BEECH TF	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
6	Substitute	20000750	Botanical	30000370	BLEND 50	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
7	Substitute	20000750	Botanical	30000627	CHICORY	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
8	Substitute	20000750	Botanical	30001095	GARBANZO/CHICKPEA							
9	50000000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee				
10	Substitute	20000750	Botanical	30001427	LIBERICA	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
11	Substitute	20000750	Botanical	30001766	PARSNIP	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
12	Substitute	20000750	Botanical	30001792	PEABERR	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
13	Substitute	20000750	Botanical	30014441	ROBUSTA	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
14	Substitute	20000750	Botanical	30002080	RYE 5000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
15	Substitute	20000750	Botanical	30002515	UNCLASS	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
16	Substitute	20000750	Botanical	30002518	UNIDENTI	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
17	Substitute	20000750	Botanical	30002597	WHEAT 50	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
18	Substitute	20000352	Formation	30000295	COFFEE E	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
19	Substitute	20000352	Formation	30001196	GROUND							
20	50000000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee				
21	Substitute	20000352	Formation	30013164	PRE-MEASURED							
22	50000000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee				
23	Substitute	20000352	Formation	30002518	UNIDENTIFIED							
24	50000000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee				
25	Substitute	20000224	If Decaffeir	30002960	NO 50000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
26	Substitute	20000224	If Decaffeir	30002518	UNIDENTIFIED							
27	50000000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee				
28	Substitute	20000224	If Decaffeir	30002654	YES 5000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
29	Substitute	20000239	If Flavoure	30002960	NO 50000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
30	Substitute	20000239	If Flavoure	30002518	UNIDENTI	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
31	Substitute	20000239	If Flavoure	30002654	YES 5000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
32	Substitute	20000142	If Organic	30002960	NO 50000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
33	Substitute	20000142	If Organic	30002518	UNIDENTI	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
34	Substitute	20000142	If Organic	30002654	YES 5000	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
35	Substitute	20000501	Origin of C	30005832	BRAZIL - E	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
36	Substitute	20000501	Origin of C	30005833	BRAZIL - E	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
37	Substitute	20000501	Origin of C	30005834	BRAZIL - (	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
38	Substitute	20000501	Origin of C	30005835	BRAZIL - N	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee
39	Substitute	20000501	Origin of C	30005836	BRAZIL - F	Food/Beve	50200000	Beverages	50201700	Coffee/Tea	10000111	Coffee/Coffee

Procedure

1. Right-click the newly created Classification Scheme, select **Import > Import > Classification Code**.



2. The CSV file import Wizard is displayed.


Classification code Import Wizard
CSV File Import Wizard
Import csv file that contains classification codes
FBT-Scheme
Classification codes CSV file
Browse...
ColumnSet 2
Finish
Cancel

3. Click **Browse** to navigate to the path where you have saved the CSV file.
4. Define the **ColumnSet** field based on selected CSV column set definition.
5. Click **Finish**.
6. The Classification Code hierarchy is created. If the Classification Codes are already present then they are merged.

Deploy Classification Scheme

The Classification Scheme are created for an existing repository model and contains all the necessary information. When a repository model (.rep file) is deployed, all the corresponding Classification Schemes in the repository are also deployed.

For more information on how to deploy Classification Scheme, see Deployment.

Perspective Editor

This chapter introduces you to the new wizard for creating perspectives for a repository in TIBCO MDM Studio and also allows you to deploy the perspective in the TIBCO MDM Server.

Perspective Editor Overview

In TIBCO MDM server, a repository may consists of large number of attributes, multiple relationships defined with other repositories, and relationship attributes. However you may be interested in only a subset of the attributes and the relationships to perform various record operations. Navigating through the entire repository metadata is a significant overhead and adversely affects the performance.

Using the TIBCO MDM Studio wizard you can define a subset of the attributes, the relationship attributes and the relationships of a repository during the design time. This subset view is called 'Perspective'.

Once the perspectives are defined for a repository, it is deployed to TIBCO MDM server. In addition the perspectives are also saved to the database. The perspectives defined using the TIBCO MDM Studio are viewed using the TIBCO MDM server User Interface. The records and related records added to a repository are browsed based on the perspectives that are defined for the repository.

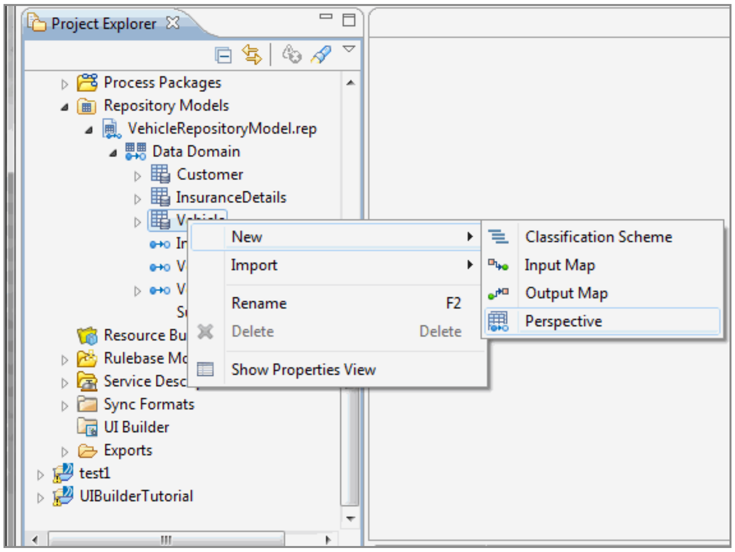
Note: Perspectives will not be migrated or imported from older versions. Perspectives must be recreated after migration.

Creating a Perspective

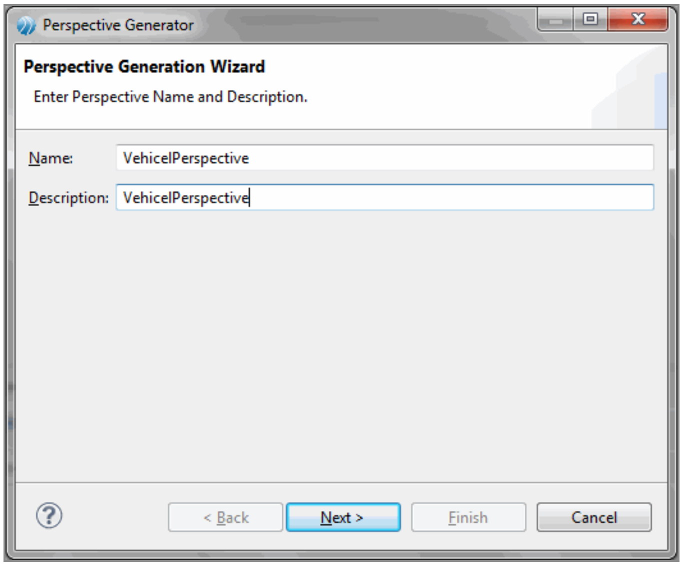
You can define a subset for the attributes , attribute groups, relationships, and relationship attributes in the root repository or target repository.

Procedure

1. In the Project Explorer under the Repository model file, Right-click the Repository on which user want to create a perspective and select **New-- > Perspective**.



2. The Perspective Generator Wizard is displayed.

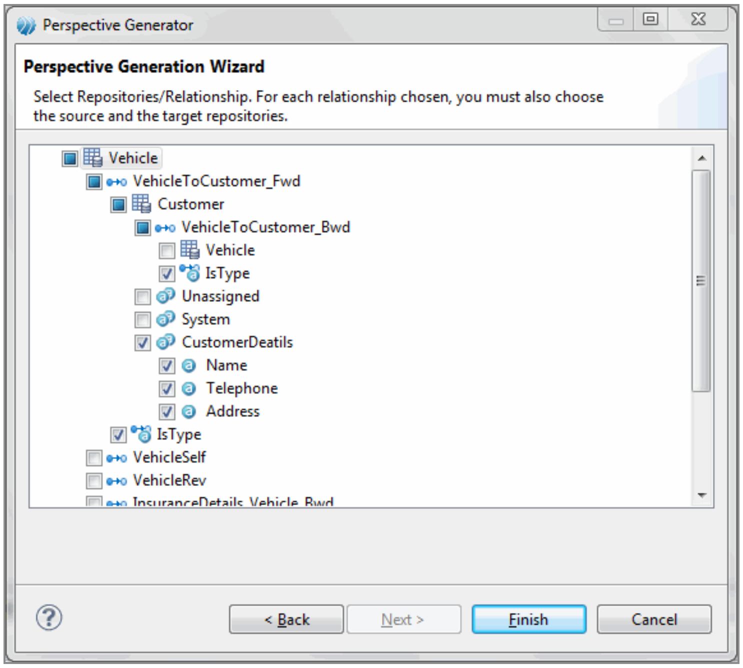

Perspective Generator
Perspective Generation Wizard
Enter Perspective Name and Description.
< Back
Next >
Finish
Cancel

Name: VehicelPerspective

Description: VehicelPerspective

3. Enter the Perspective name in the **Name** field and enter the description in the **Description** field.

4. Click **Next**.



Perspective Generator

Perspective Generation Wizard

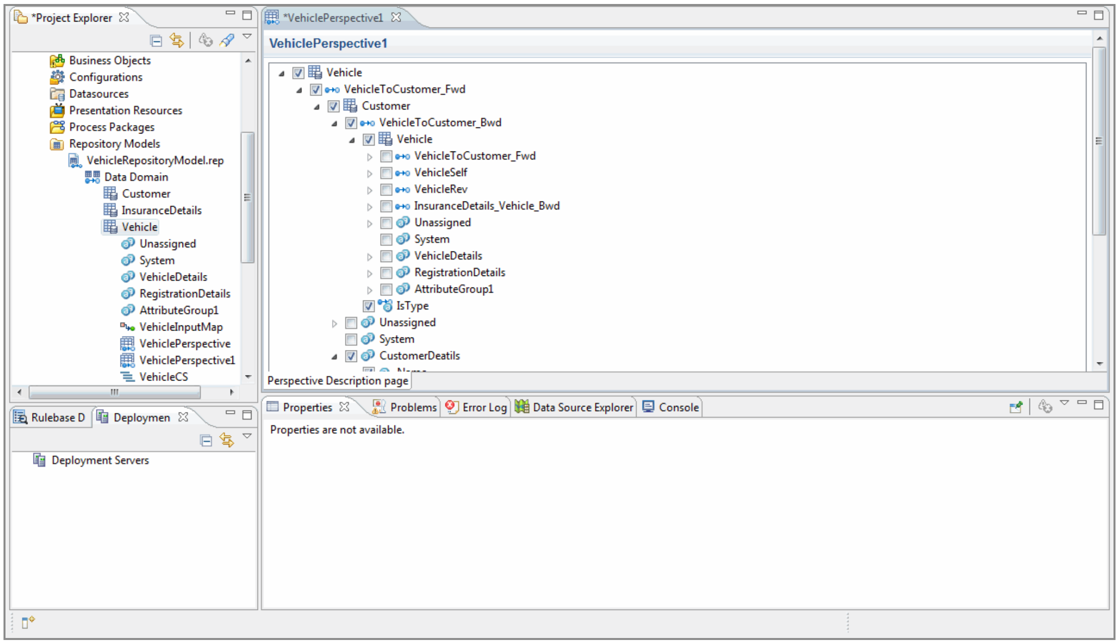
Select Repositories/Relationship. For each relationship chosen, you must also choose the source and the target repositories.

- ▣ Vehicle
  - ▣ VehicleToCustomer_Fwd
    - ▣ Customer
      - ▣ VehicleToCustomer_Bwd
        - □ Vehicle
        - ☑ IsType
      - □ Unassigned
      - □ System
      - ☑ CustomerDeatils
        - ☑ Name
        - ☑ Telephone
        - ☑ Address
    - ☑ IsType
  - □ VehicleSelf
  - □ VehicleRev
  - □ InsuranceDetails_Vehicle_Bwd

5. Select the relationships, attribute group or the individual attributes, or relationship attributes which you want to view in the perspective.

6. Click **Finish**.

7. The Perspective Editor with the newly defined perspective is displayed.



*Project Explorer

- Business Objects
- Configurations
- Datasources
- Presentation Resources
- Process Packages
- Repository Models
  - VehicleRepositoryModel.rep
    - Data Domain
      - Customer
      - InsuranceDetails
      - Vehicle
        - Unassigned
        - System
        - VehicleDetails
        - RegistrationDetails
        - AttributeGroup1
        - VehicleInputMap
        - VehiclePerspective
        - VehiclePerspective1
        - VehicleCS

Rulebase D | Deploymen

- Deployment Servers

*VehiclePerspective1

VehiclePerspective1

- ☑ Vehicle
  - ☑ VehicleToCustomer_Fwd
    - ☑ Customer
      - ☑ VehicleToCustomer_Bwd
        - ☑ Vehicle
          - □ VehicleToCustomer_Fwd
          - □ VehicleSelf
          - □ VehicleRev
          - □ InsuranceDetails_Vehicle_Bwd
          - □ Unassigned
          - □ System
          - □ VehicleDetails
          - □ RegistrationDetails
          - □ AttributeGroup1
        - ☑ IsType
      - □ Unassigned
      - □ System
      - ☑ CustomerDeatils

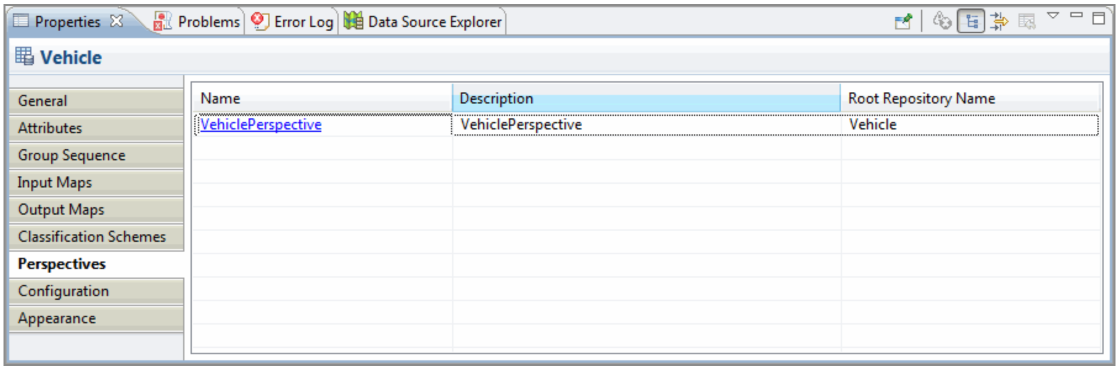
Perspective Description page

Properties | Problems | Error Log | Data Source Explorer | Console

Properties are not available.

8. The Perspective Editor displays the tree representation of the perspective. The tree representation is not only accurate representation of your selection of the attributes, relationship attributes and the relationships but also provides better and more effective way of editing the perspective metadata after creating the perspective.

9. In the Properties section of the Repository model all the defined perspectives are displayed.



Name	Description	Root Repository Name
VehiclePerspective	VehiclePerspective	Vehicle

10. Click the perspective link to view the perspective in the Perspective Editor.

Deploying a Perspective

The perspectives are created for an existing repository model and contains all the necessary information. When a repository model (.rep file) is deployed, all the corresponding perspectives in the repository are also deployed.

In addition, a set of selected perspectives can also be deployed individually by incremental deployment. For more information on how to deploy perspective see, Deployment.

Resource Bundle

A new editor is added to customize the resource bundle in TIBCO MDM Studio. After customizing the resource bundle, it is deployed in TIBCO MDM Server.

Resource Bundle Overview

TIBCO MDM supports multiple locale resource bundles. There are five resource bundles to localize or externalize the text within different components.

- htmlresources.properties: Resource bundle that contains static strings from all HTML pages in the UTF-8 format.
- jsresources.properties: Resource bundle that contains strings from JavaScript files and the JavaScript section of the HTML files.
- SharedDBStringResources.properties: Resource bundle that contains user interface strings from selected tables of the database.
- SharedStringResources.properties: Resource bundle that contains:
    - Display text of all menu items
    - All retailer or supplier terminologies
    - Descriptions and variable names defined in business process rules, rulebase and workflows
    - Any generic text displayed on UI
    - User-specified text which needs translation and is displayed on UI
- UserText.properties: Resource bundle that contains dynamically created informational, error, or warning messages displayed on the user interface.

All the default resource bundles are bundled within the ECM.ear in ECMClasses.jar file. The resource values can be customized by overriding the default resource value. You must create a new custom*.properties file and copy it under `$MQ_HOME/custom/resource` folder for WebLogic and WebSpehere and for JBoss copy it under `JBOSS_HOME\\modules\com\tibco\mdm\main\com\tibco\mdm\properties\mdm`.

Since user on the cloud do not have access to the TIBCO MDM Server, it is difficult to deploy the customizations. Hence a new editor to customized resource files is supported in TIBCO MDM Studio. After customization, the resource bundle is deployed on the server.

Limitations

The following are the limitations of resource bundle:

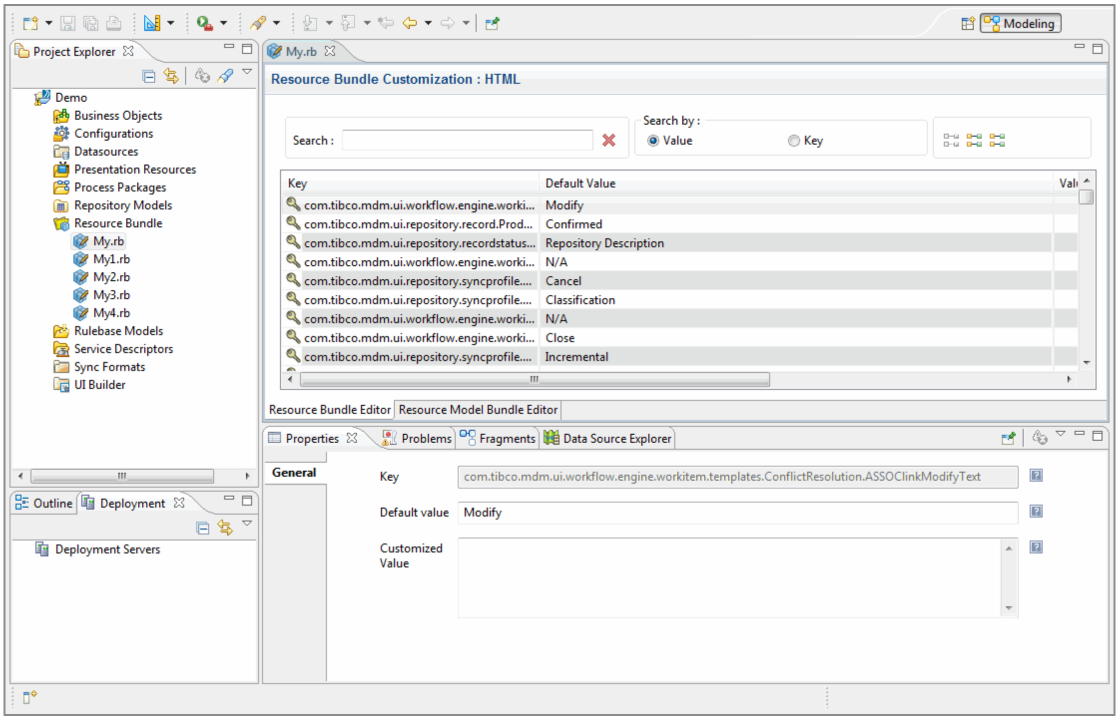
- Java Script is not supported for hot deployment.
- It is recommended to restart the server after deployment the resource bundle for the first time.
- All common resources (like login page, reset password) for all enterprises must be deployed at application level .

Resource Bundle Editor

The resource bundle editor is divided into two section, one with list of resource bundle in tabular format in key-value pair and another the property section for the properties of each resource bundle.

You can perform various functions like:

- Customize a resource bundle
- Search by value or key
- Filter



Resource Bundle Customization : HTML

Search :

Search by : ● Value ○ Key

Key	Default Value	Val
com.tibco.mdm.ui.workflow.engine.worki...	Modify	
com.tibco.mdm.ui.repository.record.Prod...	Confirmed	
com.tibco.mdm.ui.repository.recordstatus...	Repository Description	
com.tibco.mdm.ui.workflow.engine.worki...	N/A	
com.tibco.mdm.ui.repository.syncprofile....	Cancel	
com.tibco.mdm.ui.repository.syncprofile....	Classification	
com.tibco.mdm.ui.workflow.engine.worki...	N/A	
com.tibco.mdm.ui.workflow.engine.worki...	Close	
com.tibco.mdm.ui.repository.syncprofile....	Incremental	

Key: com.tibco.mdm.ui.workflow.engine.workitem.templates.ConflictResolution.ASSOClinkModifyText

Default value: Modify

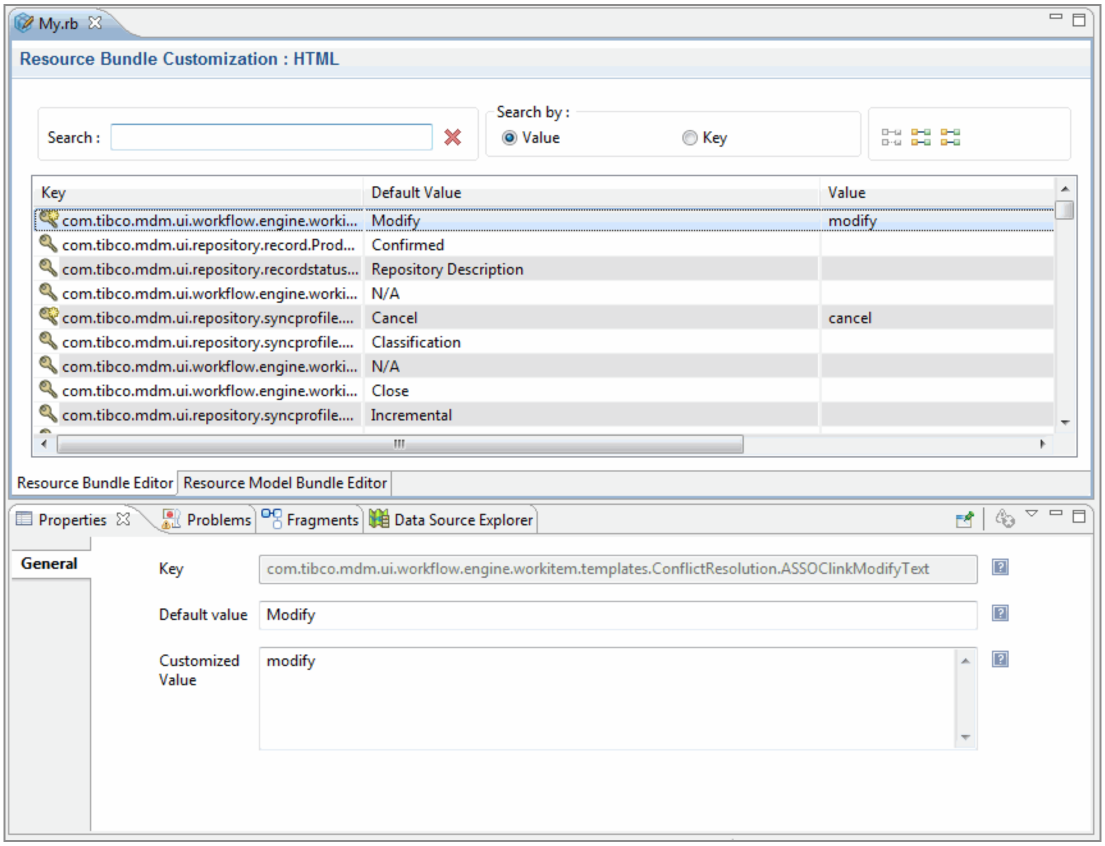
Customized Value:

Customize

To customize the resource bundle, select the property you want to customize. The details of the selected property is displayed in the Properties section.

Procedure

1. The **Key** field displays the selected properties key. It is a noneditable field.
2. The default value associated with the key is displayed in the **Default value** field. This is a noneditable field.
3. Enter the customized value for the selected property in the **Customized Value** field in the Property Section. Alternatively, you can also double click the **Value** field in the table and enter the custom value in the **Value** field.



Search by : ● Value ○ Key

Key	Default Value	Value
com.tibco.mdm.ui.workflow.engine.worki...	Modify	modify
com.tibco.mdm.ui.repository.record.Prod...	Confirmed	
com.tibco.mdm.ui.repository.recordstatus...	Repository Description	
com.tibco.mdm.ui.workflow.engine.worki...	N/A	
com.tibco.mdm.ui.repository.syncprofile....	Cancel	cancel
com.tibco.mdm.ui.repository.syncprofile....	Classification	
com.tibco.mdm.ui.workflow.engine.worki...	N/A	
com.tibco.mdm.ui.workflow.engine.worki...	Close	
com.tibco.mdm.ui.repository.syncprofile....	Incremental	

Key: com.tibco.mdm.ui.workflow.engine.workitem.templates.ConflictResolution.ASSOClinkModifyText

Default value: Modify

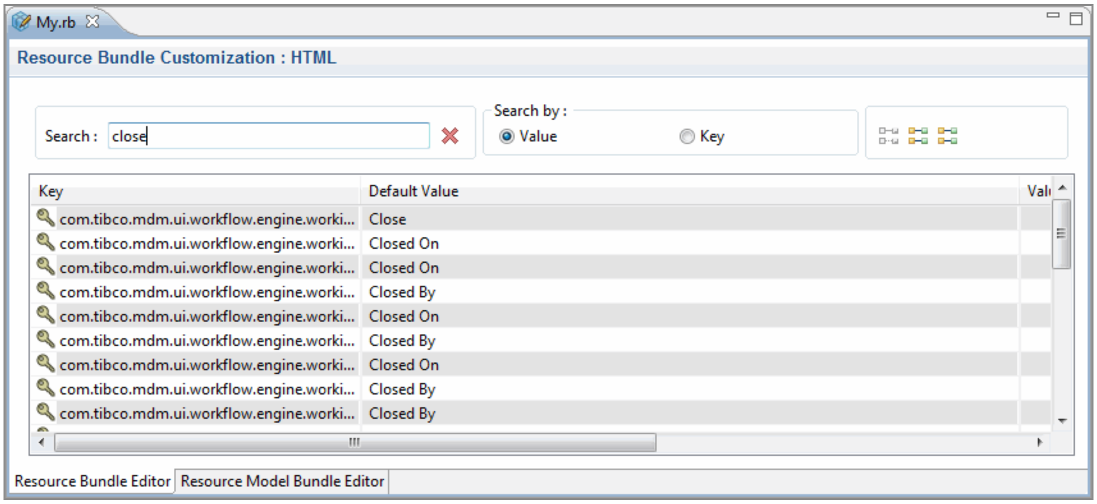
Customized Value: modify

Search

The resource bundle can be searched based on the key or value.

Procedure

1. Select the appropriate **Search by** option. The available options are **Value** and **Key**.
2. Specify the search criteria in the **Search** field. For example, you can search for a key containing the word "close" by specifying the text "close" in Search field with Search by criteria as Value.



My.rb

Resource Bundle Customization : HTML

Search : close

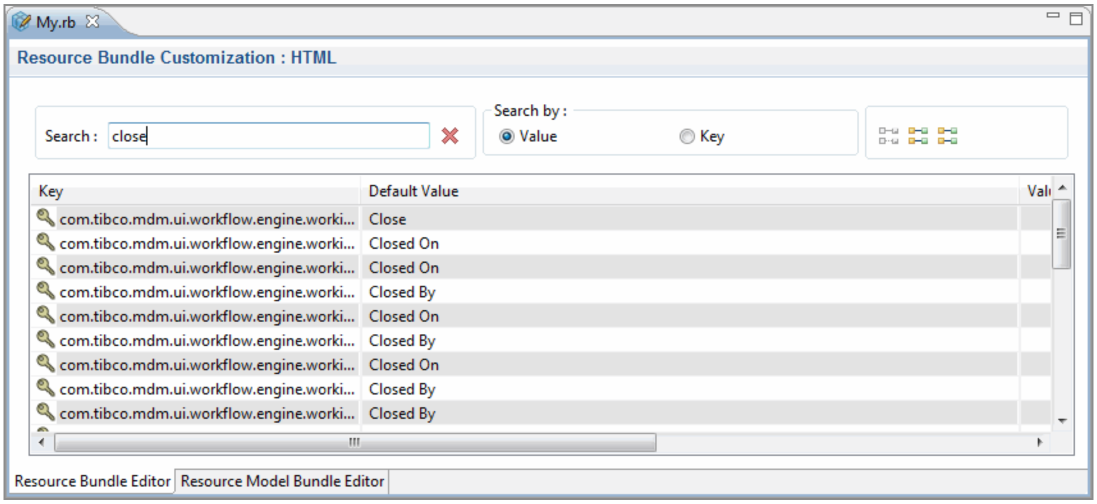
Search by : ● Value ○ Key

Key	Default Value	Valı
com.tibco.mdm.ui.workflow.engine.worki...	Close	
com.tibco.mdm.ui.workflow.engine.worki...	Closed On	
com.tibco.mdm.ui.workflow.engine.worki...	Closed On	
com.tibco.mdm.ui.workflow.engine.worki...	Closed By	
com.tibco.mdm.ui.workflow.engine.worki...	Closed On	
com.tibco.mdm.ui.workflow.engine.worki...	Closed By	
com.tibco.mdm.ui.workflow.engine.worki...	Closed On	
com.tibco.mdm.ui.workflow.engine.worki...	Closed By	
com.tibco.mdm.ui.workflow.engine.worki...	Closed By	

Resource Bundle Editor | Resource Model Bundle Editor

Filter

To filter, perform the following:



My.rb

Resource Bundle Customization : HTML

Search : close

Search by : ● Value ○ Key

Key	Default Value	Valı
com.tibco.mdm.ui.workflow.engine.worki...	Close	
com.tibco.mdm.ui.workflow.engine.worki...	Closed On	
com.tibco.mdm.ui.workflow.engine.worki...	Closed On	
com.tibco.mdm.ui.workflow.engine.worki...	Closed By	
com.tibco.mdm.ui.workflow.engine.worki...	Closed On	
com.tibco.mdm.ui.workflow.engine.worki...	Closed By	
com.tibco.mdm.ui.workflow.engine.worki...	Closed On	
com.tibco.mdm.ui.workflow.engine.worki...	Closed By	
com.tibco.mdm.ui.workflow.engine.worki...	Closed By	

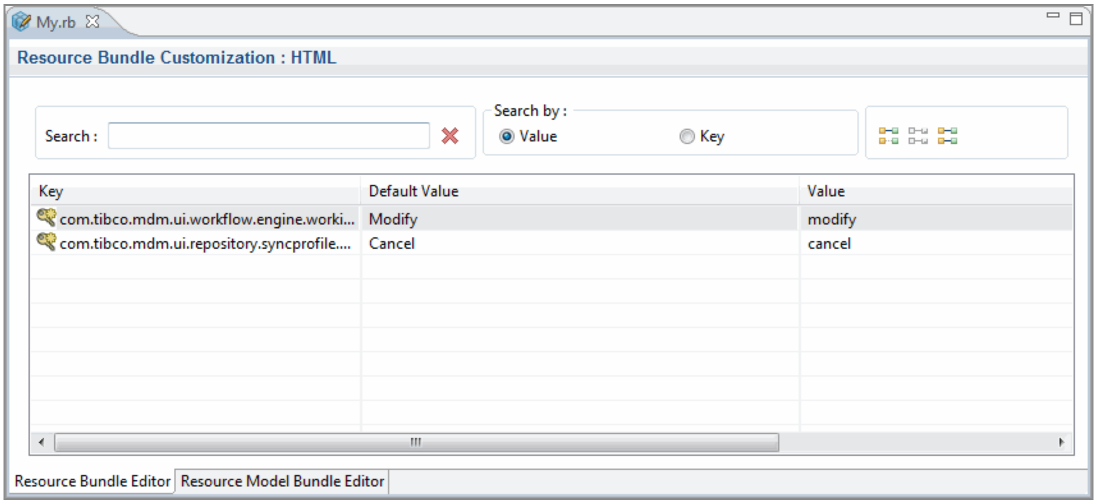
Resource Bundle Editor | Resource Model Bundle Editor

Procedure

1. To filter the resource based on the customized resource list or all resources, click .

2. By default "All Default Resource" is displayed. To display "Custom Resources" click the second filter icon. It displays only the custom resource. For example, the default value "Modify" is customized to "modify" in the value field. Similarly to display "All Custom and Model Resources" click the third filter icon.

Note: The model resource is the new key or the repository model customization which is done using the **Resource Model Bundle Editor** .



Resource Bundle Customization : HTML

Search by : ● Value ○ Key

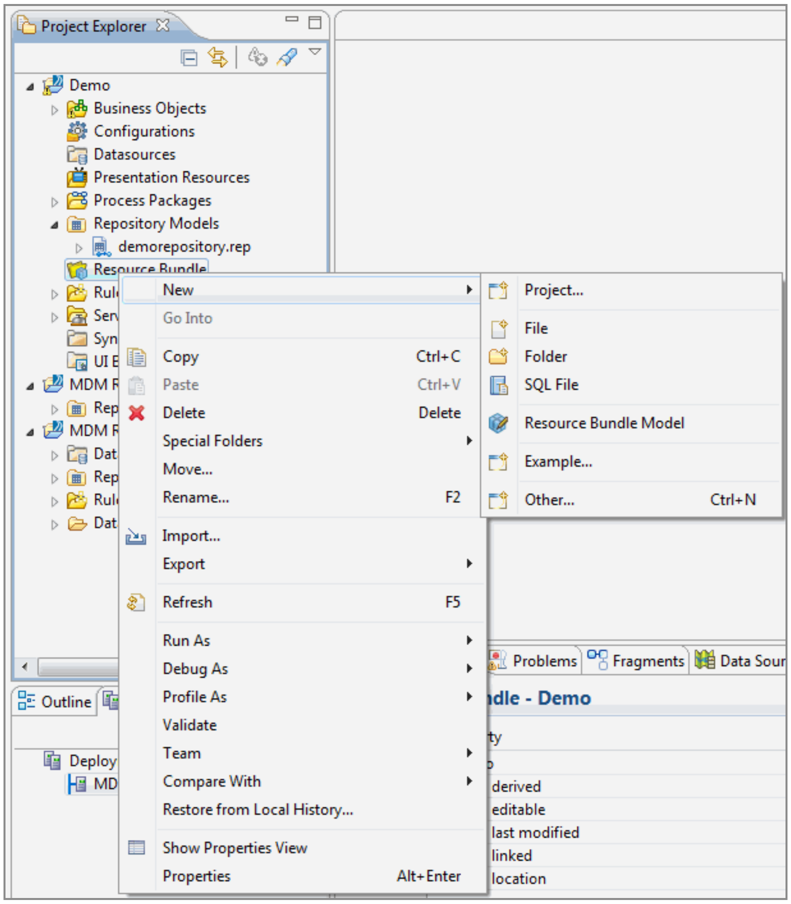
Key	Default Value	Value
com.tibco.mdm.ui.workflow.engine.worki...	Modify	modify
com.tibco.mdm.ui.repository.syncprofile....	Cancel	cancel

Creating a Resource Bundle

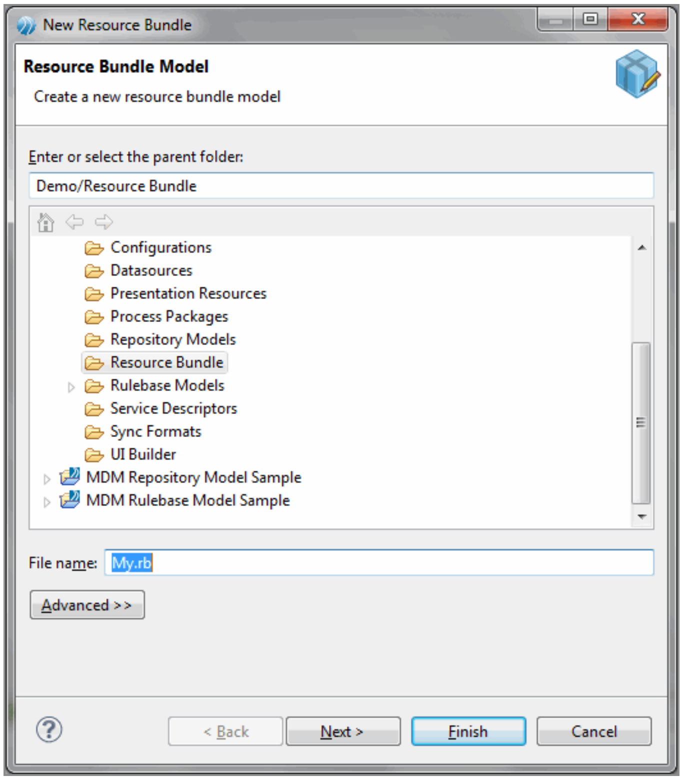
Using this option, you can create a resource bundle.

Procedure

1. In the Project Explorer under the Repository model file, Right click on Resource Bundle on which user want to create a resource bundle and select **New > Resource Bundle Model.**



2. The Resource Bundle Model Wizard is displayed.



New Resource Bundle

Resource Bundle Model

Create a new resource bundle model

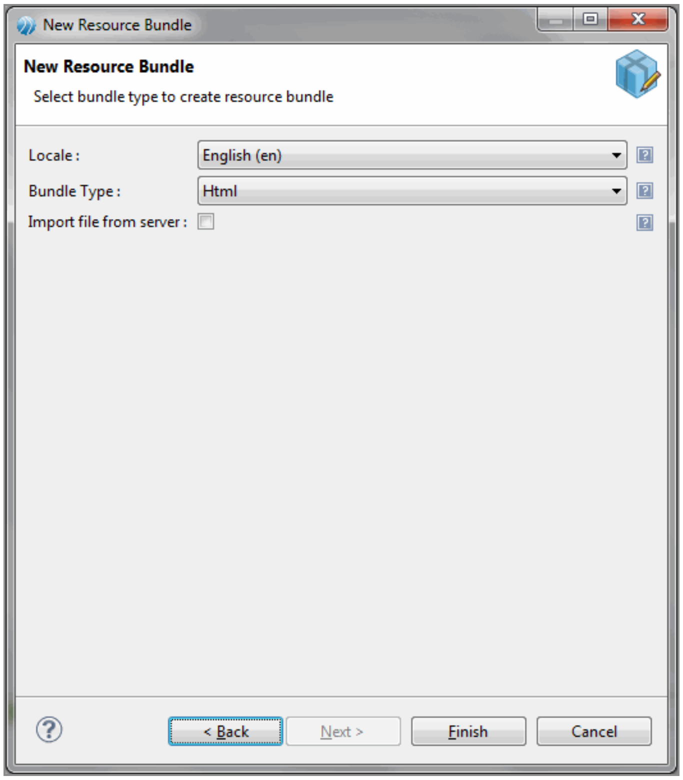
Enter or select the parent folder:

Demo/Resource Bundle

- Configurations
- Datasources
- Presentation Resources
- Process Packages
- Repository Models
- Resource Bundle
- Rulebase Models
- Service Descriptors
- Sync Formats
- UI Builder
- MDM Repository Model Sample
- MDM Rulebase Model Sample

File name: My.rb

3. Enter the Resource bundle name in the **File name** field and click **Next**.
4. The New Resource Bundle wizard displays the bundle type.



New Resource Bundle

New Resource Bundle

Select bundle type to create resource bundle

Locale : English (en)

Bundle Type : Html

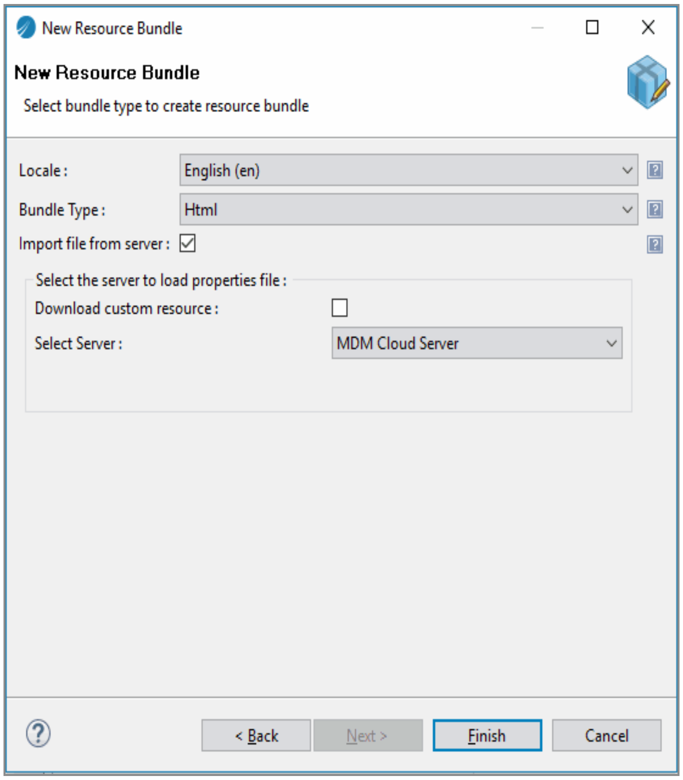
Import file from server : □

5. Select the **Locale**, **Bundle Type** from the drop-down list. Select **Import file from server** checkbox if you want to import the resource file from the TIBCO MDM server. For the first time you need to select the import file from server checkbox.

6. If you select the **Import file from server** checkbox, the **Select the server to load properties file** section is displayed.

   - Select the **Download custom resource** checkbox if you have already deployed the custom resource and you want to import the custom resource for customization. For more information on download custom resource refer to Download Custom Resource.

   - All the available connected TIBCO MDM servers are displayed in the **Select Server** drop-down list. Select the appropriate server. For more information on creating a new server refer to Creating a MDM Deployment Server.



New Resource Bundle

New Resource Bundle

Select bundle type to create resource bundle

Locale : English (en)

Bundle Type : Html

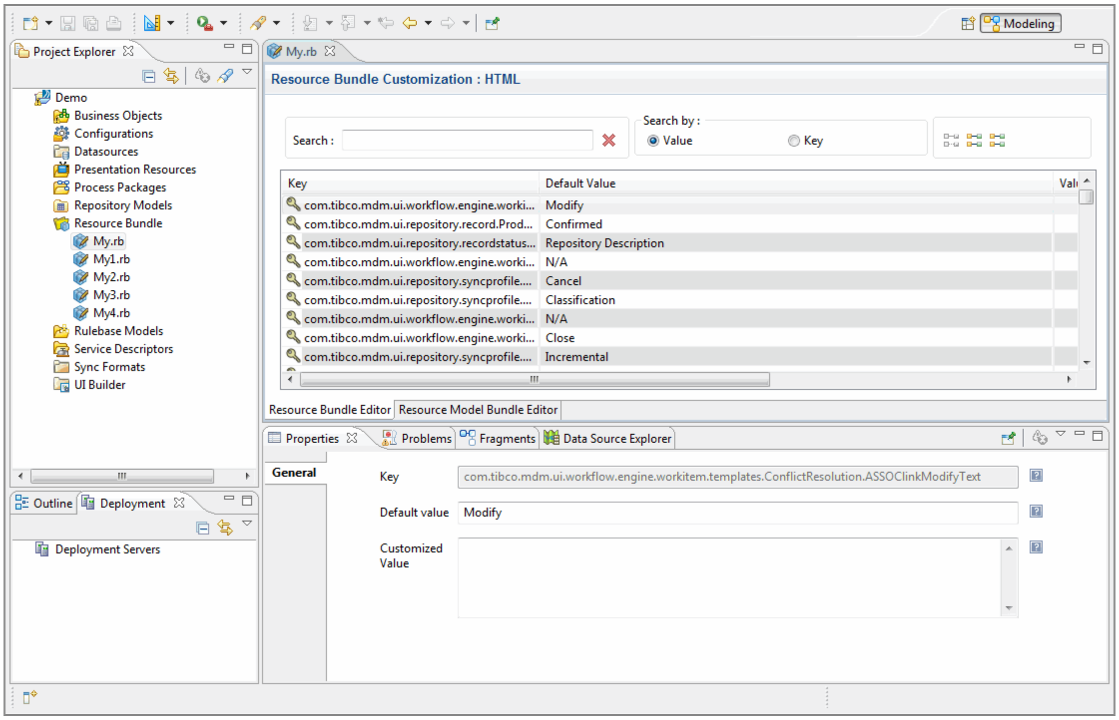
Import file from server : ☑

Select the server to load properties file :

Download custom resource : □

Select Server : MDM Cloud Server

7. Click **Finish**. The Resource Bundle editor with the newly created resource bundle is displayed. After clicking finish button custom resources will be downloaded for the selected locale, bundle type from the enterprise selected.



Search by: ● Value ○ Key

Key	Default Value	Valı
com.tibco.mdm.ui.workflow.engine.worki...	Modify	
com.tibco.mdm.ui.repository.record.Prod...	Confirmed	
com.tibco.mdm.ui.repository.recordstatus...	Repository Description	
com.tibco.mdm.ui.workflow.engine.worki...	N/A	
com.tibco.mdm.ui.repository.syncprofile....	Cancel	
com.tibco.mdm.ui.repository.syncprofile....	Classification	
com.tibco.mdm.ui.workflow.engine.worki...	N/A	
com.tibco.mdm.ui.workflow.engine.worki...	Close	
com.tibco.mdm.ui.repository.syncprofile....	Incremental	

Key: com.tibco.mdm.ui.workflow.engine.workitem.templates.ConflictResolution.ASSOClinkModifyText

Default value: Modify

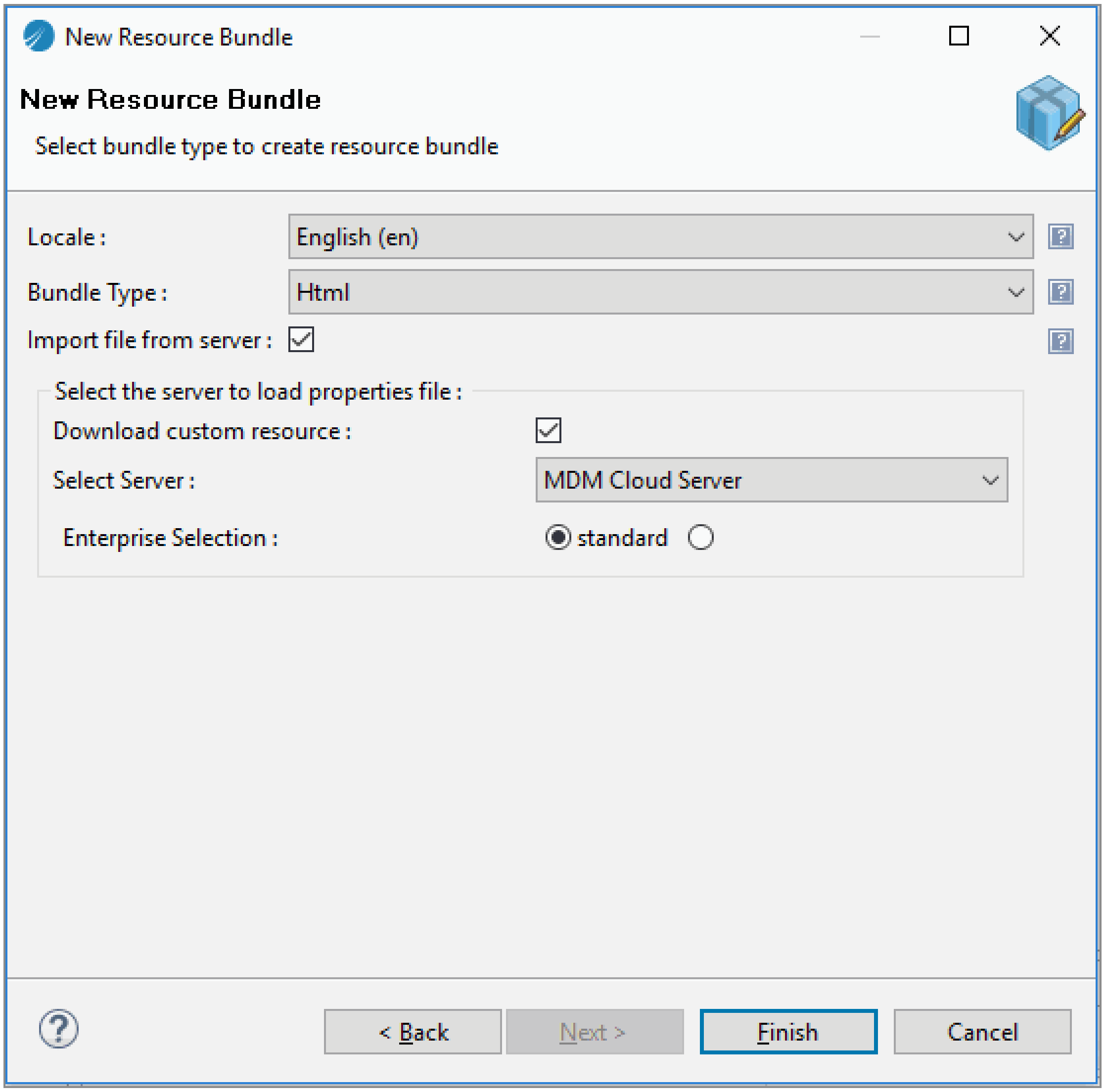
Customized Value:

Download Custom Resource

The download custom resource is used to customized the custom resources which is already deployed.

Procedure

1. Select the Download Custom resource checkbox.



New Resource Bundle

New Resource Bundle

Select bundle type to create resource bundle

Locale : English (en)

Bundle Type : Html

Import file from server : ☑

Select the server to load properties file :

Download custom resource : ☑

Select Server : MDM Cloud Server

Enterprise Selection : ● standard ○

2. On selecting the download custom resources checkbox then **Enterprise Selection** option is displayed. The available option are **standard** and <your enterprise>. Select the standard option if you want to download from application level or select <your enterprise> for example, techpubs. By default the standard enterprise is selected.
3. Click **Finish**. The resource bundle editor with the custom resource is imported for customization.

Resource Model Bundle Editor

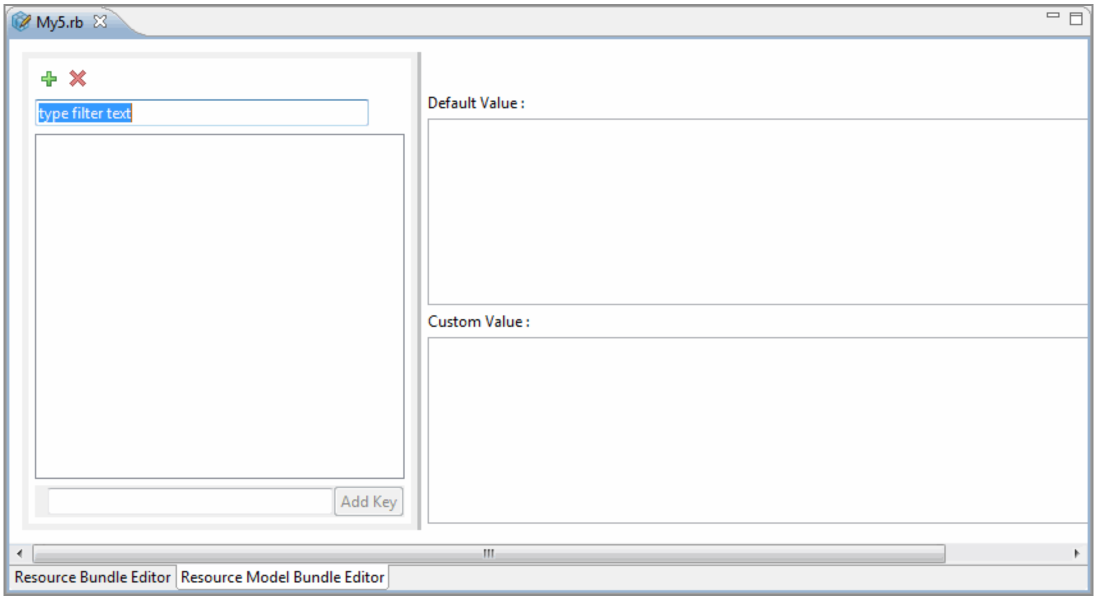
The Resource bundle editor is used to add new keys for all resource types.

Repository Model File for Generating Resource Keys

In case of Shared String resource bundle, the Resource Model bundle editor is used to customize the repository model.

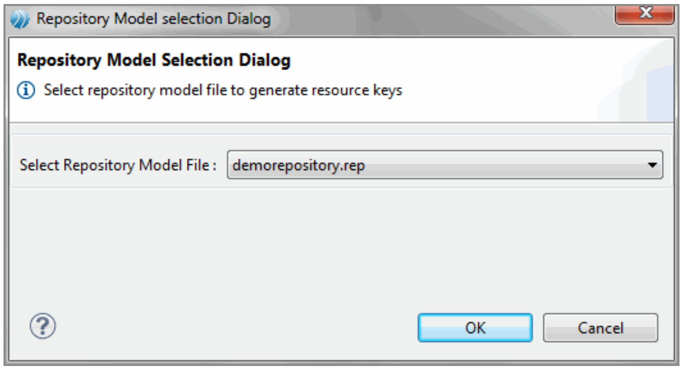
Procedure

1. Click on the **Resource Model Bundle Editor** tab, the editor with the various sections is displayed.



2. Click ✚ .

3. The Repository model Selection Dialog is displayed.



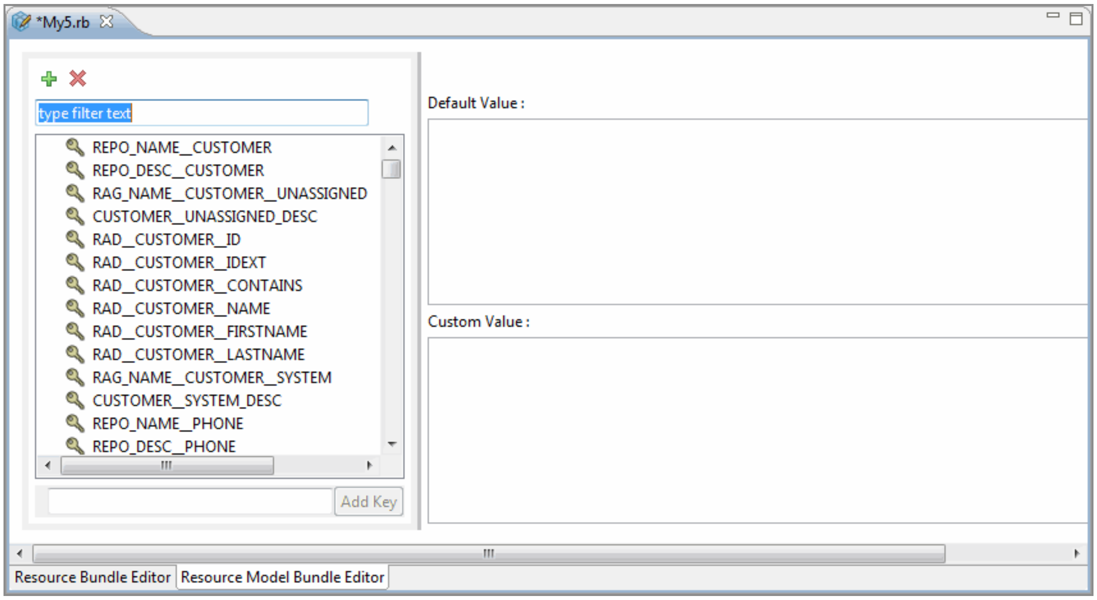
Repository Model Selection Dialog

Select repository model file to generate resource keys

Select Repository Model File : demorepository.rep

4. From the **Select Repository Model File** drop-down list, select the repository model file for which you want to generate the keys.

5. Click **OK**.
6. The keys are generated for the repository, attribute groups and attributes names and descriptions.



Customizing an Existing Properties from Model file

Each of the existing properties can be customized.

Procedure

1. Select the property you want to customize from the left tree panel.
2. To search for a property, type the property name in the **type filter text** field.
3. The default value of the properties is displayed in the **Default Value** field.
4. Enter the customized value in the **Custom Value** field. For example, for RAD__CUSTOMER__FIRSTNAME key the custom value is "firstname". For more information on the standards to provide customization keys, refer to *TIBCO MDM Customization Guide*.
5. Save the customized value. The newly created customized value overrides the default value.

Adding a New Resource Property to the Model file

Any number of new property keys can be added.

Procedure

1. Type the complete resource key string in the text box below the left tree panel. The add key button is enabled.
2. Click Add Key.

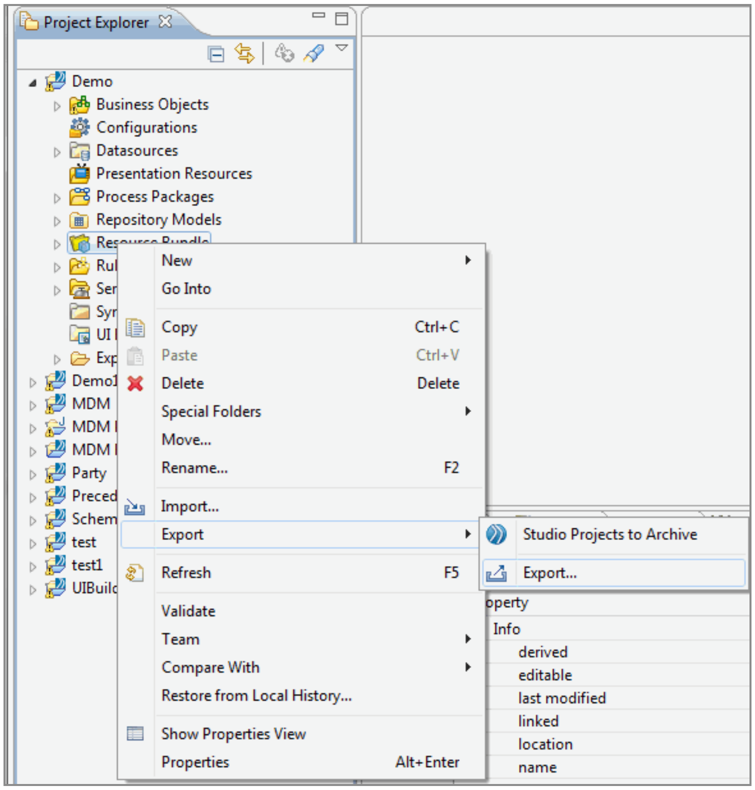
   The new property key is added to the resource list.
3. Enter the custom value in the Custom Value field. The default value for the newly added resource key would be empty.
4. The resource key string cannot be empty or does not allow white spaces. You can enter only alphanumeric and underscores. You cannot enter duplicate keys. On entering a key which is already present in the resource list ,the Add Key button is disable and an error message "Key with this name is already present. Duplicate keys are not allowed " is displayed.

Exporting a Resource Bundle

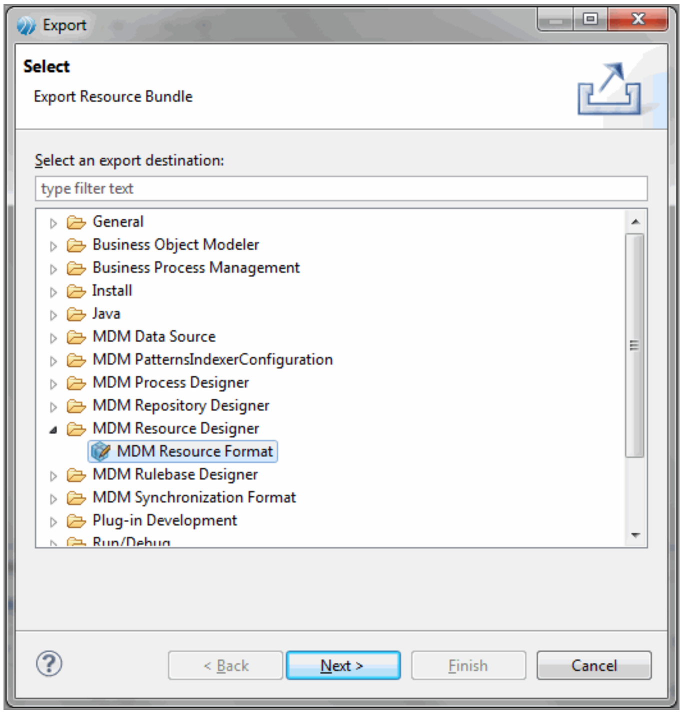
Using Export Wizard you can export the Resource Bundle.

Procedure

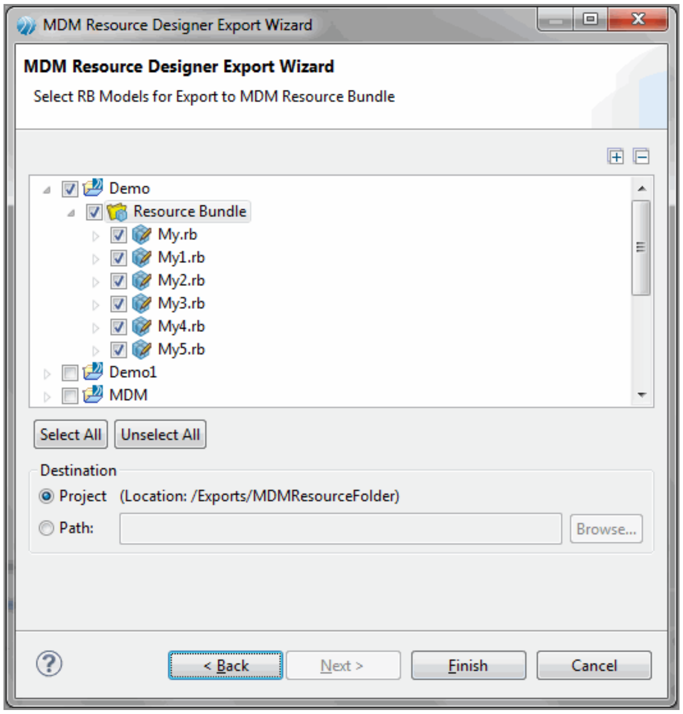
1. Right-click on the **Resource Bundle** folder in the Project Explorer and click **Export--> Export**.



2. The Export Wizard is displayed. Expand **MDM Resource Designer** from export destination list, select **MDM Resource Format** and click **Next.**



3. Select the resource bundle which you want to export. By default resource bundle is exported to Exports/MDMExportFolder directory which is in same project. You can change the location by specifying different destination path. Click **Path** and browse to the folder in which you want to export the resource bundle file and click **Finish.**



MDM Resource Designer Export Wizard

MDM Resource Designer Export Wizard

Select RB Models for Export to MDM Resource Bundle

- ☑ Demo
  - ☑ Resource Bundle
    - ☑ My.rb
    - ☑ My1.rb
    - ☑ My2.rb
    - ☑ My3.rb
    - ☑ My4.rb
    - ☑ My5.rb
- □ Demo1
- □ MDM

Destination

● Project (Location: /Exports/MDMResourceFolder)

○ Path:

Note: For more information on resource bundle, refer to TIBCO MDM Customization Guide.

Deploying the Custom Resource Bundle

The customized resource bundle is deployed at the enterprise level or for the entire application.

Procedure

1. Select the custom resource bundle to deploy from selected module.
2. Select the enterprise to which you want to deploy from **Enterprise Selection** section. On selecting the ’Standard' enterprise, the custom bundle is deployed at the application level.
3. Click **Finish**.

Note: Hot deployment is supported for all type of resource bundle except Java Script.

Repository Model Report

The TIBCO MDM Studio model is used to develop a physical model for the TIBCO MDM Server.

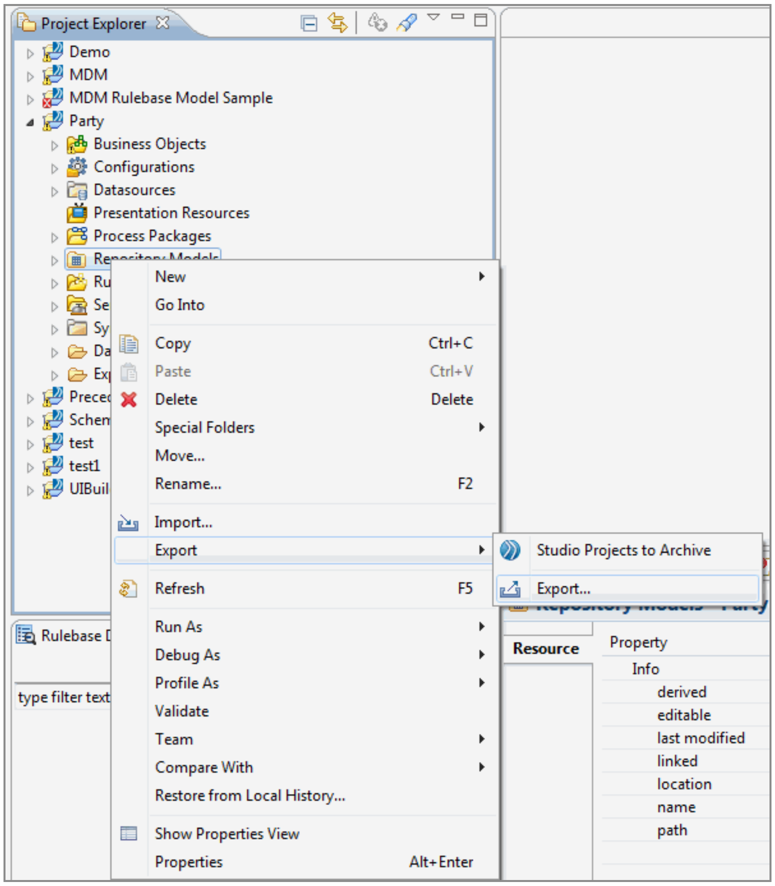
The TIBCO MDM server defines sophisticated models, which is difficult to interpret without the TIBCO MDM Studio. The new Repository Report feature can be used by all stakeholders in the various design phase of the models. The Repository Report will make the model easy to understand and thereby help different stakeholders contribute in the design process.

Creating a Repository Model Report

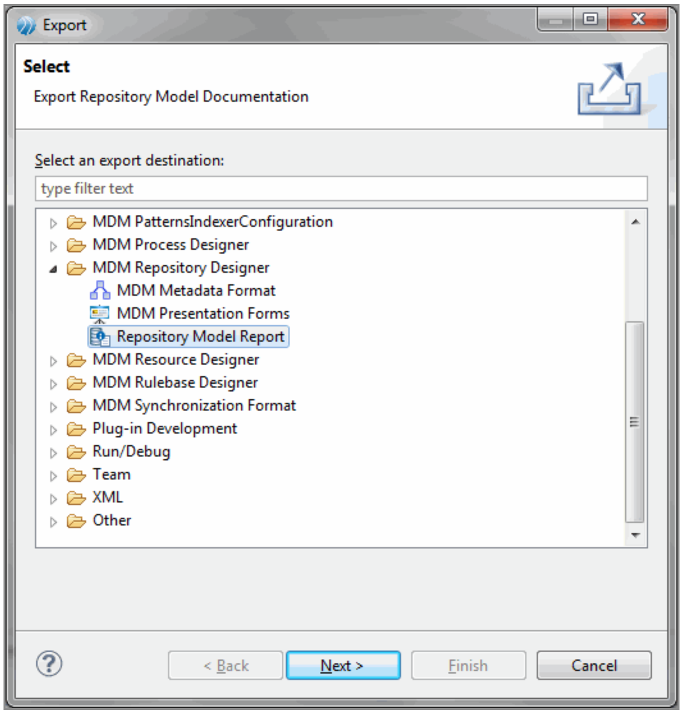
Using this option, you can create a repository model report.

Procedure

1. In the Project Explorer under the Repository model file, Right click on Repository Model and select **Export-- > Export.**



2. The Export Repository Model Documentation wizard is displayed.



Export

Select

Export Repository Model Documentation

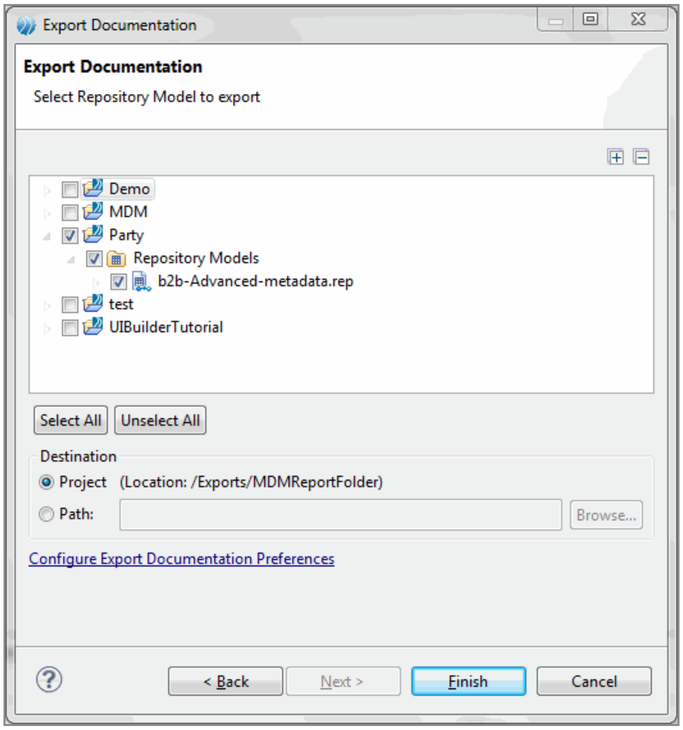
Select an export destination:

type filter text

3. Expand the **MDM Repository Designer** folder and select **Repository Model Report**.

4. Click **Next**.

5. The Repository selection page is displayed on Export Documentation wizard.



Export Documentation

Export Documentation

Select Repository Model to export

- □ Demo
- □ MDM
- ☑ Party
  - ☑ Repository Models
    - ☑ b2b-Advanced-metadata.rep
- □ test
- □ UIBuilderTutorial

Destination

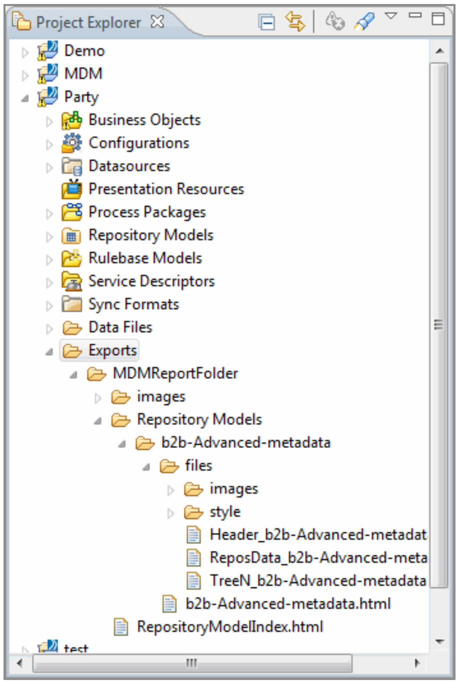
● Project (Location: /Exports/MDMReportFolder)

○ Path:

6. Select the checkbox corresponding to the repository for which you want to generated the model report. For example, b2b-Advanced-metadata.rep is selected from the Party Model.

7. Click **Finish**. By default, the report will be save in the **Exports/MDMReportFolder** under current Project directory. If you want to change the default location, select the **Path** option and browse to the absolute path where you want to save the report artifacts.

8. To view the Repository Model artifacts, expand the **Exports** folder. All the supporting files are stored in the **MDMReportFolder**.



9. The supporting images for the Index page (For example, `b2b-Advanced-metadata.html`)is stored in the **images** folder and all supporting html files are stored in the **Repository Models** folder.

10. For each repository model separate folder is created in a **Repository Models** folder. The **files** folder contains all supporting .html files. In files folder, all Model and Repository images are stored in the images folder. In **style** folder, all .CSS files are stored.

Viewing Repository Model Report

The Repository Model Report displays all the repository for which the report is generated.

You can view the Repositories, Repository Model information, Repository details, Attribute Groups details, Input Map details, Output Map details, Classification Scheme details, Relationship and Relationship attribute group details, and Perspective details.

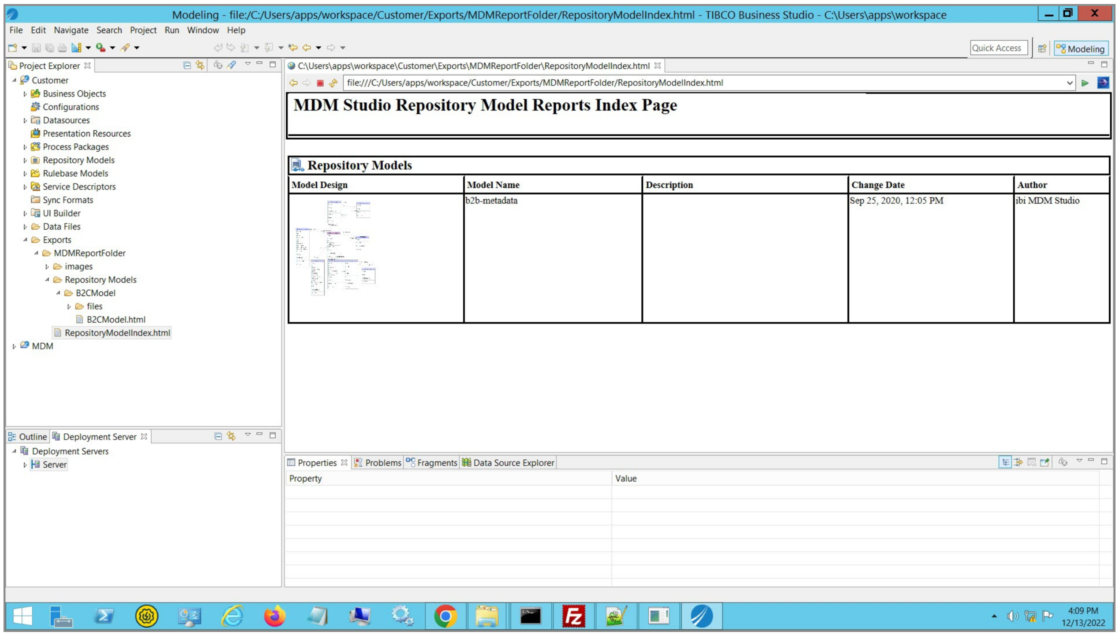
Note: For better viewing, you can view the Repository Model Report in an external browser.

Index Page

The Repository Model Index page displays the model design for each of the repository. You can access the details of the report by clicking the model designs.

Procedure

1. Select the <RepositoryModelIndex.html file and open it in a web browser. The selected <RepositoryModelIndex.html index page opens on the right hand side.
2. The index page comprised of the model design, model name, description, change date, and the author of the report.



MDM Studio Repository Model Reports Index Page

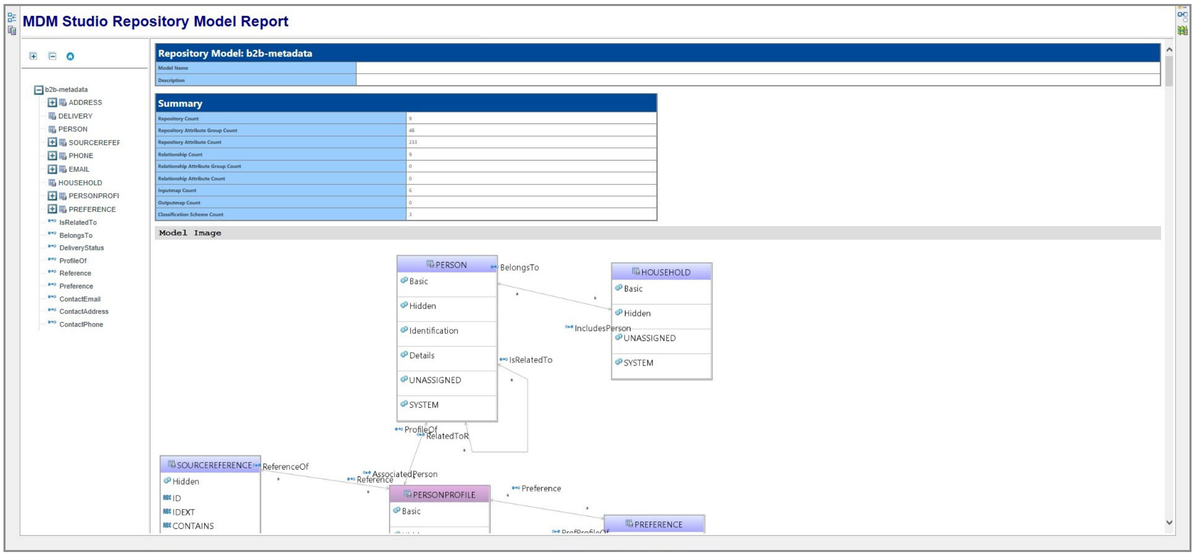
Repository Models

Model Design	Model Name	Description	Change Date	Author
	b2b-metadata		Sep 25, 2020, 12:05 PM	ibi MDM Studio

Repository Model Report

The Repository Model report shows the summary of the various parameters in the repository.

The summary mainly comprises of counts of repositories, repository attribute groups, repository attributes, relationships, relationship attribute groups, relationship attributes, inputmaps, outputmaps, and classification schemes.



MDM Studio Repository Model Report

Repository Model: b2b-metadata	
Model Name	
Description	

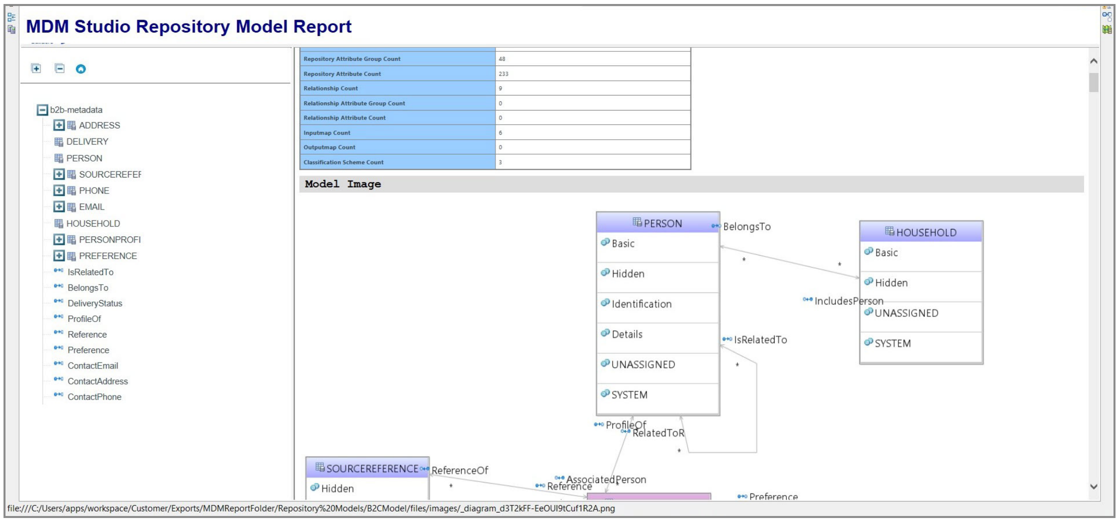
Summary	
Repository Count	9
Repository Attribute Group Count	48
Repository Attribute Count	233
Relationship Count	9
Relationship Attribute Group Count	0
Relationship Attribute Count	0
Inputmap Count	6
Outputmap Count	0
Classification Scheme Count	3

Model Image

Procedure

1. You can view the complete list of repositories, relationships in a tree view by clicking ⊞.

2. The complete list of repositories, relationship and other components of the repository are displayed.



MDM Studio Repository Model Report

Repository Attribute Group Count	48
Repository Attribute Count	233
Relationship Count	9
Relationship Attribute Group Count	0
Relationship Attribute Count	0
Inputmap Count	6
Outputmap Count	0
Classification Scheme Count	3

Model Image

file:///C:/Users/apps/workspace/Customer/Exports/MDMReportFolder/Repository%20Models/B2CModel/files/images/_diagram_d3T2kFF-EeOUl9tCuf1R2A.png

3. You can also expand the tree view by clicking ⊞.

4. To collapse the tree view click ⊟.

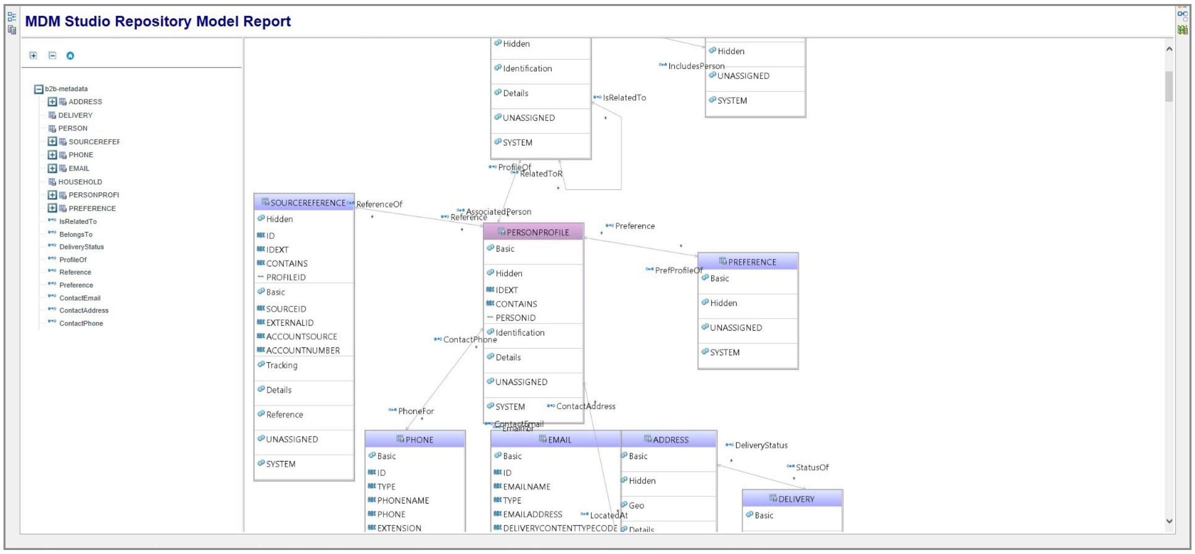
5. To go back to index page click ⌂.

6. Click the repository, relationship and so on to view their details.

Repository Model Information

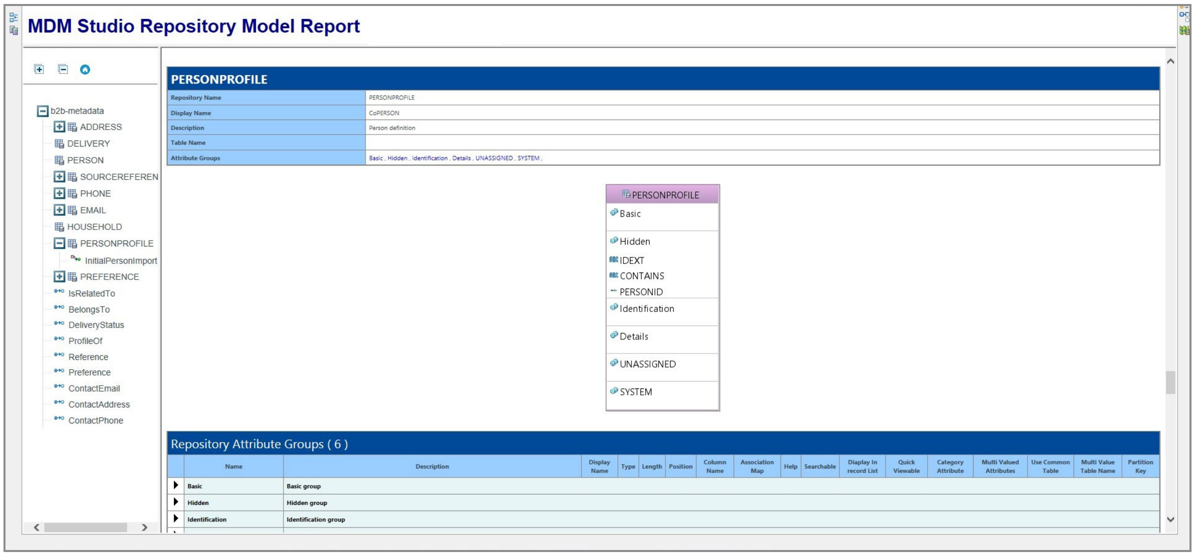
The Repository Model image displays the complete repository information in the report.

All the repositories, relationships in the repository model are visible in the report. To view the complete report, click on the Repository Model Image and the new window opens, maximize the new window to view more details.



Repository Details

The Repository Details report displays the Repository name, its Display Name, Description, Table Name and all Attribute Groups name from the repository. On clicking the link for each attribute group, their respective attributes along with their parameters are displayed.



PERSONPROFILE

Repository Name	PERSONPROFILE
Display Name	CoPERSON
Description	Person definition
Table Name	
Attribute Groups	Basic , Hidden , Identification , Details , UNASSIGNED , SYSTEM ,

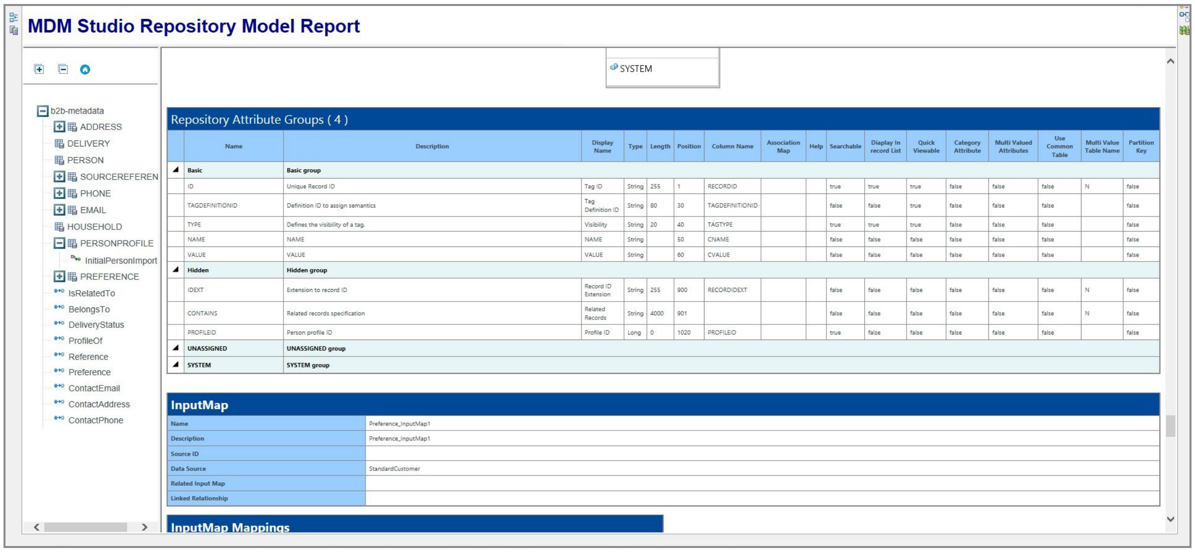
Repository Attribute Groups (6)

	Name	Description	Display Name	Type	Length	Position	Column Name	Association Map	Help	Searchable	Display In record List	Quick Viewable	Category Attribute	Multi Valued Attributes	Use Common Table	Multi Value Table Name	Partition Key
▶	Basic	Basic group															
▶	Hidden	Hidden group															
▶	Identification	Identification group															

Note: All blue names indicate links to that particular section.

Attributes Group Details

The Attribute Group details report displays all the attribute groups available in the repository. Expand each of the attribute group to view their attribute details.



Repository Attribute Groups (4)

	Name	Description	Display Name	Type	Length	Position	Column Name	Association Map	Help	Searchable	Display In record List	Quick Viewable	Category Attribute	Multi Valued Attributes	Use Common Table	Multi Value Table Name	Partition Key
◢	Basic	Basic group															
	ID	Unique Record ID	Tag ID	String	255	1	RECORDID			true	true	true	false	false	false	N	false
	TAGDEFINITIONID	Definition ID to assign semantics	Tag Definition ID	String	80	30	TAGDEFINITIONID			false	false	true	false	false	false		false
	TYPE	Defines the visibility of a tag.	Visibility	String	20	40	TAGTYPE			true	true	true	false	false	false		false
	NAME	NAME	NAME	String		50	CNAME			false	false	false	false	false	false		false
	VALUE	VALUE	VALUE	String		60	CVALUE			false	false	false	false	false	false		false
◢	Hidden	Hidden group															
	IDEXT	Extension to record ID	Record ID Extension	String	255	900	RECORDIDEXT			false	false	false	false	false	false	N	false
	CONTAINS	Related records specification	Related Records	String	4000	901				false	false	false	false	false	false	N	false
	PROFILEID	Person profile ID	Profile ID	Long	0	1020	PROFILEID			true	false	false	false	false	false		false
◢	UNASSIGNED	UNASSIGNED group															
◢	SYSTEM	SYSTEM group															

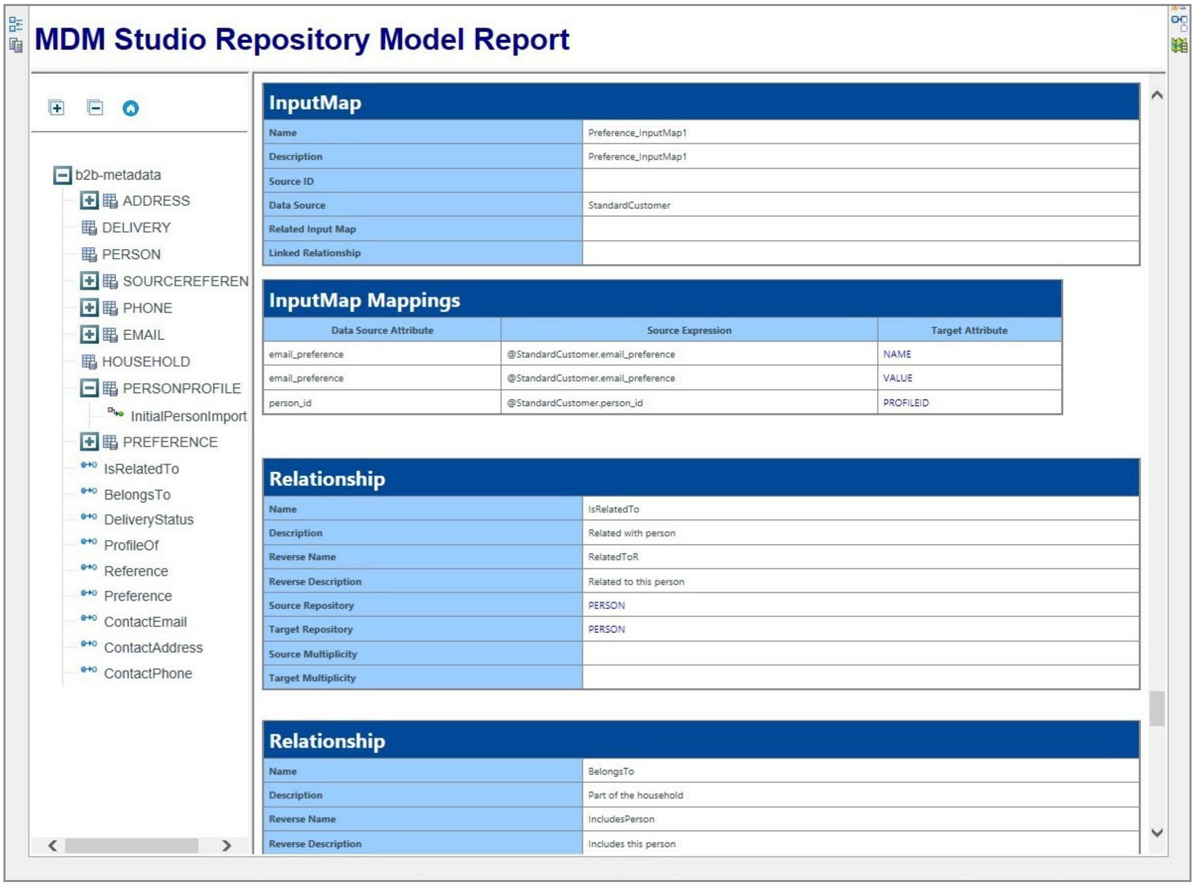
InputMap

Name	Preference_InputMap1
Description	Preference_InputMap1
Source ID	
Data Source	StandardCustomer
Related Input Map	
Linked Relationship	

InputMap Mappings

Input Map Details with Input Mappings

Input Map and Input Mapping report displays Input Map name, Description, Source ID, the Data Source Name associated with it, the Related Input Map, and Linked Relationships. Input Map mapping is displayed along with Source expression.



InputMap

Name	Preference_InputMap1
Description	Preference_InputMap1
Source ID	
Data Source	StandardCustomer
Related Input Map	
Linked Relationship	

InputMap Mappings

Data Source Attribute	Source Expression	Target Attribute
email_preference	@StandardCustomer.email_preference	NAME
email_preference	@StandardCustomer.email_preference	VALUE
person_id	@StandardCustomer.person_id	PROFILEID

Relationship

Name	IsRelatedTo
Description	Related with person
Reverse Name	RelatedToR
Reverse Description	Related to this person
Source Repository	PERSON
Target Repository	PERSON
Source Multiplicity	
Target Multiplicity	

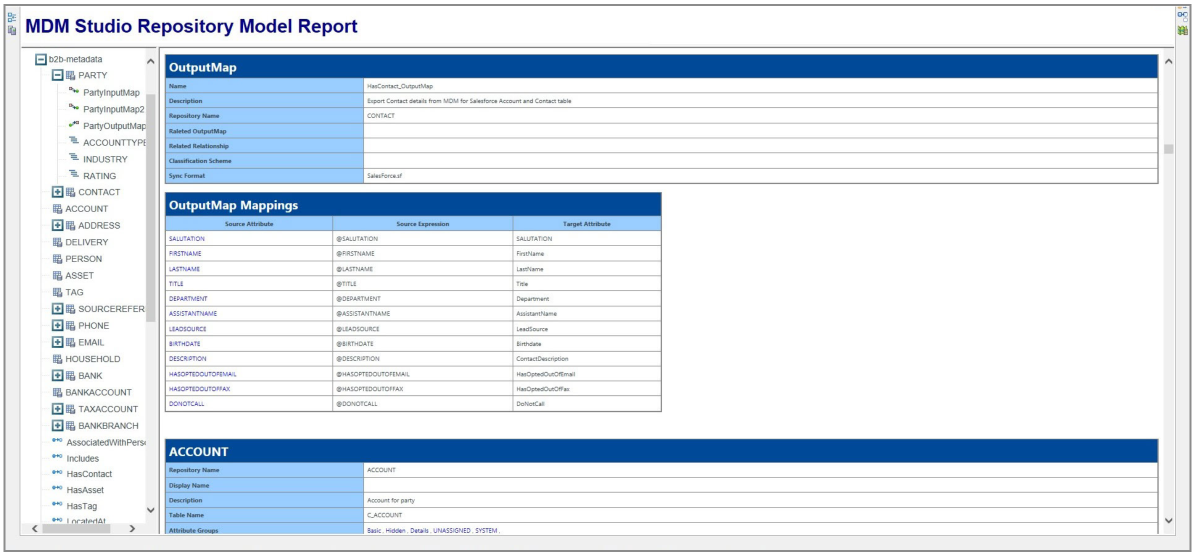
Relationship

Name	BelongsTo
Description	Part of the household
Reverse Name	IncludesPerson
Reverse Description	Includes this person

Note: All blue names indicate links to that particular section.

Output Map Details with Output Mappings

The Output Map and Output Mapping report displays Output Map Name, Description, Repository Name for which this output map is created, Related Output Map and Relationship, Classification Scheme and Sync Format associated with the output map. Output Map mapping along with source expression is displayed.



MDM Studio Repository Model Report

OutputMap

Name	HasContact_OutputMap
Description	Export Contact details from MDM for Salesforce Account and Contact table
Repository Name	CONTACT
Raleted OutputMap	
Related Relationship	
Classification Scheme	
Sync Format	SalesForce.sf

OutputMap Mappings

Source Attribute	Source Expression	Target Attribute
SALUTATION	@SALUTATION	SALUTATION
FIRSTNAME	@FIRSTNAME	FirstName
LASTNAME	@LASTNAME	LastName
TITLE	@TITLE	Title
DEPARTMENT	@DEPARTMENT	Department
ASSISTANTNAME	@ASSISTANTNAME	AssistantName
LEADSOURCE	@LEADSOURCE	LeadSource
BIRTHDATE	@BIRTHDATE	Birthdate
DESCRIPTION	@DESCRIPTION	ContactDescription
HASOPTEDOUTOFEMAIL	@HASOPTEDOUTOFEMAIL	HasOptedOutOfEmail
HASOPTEDOUTOFFAX	@HASOPTEDOUTOFFAX	HasOptedOutOfFax
DONOTCALL	@DONOTCALL	DoNotCall

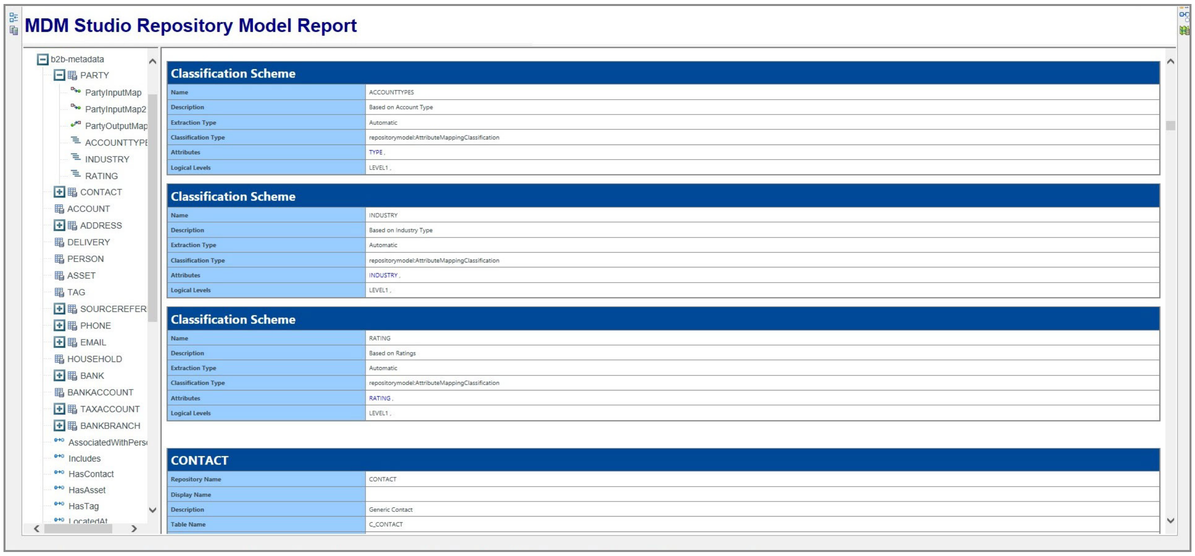
ACCOUNT

Repository Name	ACCOUNT
Display Name	
Description	Account for party
Table Name	C_ACCOUNT
Attribute Groups	Basic , Hidden , Details , UNASSIGNED , SYSTEM ,

Note: All blue names indicate links to that particular section.

Classification Schemes Details

The Classification Scheme details report displays Classification Scheme name, Description, Extraction Type, Classification Type, Attribute which participated in category creation if it is an Automatic or Mixed extraction type classification scheme, and Logical Levels in this Classification Scheme.



MDM Studio Repository Model Report

Classification Scheme

Name	ACCOUNTTYPES
Description	Based on Account Type
Extraction Type	Automatic
Classification Type	repositorymodel:AttributeMappingClassification
Attributes	TYPE ,
Logical Levels	LEVEL1 ,

Classification Scheme

Name	INDUSTRY
Description	Based on Industry Type
Extraction Type	Automatic
Classification Type	repositorymodel:AttributeMappingClassification
Attributes	INDUSTRY ,
Logical Levels	LEVEL1 ,

Classification Scheme

Name	RATING
Description	Based on Ratings
Extraction Type	Automatic
Classification Type	repositorymodel:AttributeMappingClassification
Attributes	RATING ,
Logical Levels	LEVEL1 ,

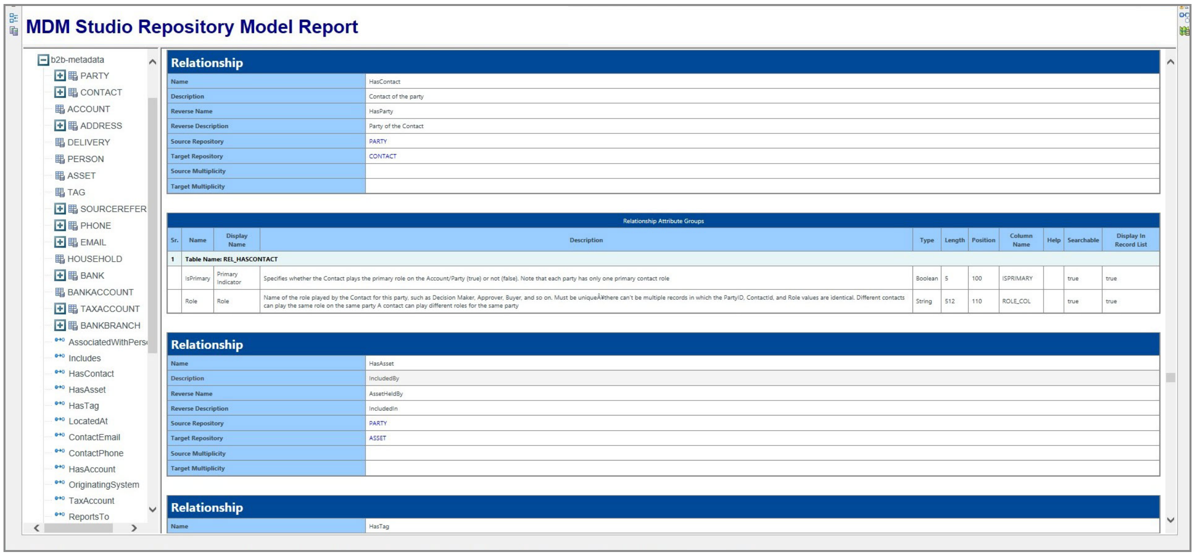
CONTACT

Repository Name	CONTACT
Display Name	
Description	Generic Contact
Table Name	C_CONTACT

Note: All blue names indicate links to that particular section.

Relationships and Relationship Attribute Group

The Relationship and the Relationship Attribute Group details report displays Relationship name, Description, Reverse Relationship Name, Reverse Relationship Description, Source and Target repository name on which this relationship is created, the source and target multiplicity. The relationship attribute group details are also displayed.



Relationship

Name	HasContact
Description	Contact of the party
Reverse Name	HasParty
Reverse Description	Party of the Contact
Source Repository	PARTY
Target Repository	CONTACT
Source Multiplicity	
Target Multiplicity	

Relationship Attribute Groups

Sr.	Name	Display Name	Description	Type	Length	Position	Column Name	Help	Searchable	Display In Record List
1	Table Name: REL_HASCONTACT									
	IsPrimary	Primary Indicator	Specifies whether the Contact plays the primary role on the Account/Party (true) or not (false). Note that each party has only one primary contact role	Boolean	5	100	ISPRIMARY		true	true
	Role	Role	Name of the role played by the Contact for this party, such as Decision Maker, Approver, Buyer, and so on. Must be uniqueÂ¥there can't be multiple records in which the PartyID, ContactId, and Role values are identical. Different contacts can play the same role on the same party A contact can play different roles for the same party	String	512	110	ROLE_COL		true	true

Relationship

Name	HasAsset
Description	IncludedBy
Reverse Name	AssetHeldBy
Reverse Description	IncludedIn
Source Repository	PARTY
Target Repository	ASSET
Source Multiplicity	
Target Multiplicity	

Relationship

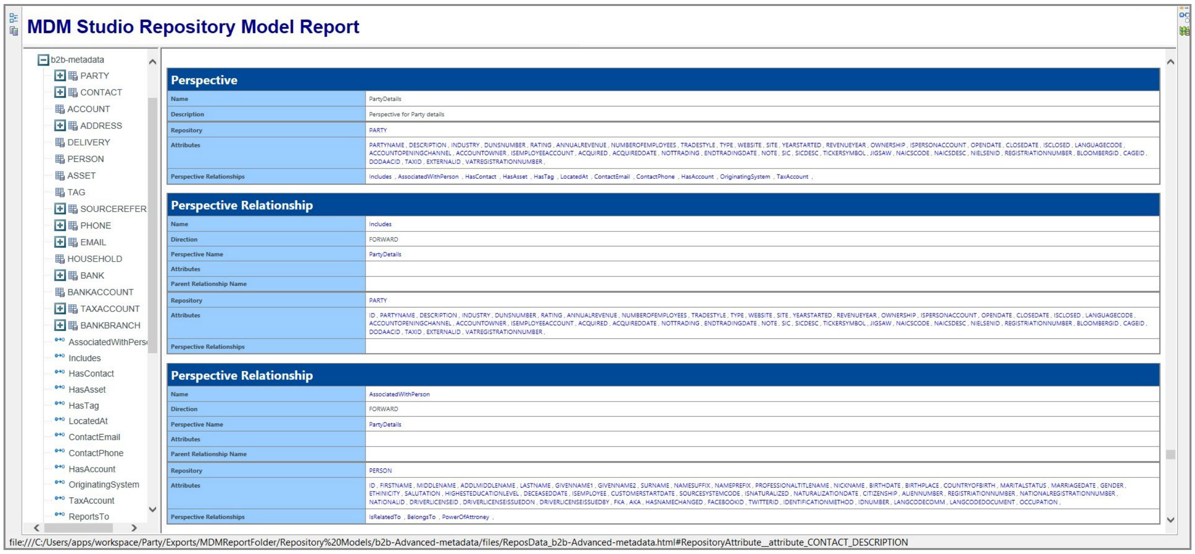
Name	HasTag

Note: All blue names indicate links to that particular section.

Perspectives Details with selected Repositories and Relationships

The Perspective Details report displays Perspective Name, Description, Repository Name on which the perspective is created, all the attribute in the perspective, and all the perspective relationships.

For each of the perspective relationship section Relationship Name, direction of the relationship (whether it is forward or backward), Perspective Name, Relationship attributes, Parent Relationship names, Target Repository Name, Target Repository attributes selected for perspective, and Next level Perspective Relationships are also displayed.



MDM Studio Repository Model Report

Perspective

Name	PartyDetails
Description	Perspective for Party details
Repository	PARTY
Attributes	PARTYNAME , DESCRIPTION , INDUSTRY , DUNSNUMBER , RATING , ANNUALREVENUE , NUMBEROFEMPLOYEES , TRADESTYLE , TYPE , WEBSITE , SITE , YEARSTARTED , REVENUEYEAR , OWNERSHIP , ISPERSONACCOUNT , OPENDATE , CLOSEDATE , ISCLOSED , LANGUAGECODE , ACCOUNTOPENINGCHANNEL , ACCOUNTOWNER , ISEMPLOYEEACCOUNT , ACQUIRED , ACQUIREDDATE , NOTTRADING , ENDTRADINGDATE , NOTE , SIC , SICDESC , TICKERSYMBOL , JIGSAW , NAICSCODE , NAICSDESC , NIELSENID , REGISTRIATIONNUMBER , BLOOMBERGID , CAGEID , DODAACID , TAXID , EXTERNALID , VATREGISTRATIONNUMBER ,
Perspective Relationships	Includes , AssociatedWithPerson , HasContact , HasAsset , HasTag , LocatedAt , ContactEmail , ContactPhone , HasAccount , OriginatingSystem , TaxAccount ,

Perspective Relationship

Name	Includes
Direction	FORWARD
Perspective Name	PartyDetails
Attributes	
Parent Relationship Name	
Repository	PARTY
Attributes	ID , PARTYNAME , DESCRIPTION , INDUSTRY , DUNSNUMBER , RATING , ANNUALREVENUE , NUMBEROFEMPLOYEES , TRADESTYLE , TYPE , WEBSITE , SITE , YEARSTARTED , REVENUEYEAR , OWNERSHIP , ISPERSONACCOUNT , OPENDATE , CLOSEDATE , ISCLOSED , LANGUAGECODE , ACCOUNTOPENINGCHANNEL , ACCOUNTOWNER , ISEMPLOYEEACCOUNT , ACQUIRED , ACQUIREDDATE , NOTTRADING , ENDTRADINGDATE , NOTE , SIC , SICDESC , TICKERSYMBOL , JIGSAW , NAICSCODE , NAICSDESC , NIELSENID , REGISTRIATIONNUMBER , BLOOMBERGID , CAGEID , DODAACID , TAXID , EXTERNALID , VATREGISTRATIONNUMBER ,
Perspective Relationships	

Perspective Relationship

Name	AssociatedWithPerson
Direction	FORWARD
Perspective Name	PartyDetails
Attributes	
Parent Relationship Name	
Repository	PERSON
Attributes	ID , FIRSTNAME , MIDDLENAME , ADDLMIDDLENAME , LASTNAME , GIVENNAME1 , GIVENNAME2 , SURNAME , NAMESUFFIX , NAMEPREFIX , PROFESSIONALTITLENAME , NICKNAME , BIRTHDATE , BIRTHPLACE , COUNTRYOFBIRTH , MARITALSTATUS , MARRIAGEDATE , GENDER , ETHINICITY , SALUTATION , HIGHESTEDUCATIONLEVEL , DECEASEDDATE , ISEMPLOYEE , CUSTOMERSTARTDATE , SOURCESYSTEMCODE , ISNATURALIZED , NATURALIZATIONDATE , CITIZENSHIP , ALIENNUMBER , REGISTRIATIONNUMBER , NATIONALREGISTRATIONNUMBER , NATIONALID , DRIVERLICENSEID , DRIVERLICENSEISSUEDON , DRIVERLICENSEISSUEDBY , FKA , AKA , HASNAMECHANGED , FACEBOOKID , TWITTERID , IDENTIFICATIONMETHOD , IDNUMBER , LANGCODECOMM , LANGCODEDOCUMENT , OCCUPATION ,
Perspective Relationships	IsRelatedTo , BelongsTo , PowerOfAttroney ,

file:///C:/Users/apps/workspace/Party/Exports/MDMReportFolder/Repository%20Models/b2b-Advanced-metadata/files/ReposData_b2b-Advanced-metadata.html#RepositoryAttribute__attribute_CONTACT_DESCRIPTION

Note: All blue names indicate links to that particular section.

Customizing the Repository Model Report

The Repository Model report can be customized to change the colors and font of the report.

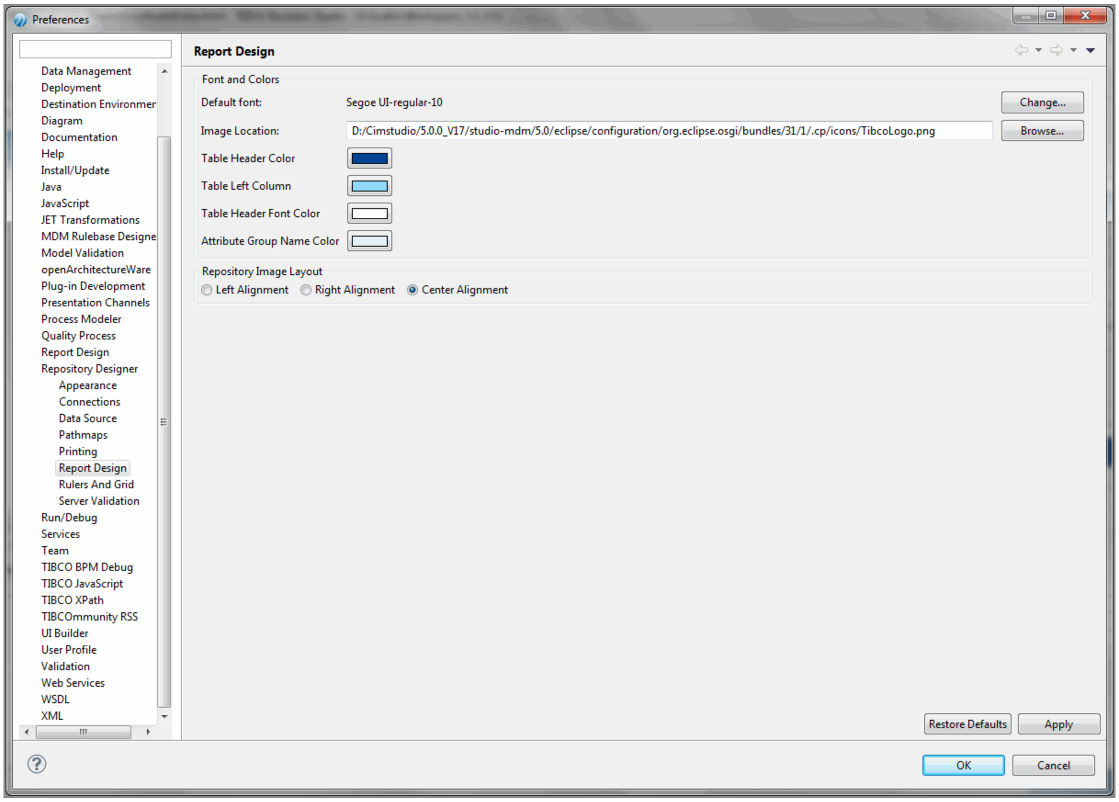
The repository model report can be used to customize the following:

- Default font
- Image
- Table header color
- Table left column
- Table header font color
- Attribute group name color
- Image layout in the Repository section

  Customizing the Repository Model Report

To customize the report model report, perform the following:

1. From **Window** menu, click **Preferences**.
2. Click on Repository Designer->Report Design.



Report Design

Font and Colors

Default font: Segoe UI-regular-10

Image Location: D:/Cimstudio/5.0.0_V17/studio-mdm/5.0/eclipse/configuration/org.eclipse.osgi/bundles/31/1/.cp/icons/TibcoLogo.png

Table Header Color

Table Left Column

Table Header Font Color

Attribute Group Name Color

Repository Image Layout

○ Left Alignment ○ Right Alignment ● Center Alignment

3. Click **Change** to select the font.
4. The **Font** window displays, select the appropriate font, font size, and font style and click **OK**.
5. To change the Image, click **Browse** and navigate to the new image in your window open dialog box and click **Open**. The path of the new image is displayed in the **Image Location** field.
6. Click the button corresponding to the **Table Header Color**, a new color window opens, select the new color and click **OK**.
7. Click the button corresponding to the **Table Left Column**, a new color window opens, select the new color and click **OK**.
8. Click the button corresponding to the **Table Header Font Color**, a new color window opens, select the new color and click **OK**.
9. Click the button corresponding to the **Attribute Group Name Color**, a new color

window opens, select the new color and click **OK**.

10. Select the appropriate radio option from the **Repository Image Layout**. The available options are **Left Alignment**, **Right Alignment**, **Center Alignment**.
11. Click **Apply**.
12. Click **OK**.

Input Map Editor

Input Maps are used to import a large amount of data from external sources. Input Map uses data sources as primary input and allow you to map the data source attributes to repository attributes. You can join more than one data source to map all or some of the repository attributes. You can define an expression for a mapping, to import the data.

Input Maps

A repository can have more than one input map and can use same or distinct data sources for defining mappings. The Input map name is unique for a repository.

Input Map can also have relationships across the repositories or in the same repository. For input maps to be related, you must predefine the relationships between the repositories. During import, related input maps (across repositories) can be used to map data to more than one repository by selecting the respective relationship names while creating the input map.

The Input Map Editor has multi tab editor. The each tab has the mapping for a specific attribute group in a repository. The first tab always has all the attribute groups and their mappings. In addition a filter is added to the Input map tree to search a repository attribute. The search supports wild card search entries.

Input Map is enhanced to provide flexibility to add/removed category attributes. Once all repository attributes are automatically added to input map, you can either add new category attributes or remove category attributes in the input map definition.

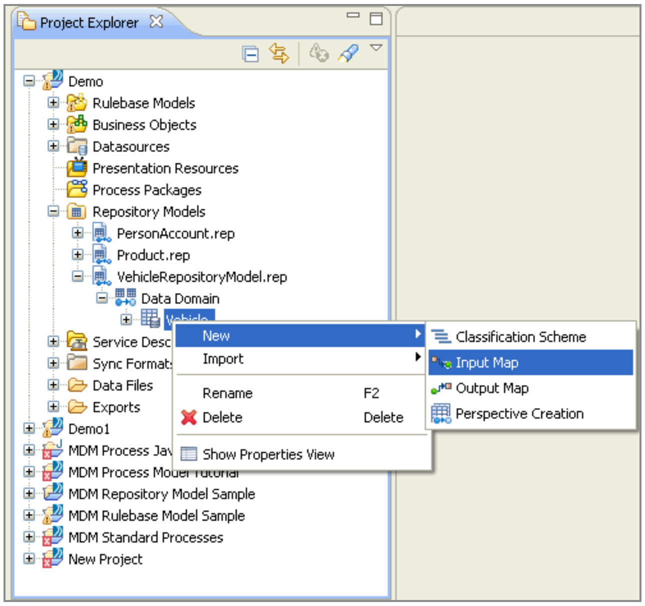
In addition you can also select the Classification Scheme and link the classification code with the category specific attributes. However, only Manual extraction mode is supported.

Creating Input Map

You can create input maps within or across repositories.

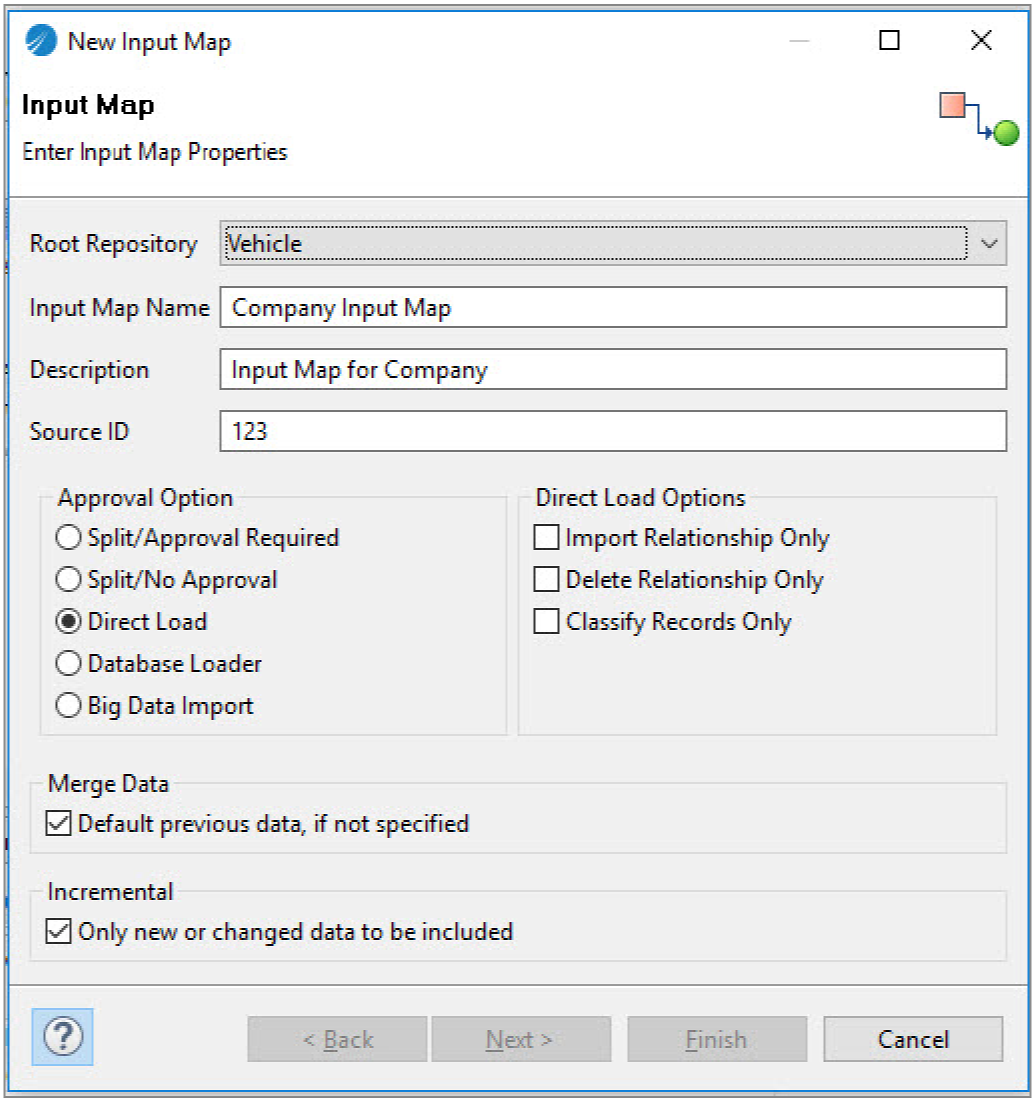
Procedure

1. In the Project Explorer window under **Repository Models**, right-click a repository on which you want to define the input map and select **New > Input Map**.

   The **Input Map** wizard is displayed.

2. In the **Input Map** wizard fields, enter the relevant details.



Input Map

Enter Input Map Properties

Root Repository: Vehicle

Input Map Name: Company Input Map

Description: Input Map for Company

Source ID: 123

Approval Option
- ○ Split/Approval Required
- ○ Split/No Approval
- ● Direct Load
- ○ Database Loader
- ○ Big Data Import

Direct Load Options
- □ Import Relationship Only
- □ Delete Relationship Only
- □ Classify Records Only

Merge Data
- ☑ Default previous data, if not specified

Incremental
- ☑ Only new or changed data to be included

The following table lists the fields and their description:

Field	Description
Root Repository	The name of the repository for which you want to create the input map is displayed.
Input Map Name	Enter the Input Map name. The Name is unique and case insensitive for a input map. The name

Field	Description
	can contain A-Z, 0-9 and _.
Description	The description for the Input Map.
Source ID	Specify a **Source ID**. The source ID allows you to associate the data with an external system. The source ID can used in workflow and business process rules to customize business processes.
Approval Option	Select one of the following approval options: • **Split/Approval Required**: When you select this option, the records are split into batches before processing, but the system waits for an approval before confirming changes for each record bundle. Conflicts, if any, must be resolved for each record bundle separately. • **Split/No Approval**: If you select this option, the records are split into batches before processing, but the system continues to process the record bundles without waiting for a confirmation. Conflicts, if any, must be resolved for each record bundle separately. The bundle is saved as a confirmed record, without initiating any approval workflows. • **Direct Load**: If you select this option, all records are processed in one go in multiple bundles and no events are created for each bundle. Changes are confirmed without approval. In case of conflicts, imported data take precedence and the conflict must be resolved

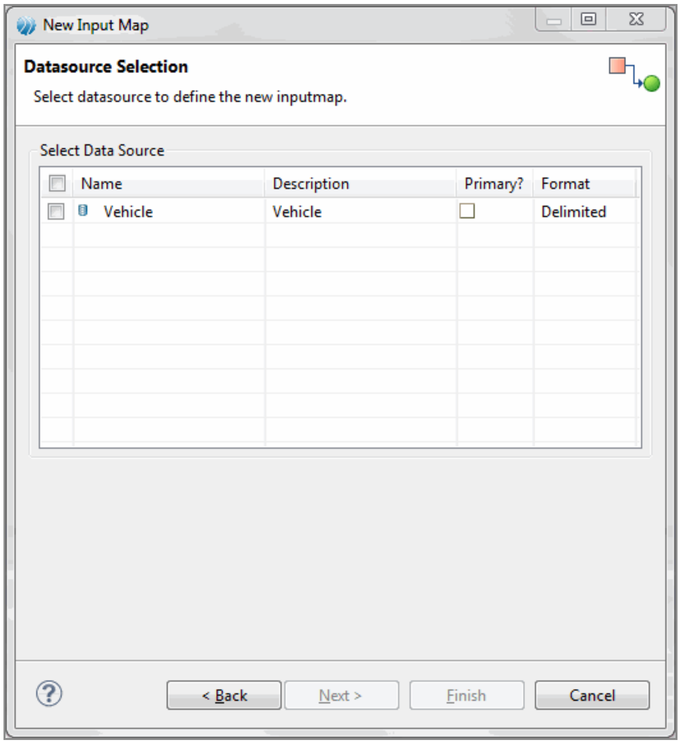
Field	Description
	separately. • **Database Loader**: If you select this option, the records are uploaded using the Database Loader. • **Big Data Import**: If you select this option, all records are processed in the Apache Spark cluster without any workflow. Big data import feature has a few limitations. For information, see *TIBCO MDM User Guide*. **Important:** ○ To use the **Big Data Import** option, you must have installed and configured Apache Spark and Apache Hadoop File Distributed System. For configuration, see *TIBCO MDM Installation and Configuration Guide*. ○ To use the big data import feature through TIBCO MDM UI, see *TIBCO MDM User Guide*.
Merge Data	Select the **Merge Data** check box. When checked, it indicates that any previous child relationships must be retained. By default, Merge Data check box is selected on New Input Map wizard.
Incremental	Select the **Incremental** check box if you want to include only new data or make changes to the existing data. By default, Incremental check box

Field	Description
	is selected on New Input Map wizard.
Direct Load Options	If you select **Direct Load** Approval option then **Direct Load Options** is displayed. • **Import Relationships Only**: If you select this check box, relationships are created for existing records. • **Delete Relationships Only**: If you select this check box, relationships between records are deleted for existing records. • **Classify Records Only**: If you select this check box, classification records are created for existing records.
Fresh Data/ Mode	If you select **Database Loader** Approval option then **Freshdata** and **Mode** option is displayed. • **Freshdata**: If you select this option, it indicates data is clean and initial version records need to be imported. • **Mode**: If you select this option, it displays Load Records and Load Relationships in the drop-down list. Select Load Records to load record data with relationship. Similarly select Load Relationships to load relationship between existing records.

3. Click **Next**.

   The **Datasource Selection** wizard is displayed.

4. in the **Select Data Source** section, select a data source by selecting the check box next to it.



Datasource Selection

Select datasource to define the new inputmap.

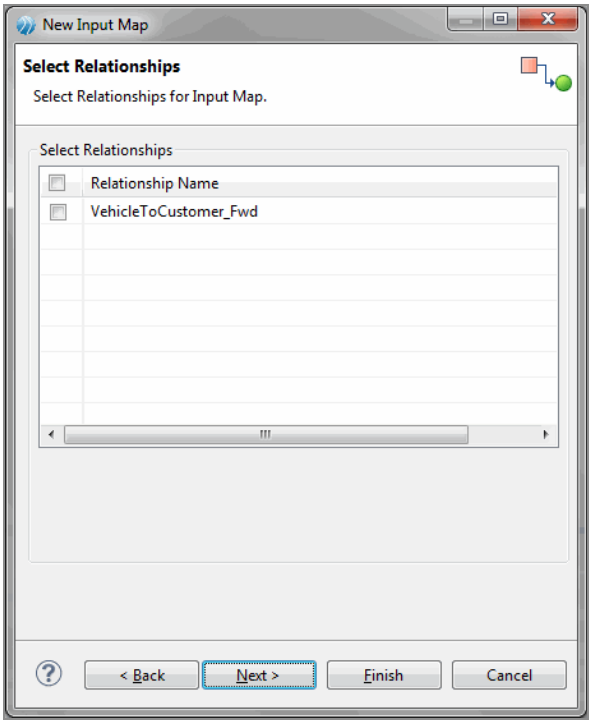
Select Data Source

☐	Name	Description	Primary?	Format
☐	Vehicle	Vehicle	☐	Delimited

5. In the **Primary** column, select the check box if you want to define the data source as primary data source.

6. Click **Next**.

   The **Select Relationships** wizard is displayed.



Select Relationships

Select Relationships for Input Map.

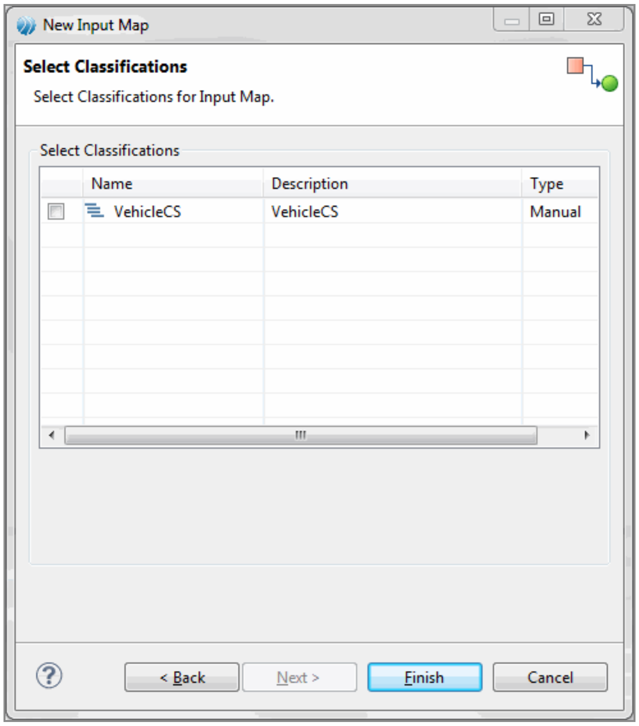
Select Relationships

□	Relationship Name
□	VehicleToCustomer_Fwd

7. If you want to define cross-repository or self-relationship (related input map), select the check box corresponding to the relationship name.

8. Click **Next**.

   The **Select Classifications** wizard is displayed.



Select Classifications

Select Classifications for Input Map.

Select Classifications

	Name	Description	Type
□	VehicleCS	VehicleCS	Manual

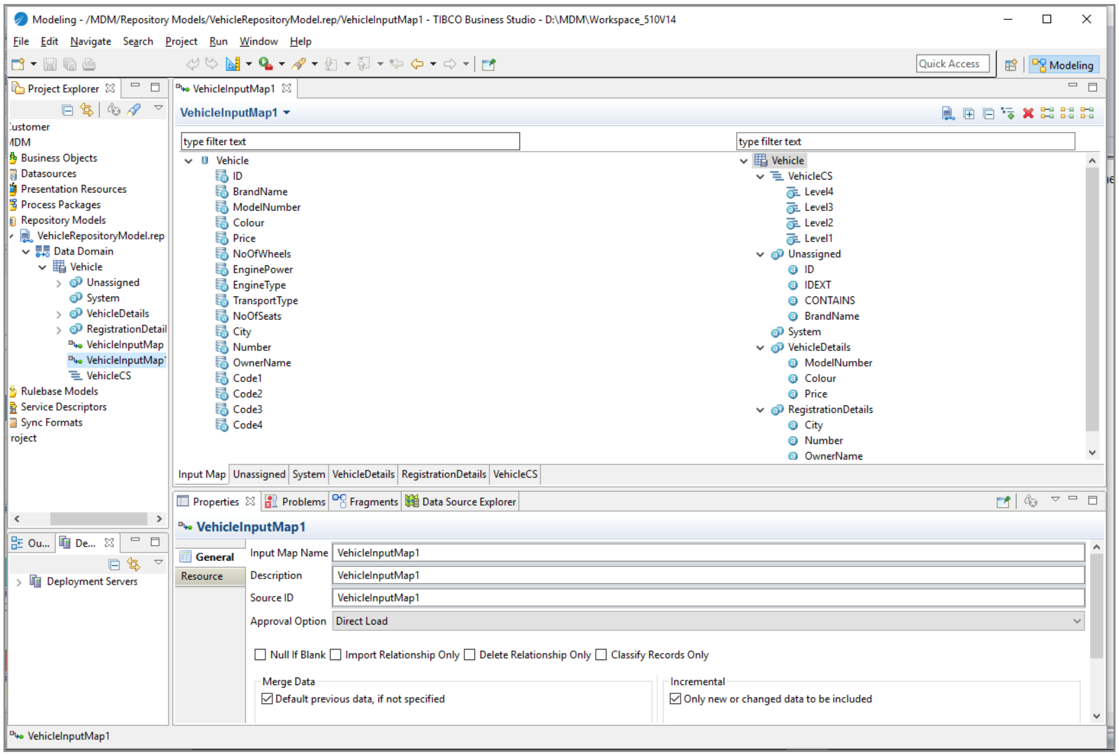
9. If you want to include classification scheme, select the check box corresponding to the classification scheme.

> **Note:** You can select the Classification Scheme for only for Manual extraction type.

10. Click **Finish**.

    The Input map is created and displayed in the Input Map editor.

    If the input map is created by selecting relationships, the related input maps are created under the related repositories. Similarly, if the input map is created by selecting classification scheme, the category specific attributes are displayed in the Input Map editor.



Input Map Name: VehicleInputMap1

Description: VehicleInputMap1

Source ID: VehicleInputMap1

Approval Option: Direct Load

☐ Null If Blank ☐ Import Relationship Only ☐ Delete Relationship Only ☐ Classify Records Only

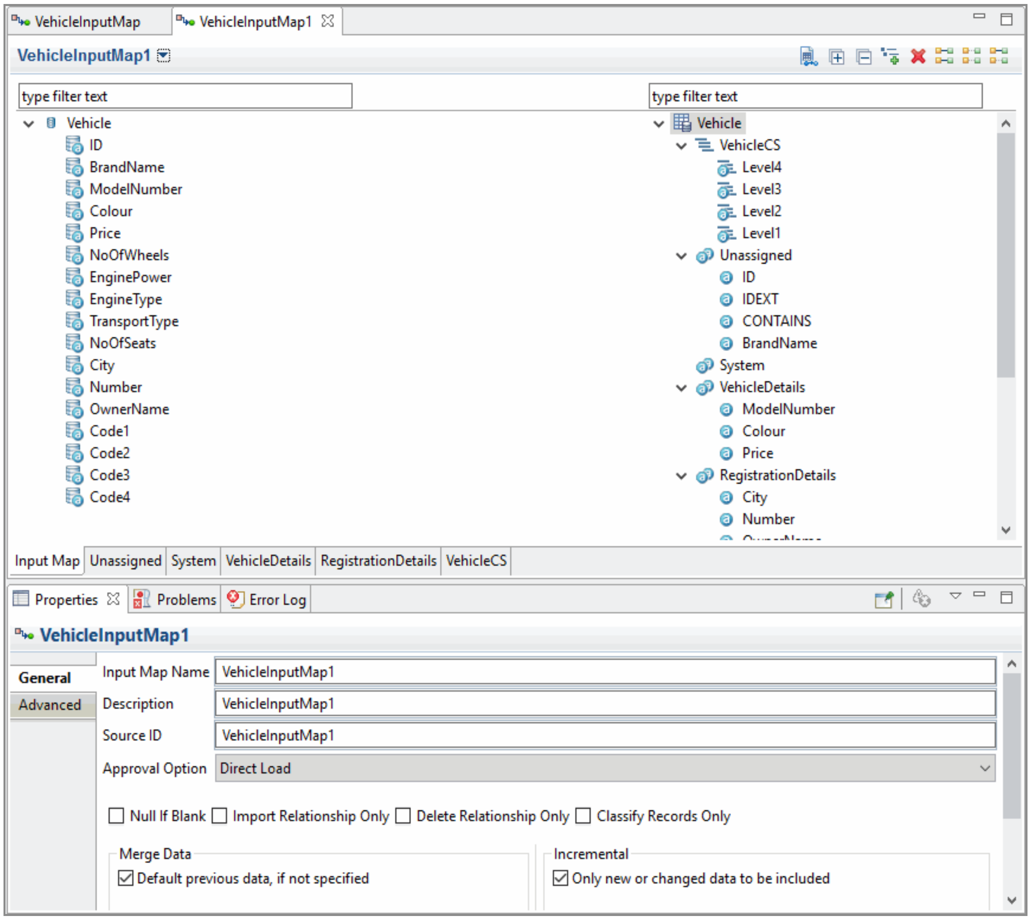
Merge Data: ☑ Default previous data, if not specified

Incremental: ☑ Only new or changed data to be included

Input Map Editor

The Input Map editor displays the newly created Input Map. It also displays the Input Map properties in Properties tab. The Input Map editor is enhanced to display the following:

- The Input Map Editor is partitioned into multiple tabs, each tab contains mapping for a specific attribute group.
- The Classification Scheme is included in the Input map Editor.
- Allows include and exclude of Category Specific Attributes.
- Allows searching of attribute using the text search filter.



Input Map Name: VehicleInputMap1

Description: VehicleInputMap1

Source ID: VehicleInputMap1

Approval Option: Direct Load

□ Null If Blank □ Import Relationship Only □ Delete Relationship Only □ Classify Records Only

Merge Data

☑ Default previous data, if not specified

Incremental

☑ Only new or changed data to be included

Input Map Controls	
▦	Repository Model of the input map.
⊞	Expand All the attributes groups in the input map.
⊟	Collapse All the attributes groups in the input map.
+	Add Category Attribute.

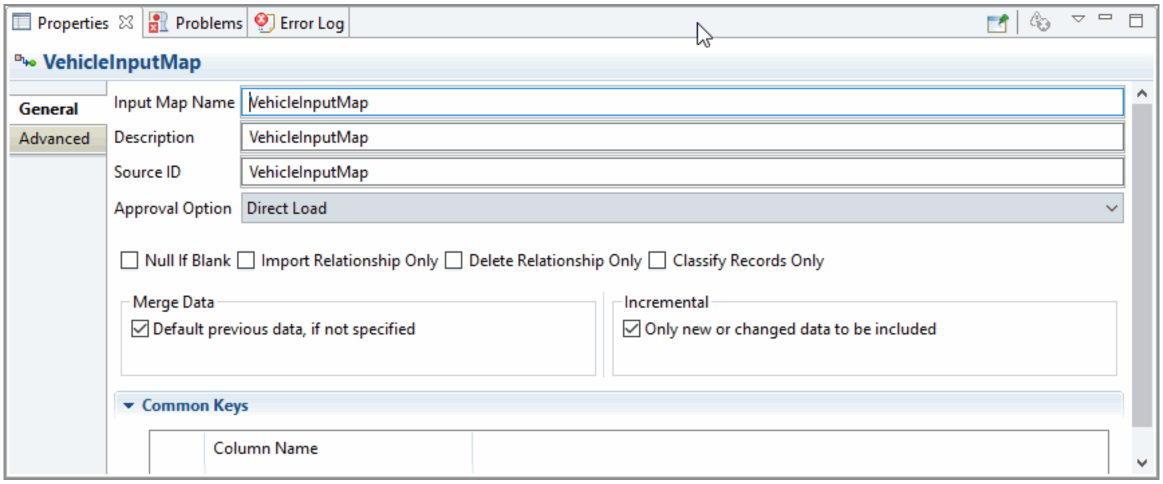
✖	Remove Category Attribute.
	Show Mapped.
	Show Unmapped.
	Show All Mapping.

Input Map Property Section

The Input Map properties can be viewed and modified using the Input Map property view. The Inputmap properties section comprises of the General and Advanced tab.

General Tab

All the information in the Inputmap section is described in the general tab.



Input Map Name: VehicleInputMap
Description: VehicleInputMap
Source ID: VehicleInputMap
Approval Option: Direct Load

□ Null If Blank □ Import Relationship Only □ Delete Relationship Only □ Classify Records Only

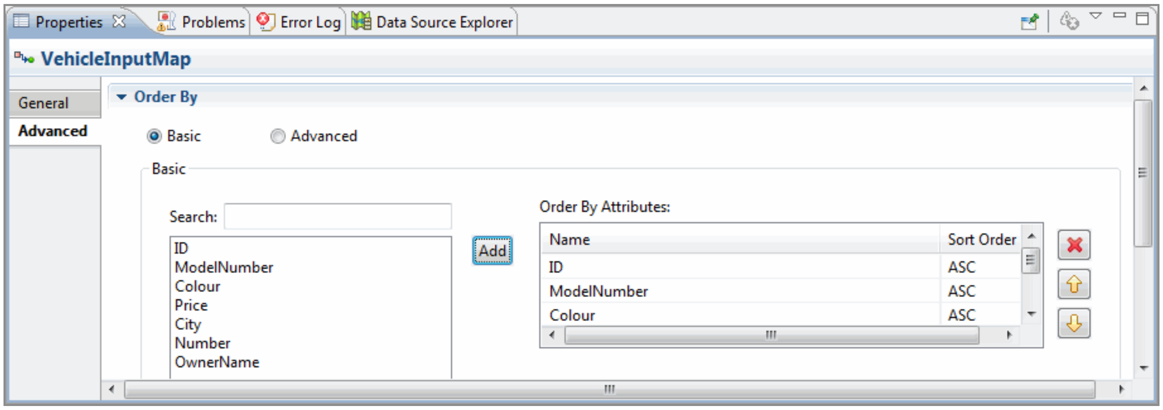
Merge Data
☑ Default previous data, if not specified

Incremental
☑ Only new or changed data to be included

You can update the **Input Map Name**, **Description** and **Approval Option**. Similarly you can also update the Merge Data and Incremental parameters. Select the **Null If Blank** check box if you want to consider a blank value. Select the **Import Relationship Only** check box if you want to create relationships for existing records. Select the **Delete Relationship Only** check box if you want to delete the relationships for existing records. Select the **Classify Records Only** check box if you want to create classification records for existing records. If you have more than one data sources for the input map, you can select **Common Keys** columns between the data sources.

Advanced Tab - Basic

An advanced property tab is used to define the order by attributes. The advanced property tab is visible only for the root input map and is not supported for related and classification input maps.



ID
ModelNumber
Colour
Price
City
Number
OwnerName

Name	Sort Order
ID	ASC
ModelNumber	ASC
Colour	ASC

Procedure

1. In the **Basic** mode, the left side list displays all the mapped attributes in input map. Select the attribute and click Add.

2. In addition a Search filter is also provided for quick search of attribute. The selected attributes are displayed in the **Order By Attributes** table.

   Note: The multi-valued attributes are not supported for order by.

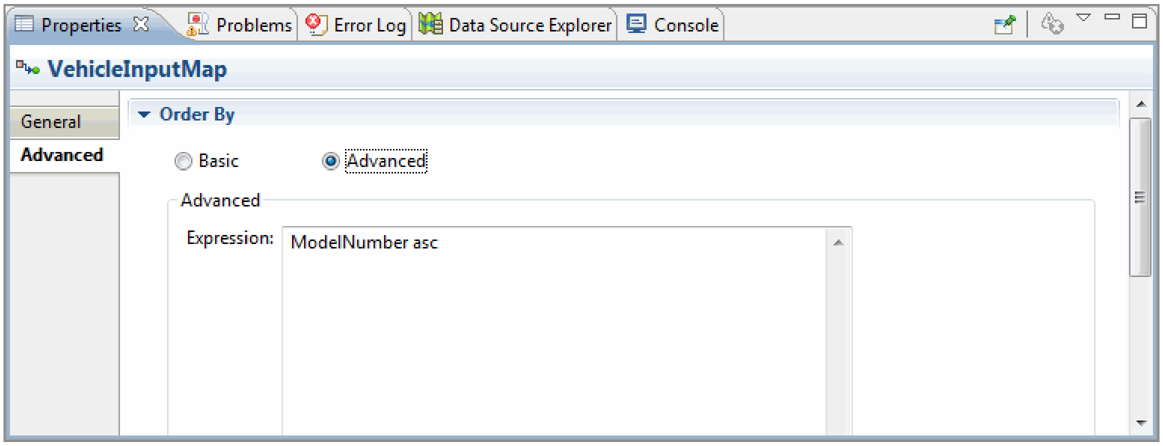
3. Specify the sort order from the Sort Order column. The available options are ASC (ascending) and DSC (descending).

4. Select the attribute and click ✖ to remove the attribute from the order by attributes table. You change the order of the attributes by clicking the ⇧ and ⇩.

Advanced Tab - Advanced

On the **Advanced** tab, you can define the exact `Order By` expression to be used during record import. The expression must be a valid `Order By` expression and is specific to a database.

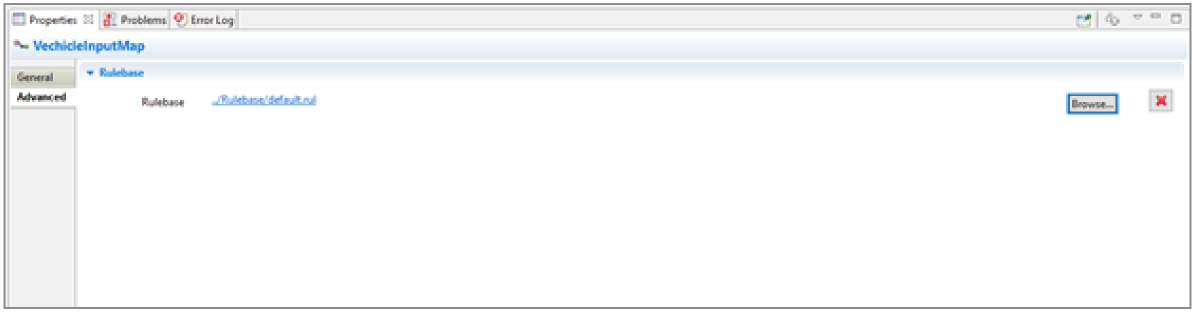
In the staging table of TIBCO MDM, all attributes are of the type string, the data type conversion must be handled in the `Order By` expression. The data type conversion may vary for different type of databases. For example in case of Oracle database, `TO_TIMESTAMP (CATTRA_TIMESTAMP,yyyy-MM-dd HH24:MI:SS.FF) DESC`.



Expression: ModelNumber asc

Advanced Tab for Big Data Import

When creating an input map, if you have selected the **Big Data Import** approval option, the **Rulebase** option is displayed on the **Advanced** tab.



Rulebase ../Rulebase/default.rul

Note: The **Rulebase** option is only supported for the input map with the **Big Data Import** approval option.

Browse to select the rulebase that is defined for the parent repository and of type **Other**. For information on creating a rulebase model and different types of rulebase types, see the

"Creating a Rulebase Model" and "Actions Allowed for Different Types of Rulebases" sections in TIBCO MDM Studio Rulebase Designer User Guide.

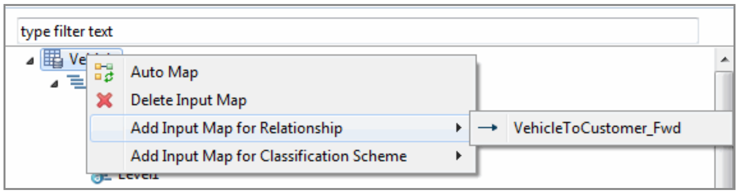
Creating an Input Map with a Cross-repository Relationship

At times, there is a need to capture data spread across data sources into multiple logically related repositories. To achieve this purpose, Input Maps of one repository can be related to Input Maps of other repositories to distribute data across related repositories.

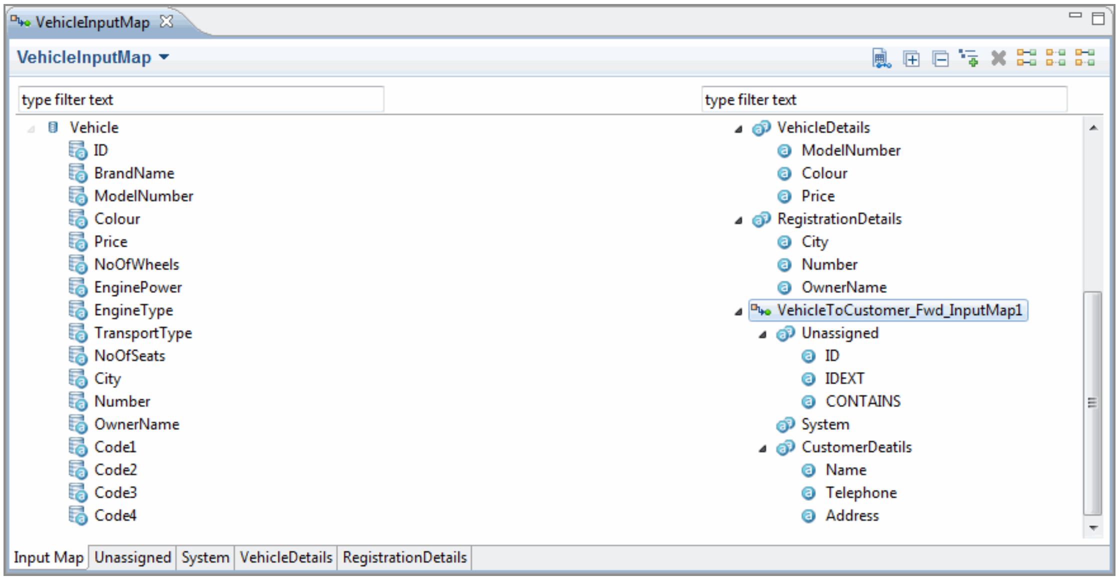
To relate input maps of any two repositories, it is essential that there exists a predefined relationship between these repositories. Related input maps must be based on same or subset of data sources selected and can be used or associated with other input maps.

Procedure

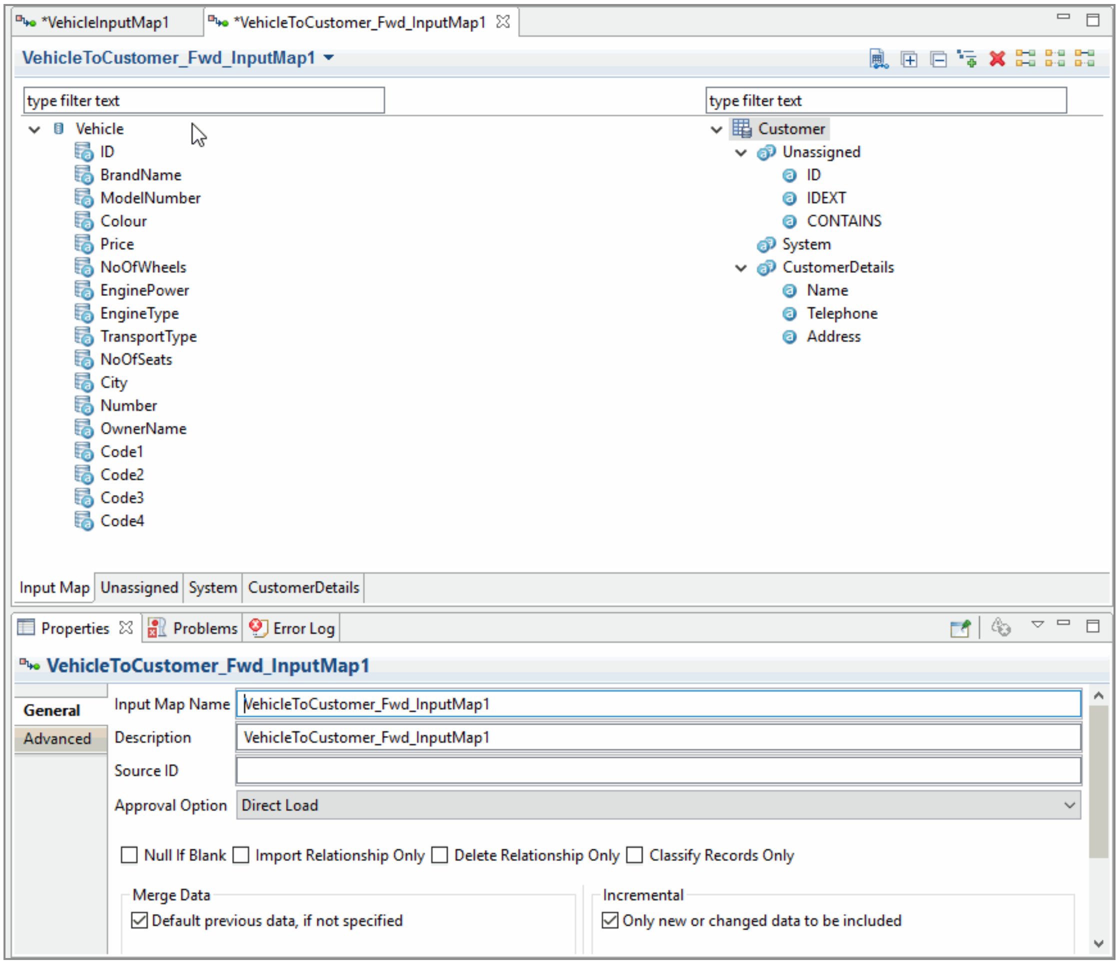
1. Right click on Repository name from the Input Map tree section on which you want to include related repository and click on Add **Input Map** for **Relationship**.


type filter text
Auto Map
Delete Input Map
Add Input Map for Relationship
Add Input Map for Classification Scheme
VehicleToCustomer_Fwd

2. Select relationship from the sub menu. A new related Input Map will be added in the Input Map tree.



3. Double click the related Input Map, it will open in an another editor.



Input Map Name: VehicleToCustomer_Fwd_InputMap1

Description: VehicleToCustomer_Fwd_InputMap1

Source ID:

Approval Option: Direct Load

□ Null If Blank □ Import Relationship Only □ Delete Relationship Only □ Classify Records Only

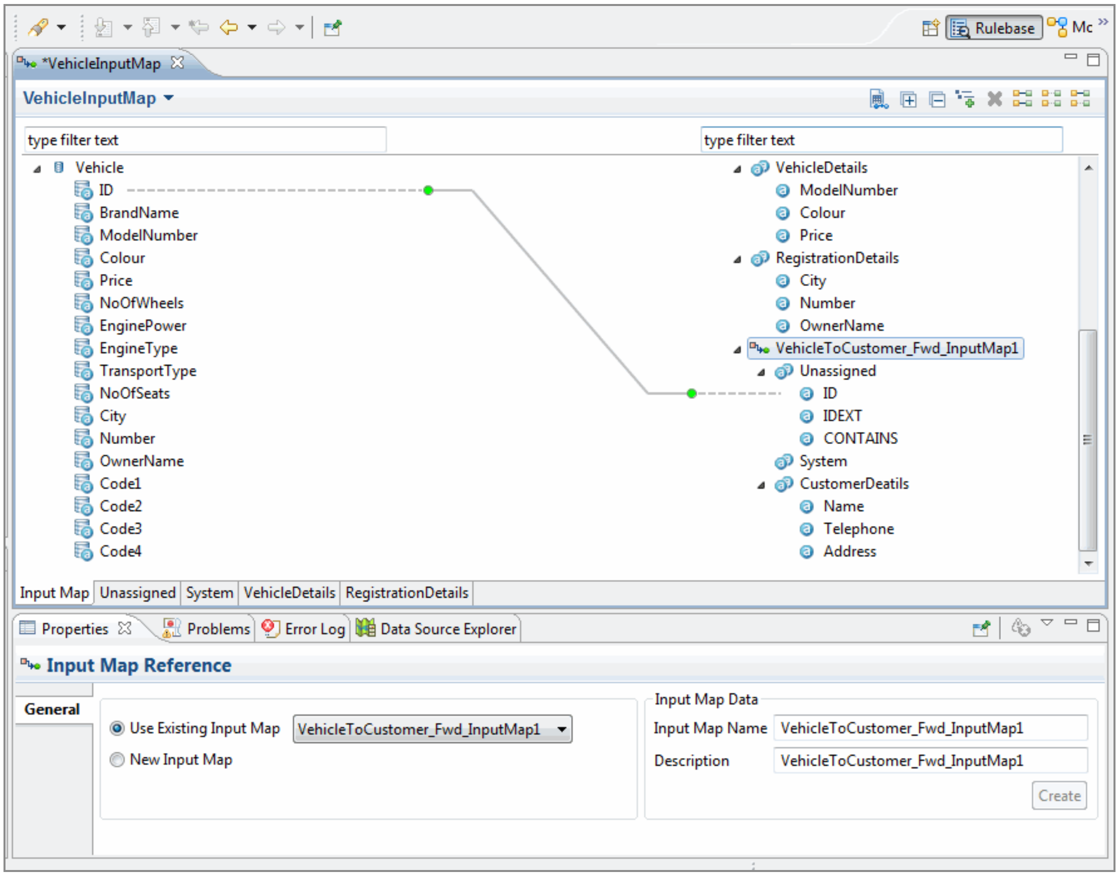
Merge Data

☑ Default previous data, if not specified

Incremental

☑ Only new or changed data to be included

4. You can create a related Input map using an existing input map or create new input map to link with the parent input map.



● Use Existing Input Map: VehicleToCustomer_Fwd_InputMap1

○ New Input Map

Input Map Data

Input Map Name: VehicleToCustomer_Fwd_InputMap1

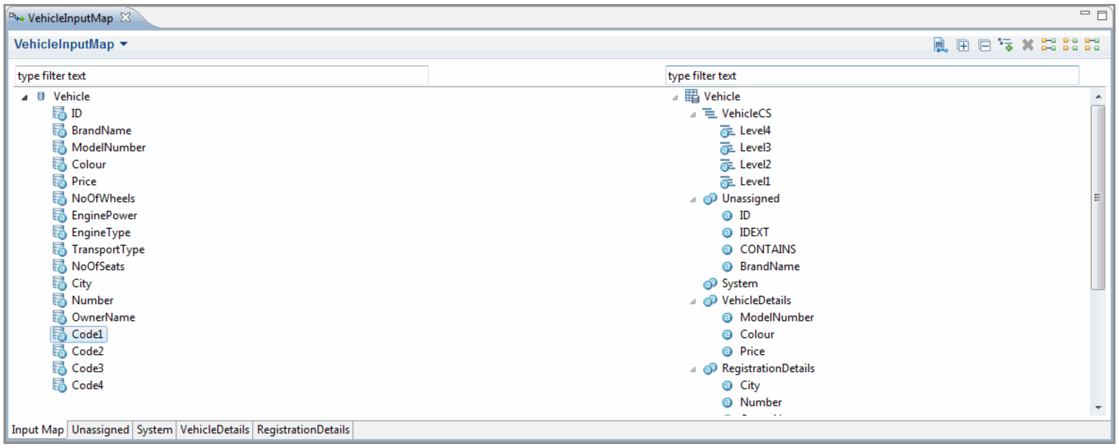
Description: VehicleToCustomer_Fwd_InputMap1

5. If you have existing input maps in the related repository, select the **Use Existing Input Map** option. When you select the **Use Existing Input Map** option, the drop-down list displays the input maps available in the related repository. Select **Create New Input Map** option to create a new input map for the related repository.

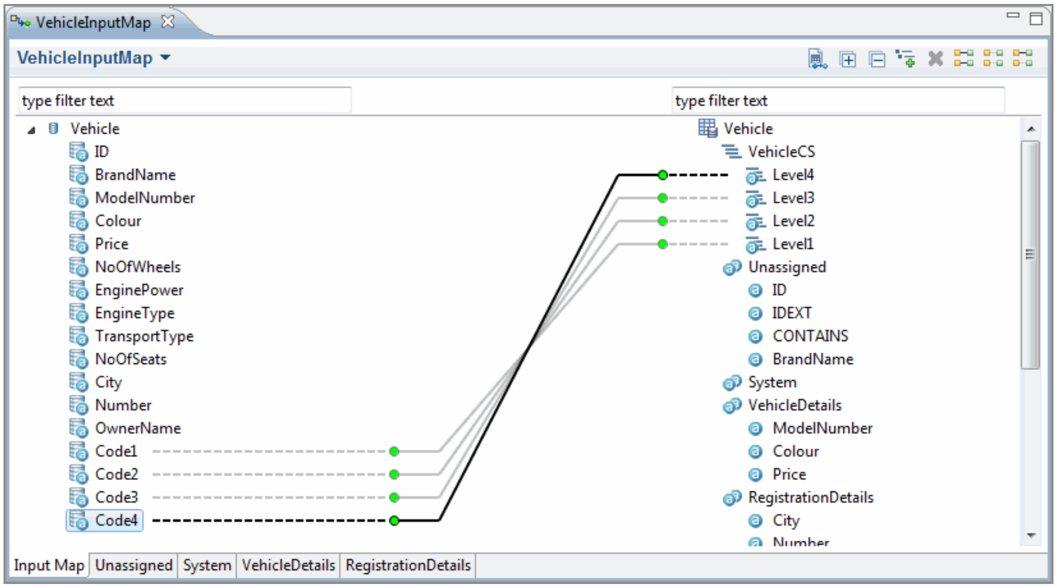
Classification Code Mapping

Procedure

1. Drag and drop the Data Source Attribute (Data Source Tree on the left side of the editor) to Classification Scheme Logical Level.



2. The classification code mapped to the Classification Scheme Logical Level is displayed.



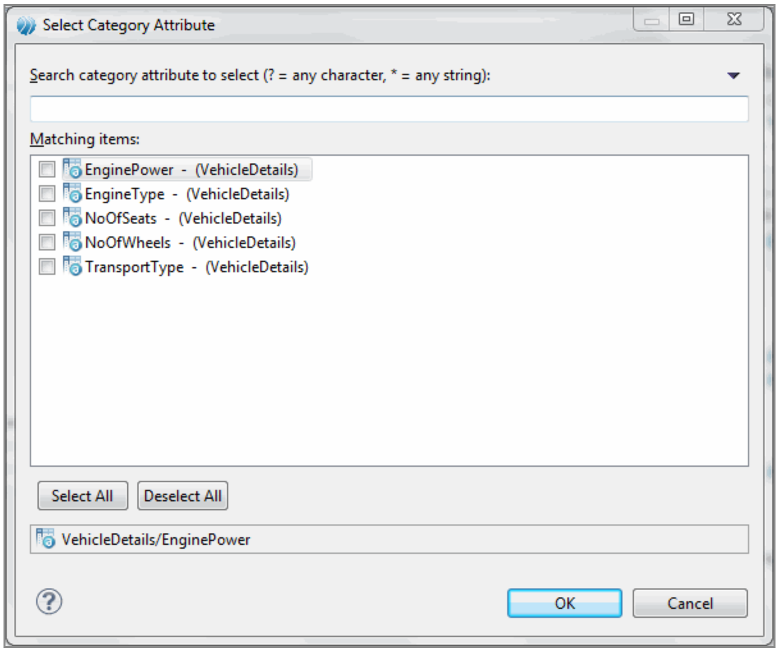
3. The property section of the classification scheme logical level displays the logical levels specified during the Classification Scheme creation. Enter the expression for the logical level.

Adding Category Attributes

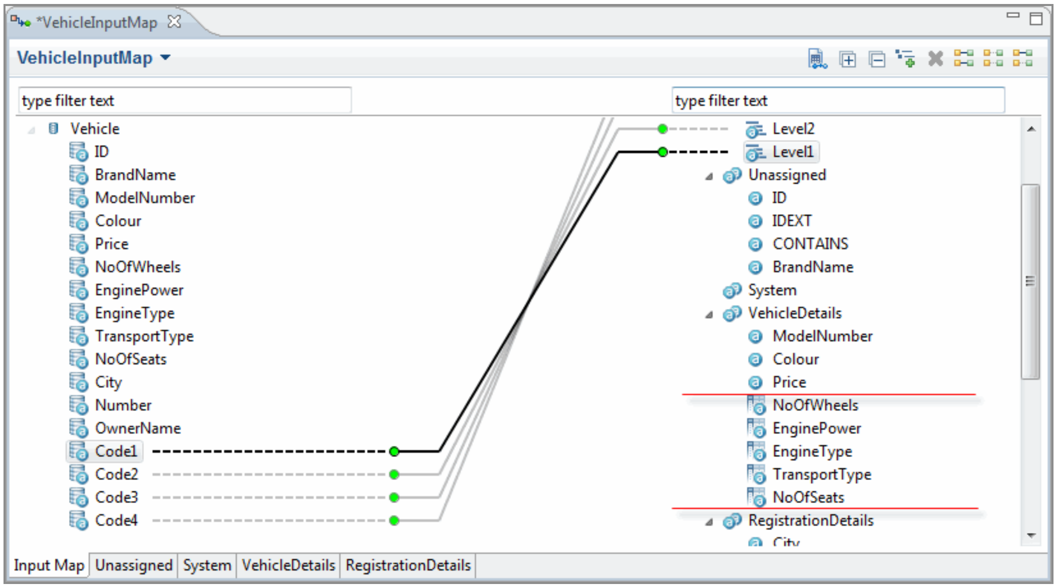
By Default Category Specific Attributes are not listed in Input Map Editor. You can add an existing Category Specific Attribute to the Input map Editor.

Procedure

1. Open Input Map and click to add new category attributes.
2. Category attribute dialog box is displayed, select the checkbox corresponding to the category attribute selection or search the category attribute using the pattern based search. For example, *e.


Select Category Attribute
Search category attribute to select (? = any character, * = any string):
Matching items:
□ EnginePower - (VehicleDetails)
□ EngineType - (VehicleDetails)
□ NoOfSeats - (VehicleDetails)
□ NoOfWheels - (VehicleDetails)
□ TransportType - (VehicleDetails)
Select All
Deselect All
VehicleDetails/EnginePower
OK
Cancel

3. Click **OK**.
4. The added category attributes are displayed in the selected input map.

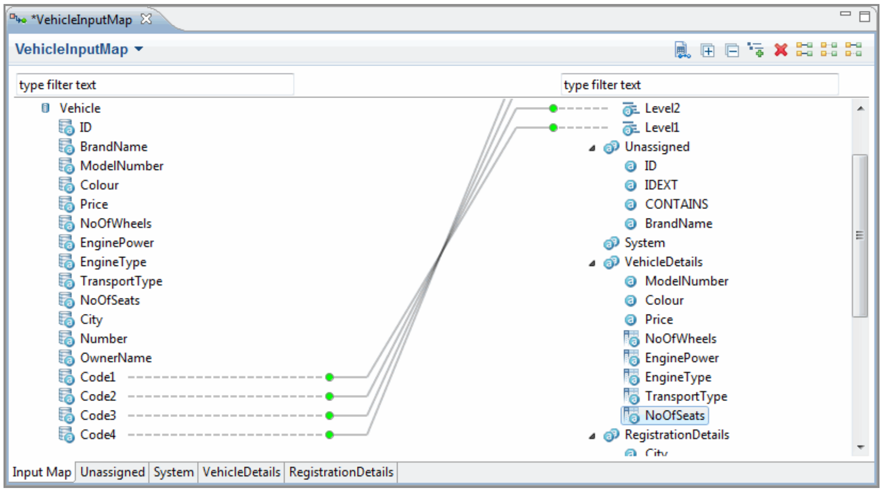


Removing Category Attributes

You can remove the category attributes from the input map.

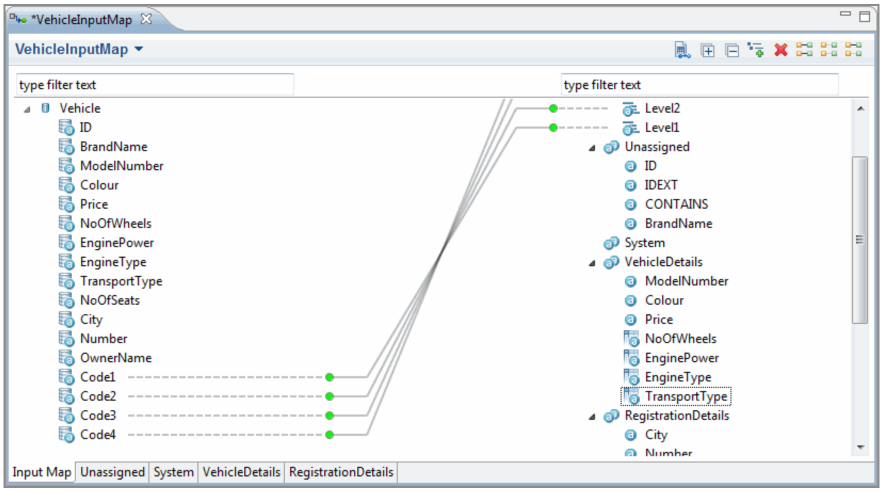
Procedure

1. Select category attributes in the input map which you want to remove. Select the category attribute which you want to delete.



2. Click ✖ to remove the selected category attributes from the input map.

3. The category attribute removed do not appear in the input map.



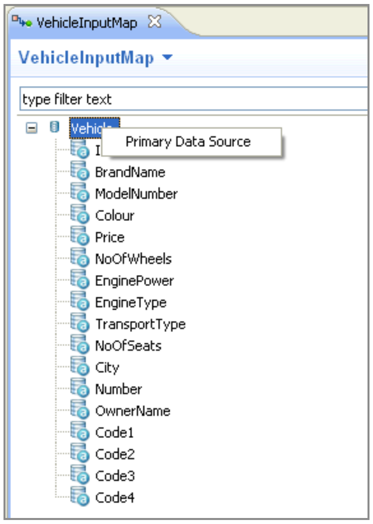
Selecting Primary Data Source

If there are multiple data sources, you can identify a primary data source and define it as Primary.

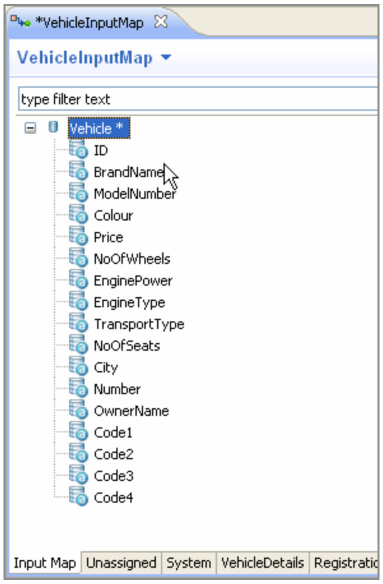
Procedure

1. In the Input Map Editor, select the data source which you want to define as Primary.
2. Right click on the data source name and click **Primary Data Source**.

Result



After selecting a particular data source as a primary data source, * will appear on the primary data source.

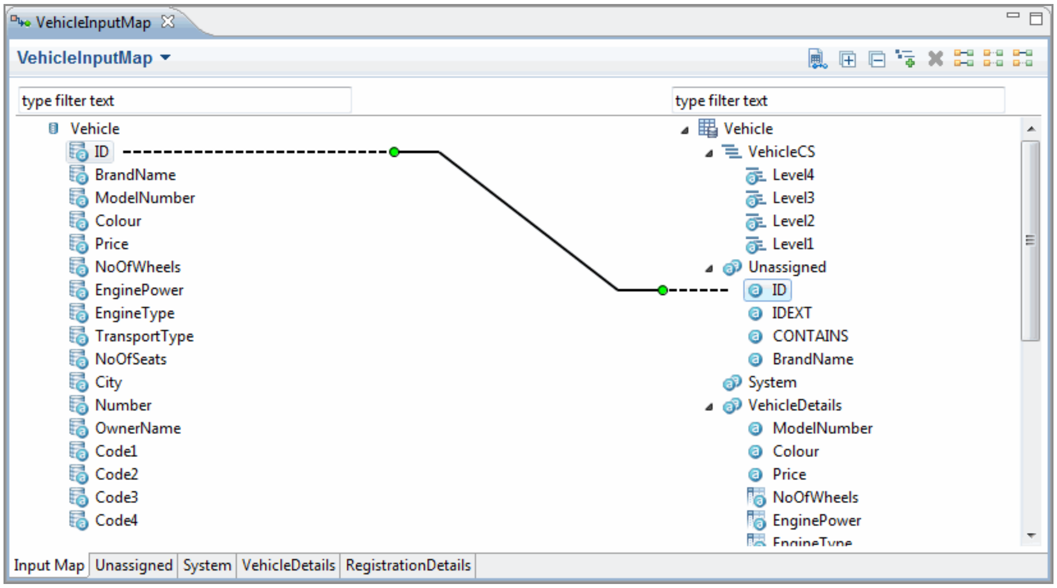


Primary data source identification is optional and is recommended only if there are more than one data sources.

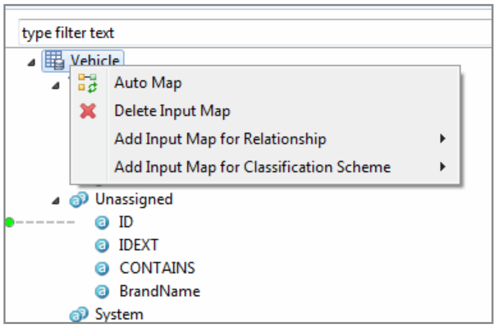
Defining Input Map Mapping

Procedure

1. To manually map the input map, Drag and drop the Data Source Attribute (Data Source Tree on the left side of the editor) to Repository/Relationship Attribute (Input Map Tree on the right side of the editor).



2. To automatically map the input map, Right click on the Repository and click **Auto Map**.

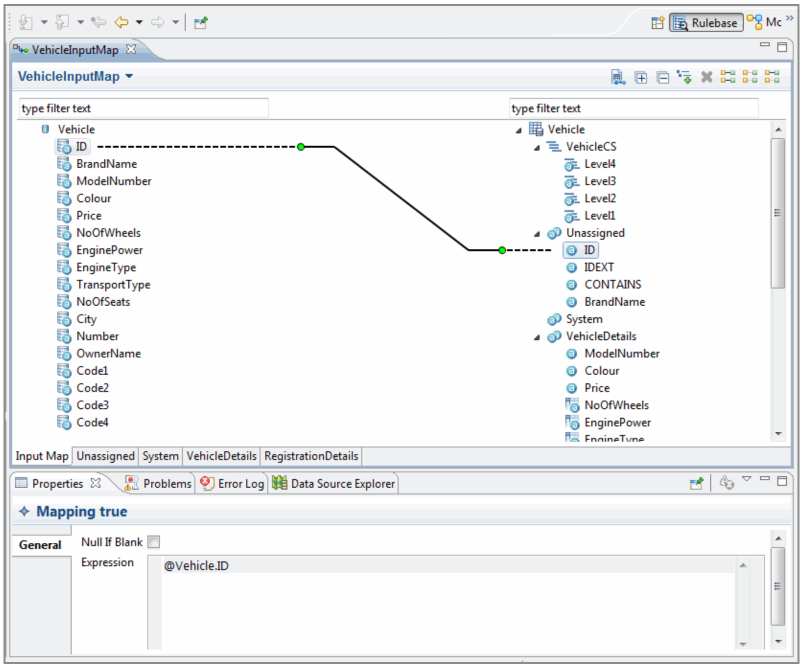


3. You can define multi-value attribute mapping in the input map editor. The multi-value attribute is identified by .

4. Drag and drop the Data Source Attribute (Data Source Tree on the left side of the editor) to multi-value Attribute (Input Map Tree on the right side of the editor).

   Alternatively you can also define multi-value attribute mapping using the property view. Select the multi-value attribute and click ✚ in the property view to define the mapping. Enter the mapping expression in the text box.

5. To enter the source expression for the Repository/Relationship Attribute, Click on the Attribute under Input Map tree (Right hand tree). The text editor will appear in the property section.



Null If Blank ☐

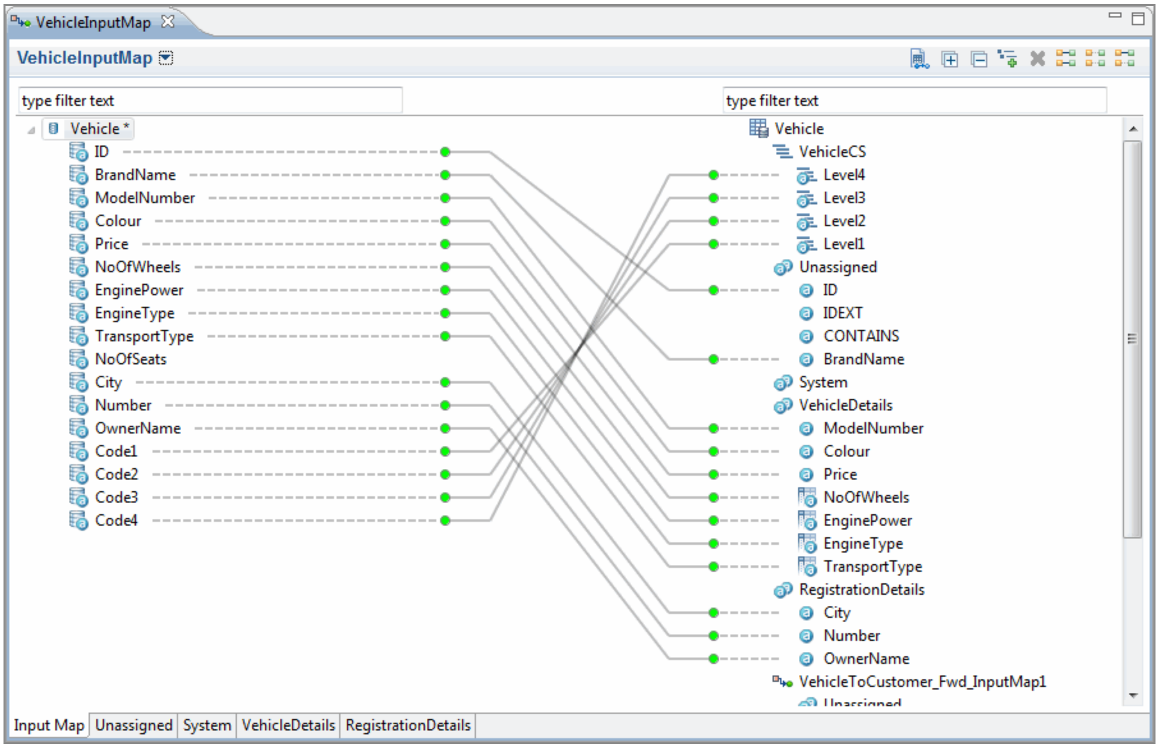
Expression @Vehicle.ID

6. Select the **Null If Blank** check box if you want to consider a blank value in the data source.

View Mapped Unmapped Mappings

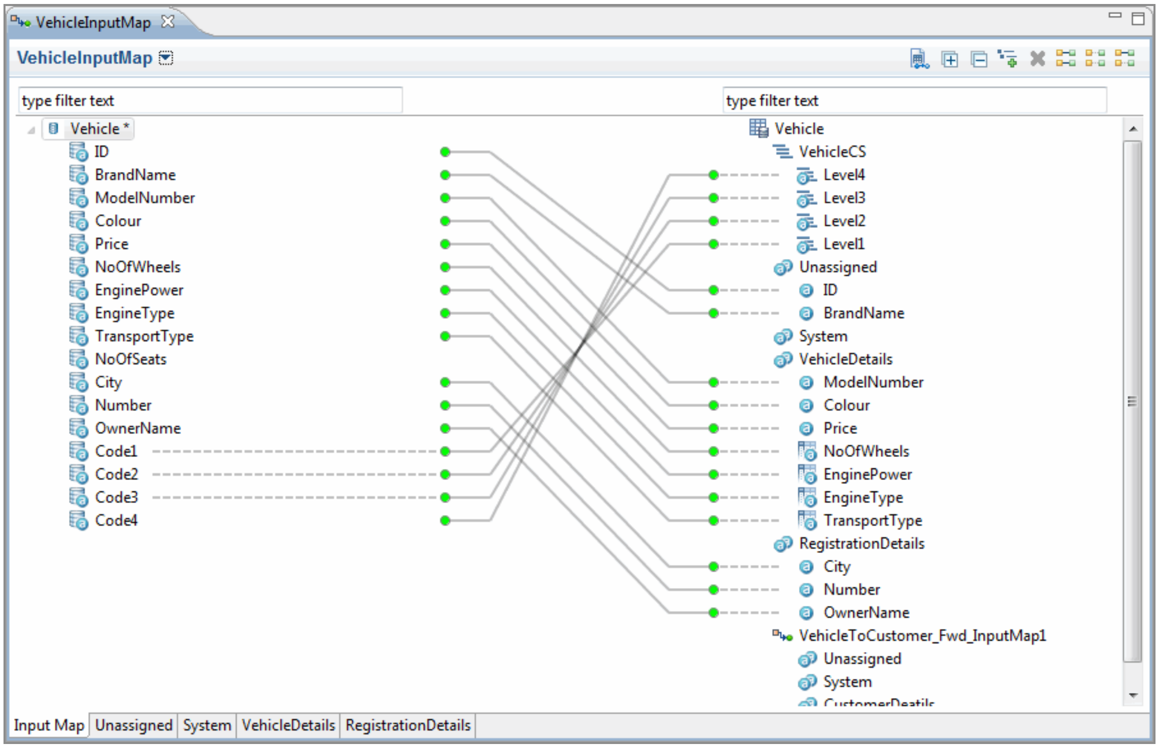
To view all mapping (both mapped and unmapped attributes) click .

The attributes values are expanded and the mapping is displayed.



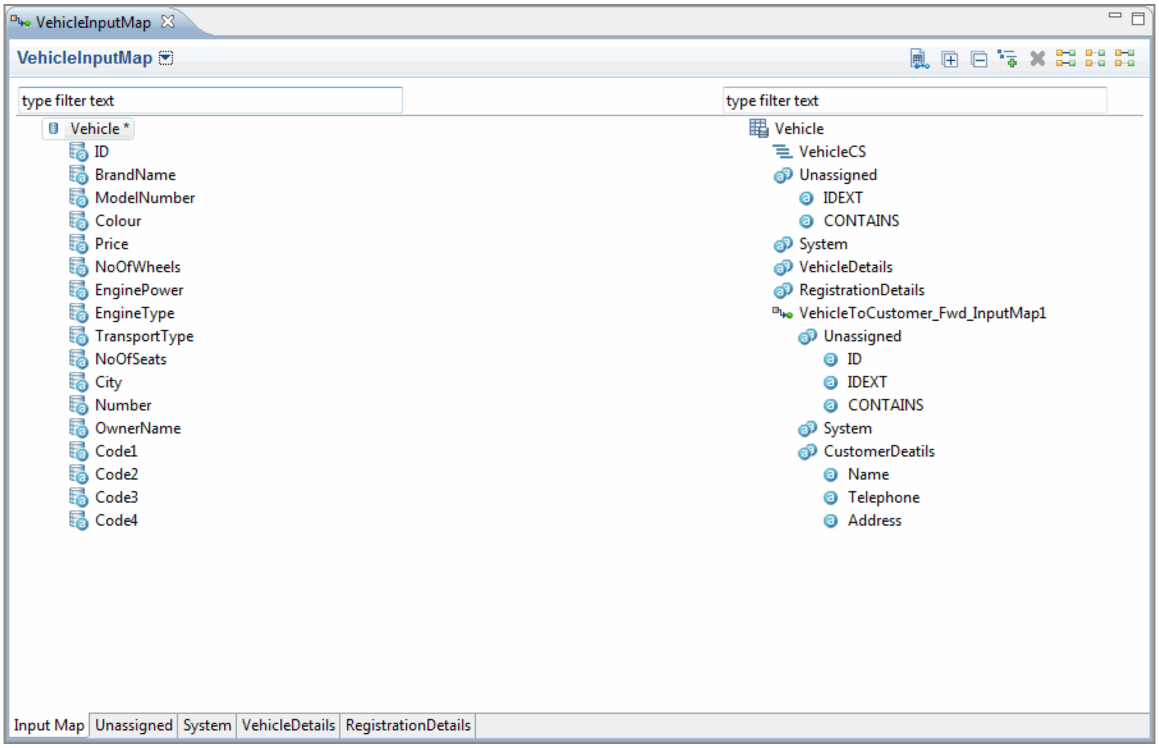
To view only mapped attributes) click .

Only the mapped attributes are displayed.



To view unmapped attributes) click .

All the unmapped attributes are displayed.

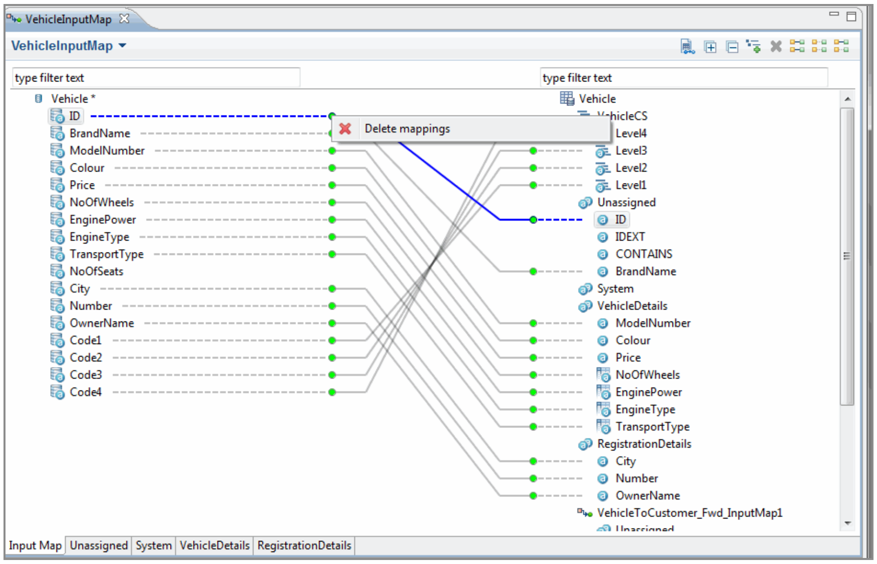


Deleting Mappings

Procedure

1. From the Input Map Editor, select the mapping which you want to delete.
2. Right click on the mapping and click **Delete** mappings.

Result



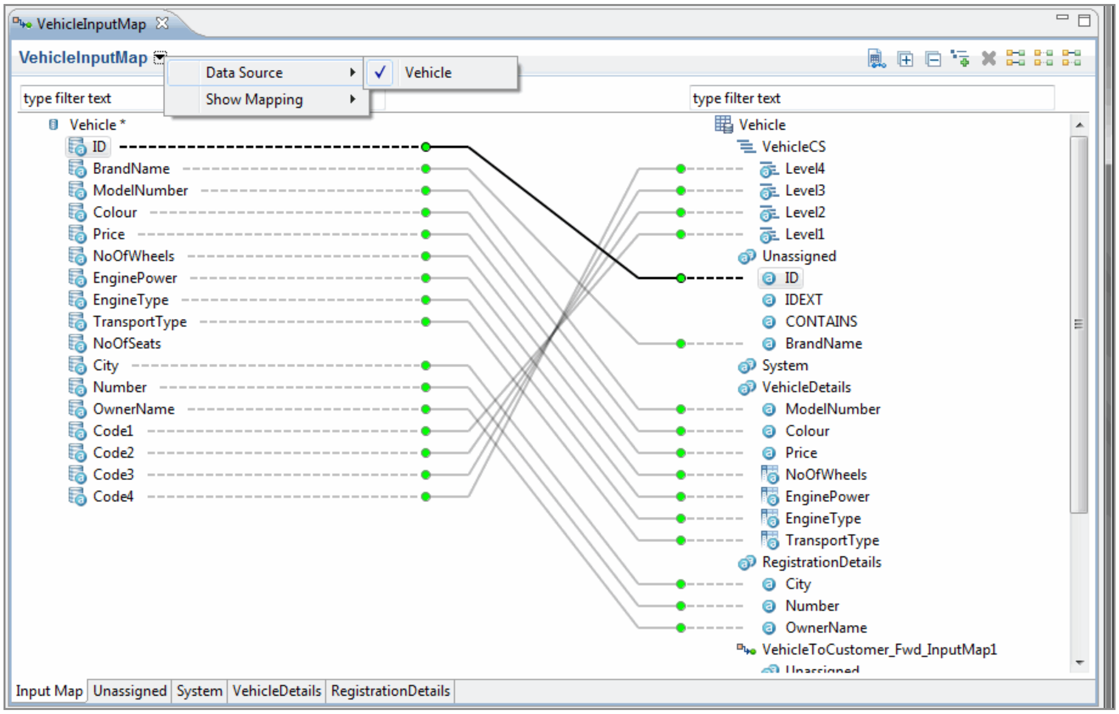
Note: If repository attributes are used in a data source and input maps model and if the attributes are either deleted from the repository model or moved from one attribute group to another, then the mapping between data source attributes and repository attributes will be deleted from the input map. Establish the mapping once again if the attributes are moved from one group to another.

Adding Removing Data Source

A Data Source can be added or removed from the existing input map in the project. All the data sources defined in project can be viewed from the Input Map Editor menu.

Procedure

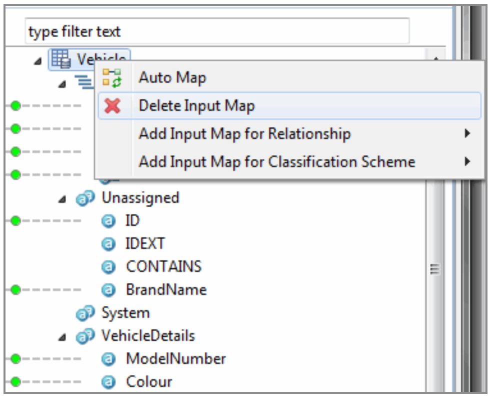
1. From the Input Map Editor, select the Input Map to which you want to add or remove a data source.
2. Click ▼ and point to Data Source and select the data source which you want to add or clear the data source which you want to delete.



Deleting Input Maps

Procedure

1. Right -click on the Input Map (Input Map Tree) which you want to delete and click **Delete Input Map**.



Deploy Input Map

When repository model is deployed associated input maps also get deployed. However input map can also be deployed separately without deploying full model.

For more information on how to deploy Input Map see Deployment

Output Maps

This chapter introduces how to design output maps between the attributes in the repository, relationship attributes and the various outbound synchronization formats.

Output maps associate repository attributes with corresponding attributes in various outbound synchronization formats.

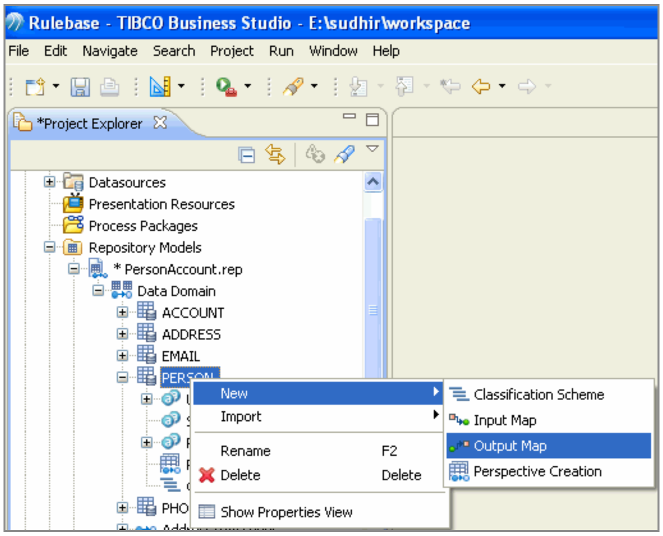
In repository relationships, to synchronize data scattered across repositories, you can create related output map by selecting the forward, self and reverse relationship. The relationship select page is enhanced to populate the self and reverse relationships along with the forward relationships.

Creating Output Map

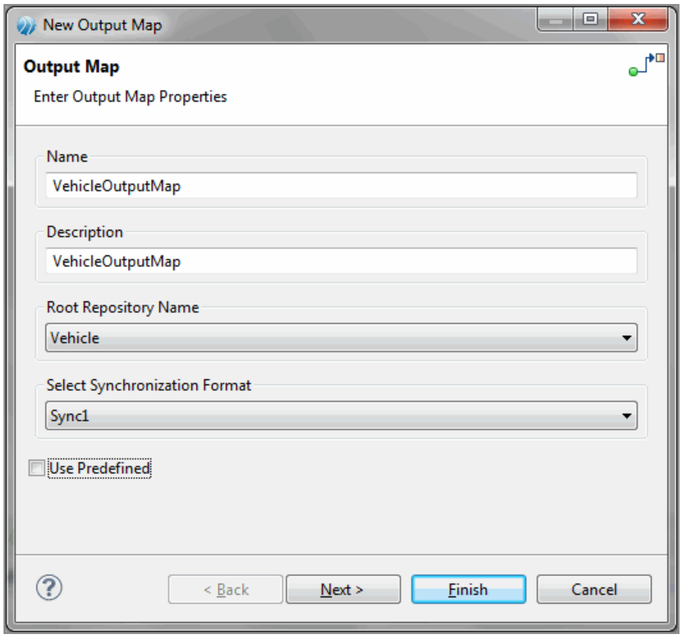
Output maps can be created within the same repository or across repositories.

Procedure

1. In the Project Explorer under the Repository model file, Right click on Repository on which user want to define the Output Map and select **New-- > Output Map.**



2. The Output Map Wizard is displayed.


New Output Map

Output Map

Enter Output Map Properties

Name: VehicleOutputMap

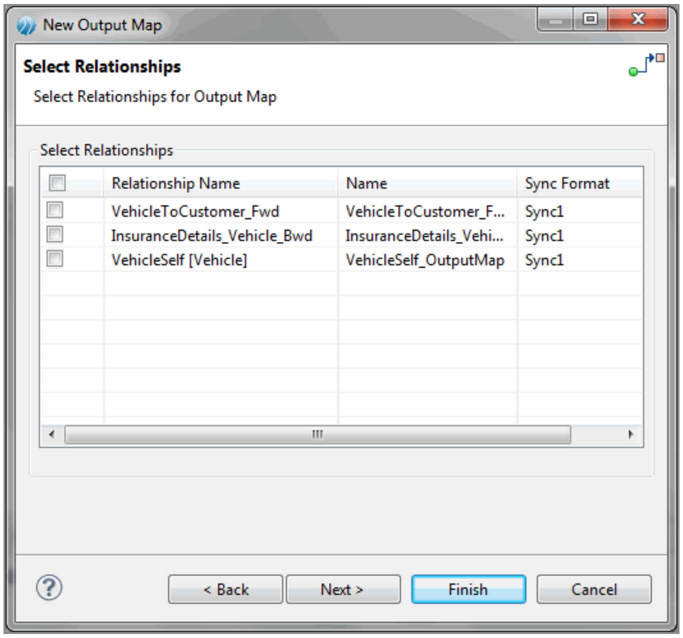
Description: VehicleOutputMap

Root Repository Name: Vehicle

Select Synchronization Format: Sync1

□ Use Predefined

3. Enter a name and description for the Output Map. The Name is unique and case insensitive and can contain A-Z, 0-9 and _.
4. Select the appropriate synchronization format from the **Select Synchronization Format** drop-down list.
5. Select the **User Predefined** checkbox if you want to use the predefined output map. For more information, refer Importing Predefined Output Map
6. Click **Next**. The Select Relationships screen is displayed.



Select Relationships

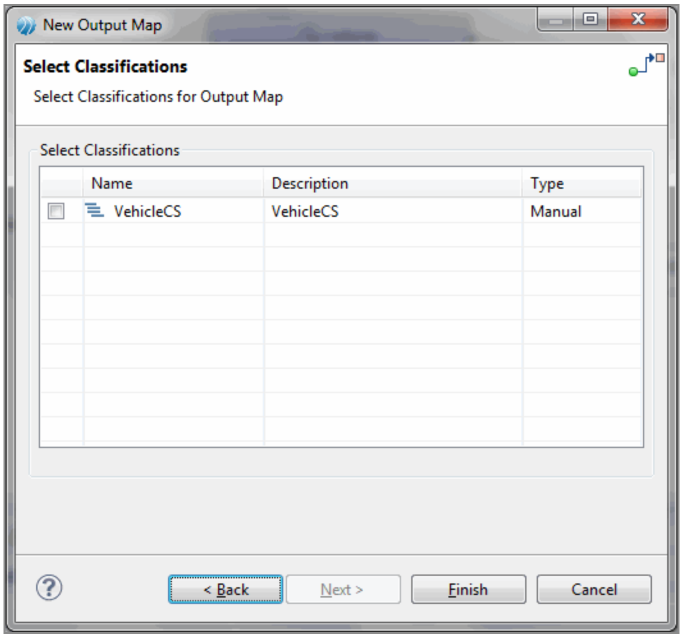
Select Relationships for Output Map

Select Relationships

□	Relationship Name	Name	Sync Format
□	VehicleToCustomer_Fwd	VehicleToCustomer_F...	Sync1
□	InsuranceDetails_Vehicle_Bwd	InsuranceDetails_Vehi...	Sync1
□	VehicleSelf [Vehicle]	VehicleSelf_OutputMap	Sync1

7. To support hierarchical all the immediate relationships of the repository (self relationship, forward and backward relationships) are displayed. You can select different or same sync formats for the relationships. Select the check box corresponding to the relationship name and click **Next**.

8. The Select Classification Scheme for Output map screen is displayed.



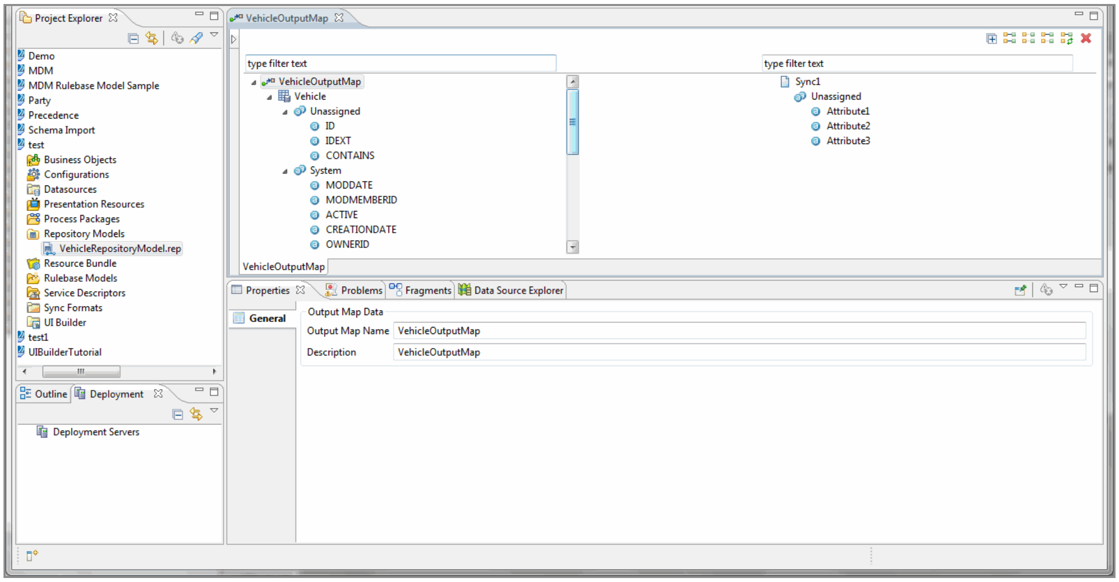
Select Classifications

Select Classifications for Output Map

Select Classifications

	Name	Description	Type
□	VehicleCS	VehicleCS	Manual

9. If you want to include classification scheme, select the check box corresponding to the classification scheme and click **Finish**.
10. The output map is created and displayed in the Output Map editor. If the output map is created by selecting forward, reverse, or self relationships then forward, reverse, relationship output maps are created in separate tabs.



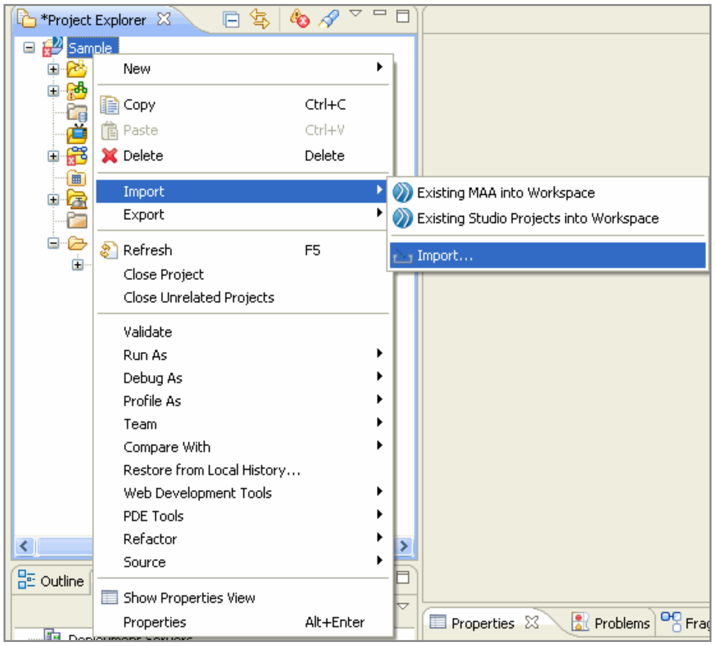
Output Map Name: VehicleOutputMap

Description: VehicleOutputMap

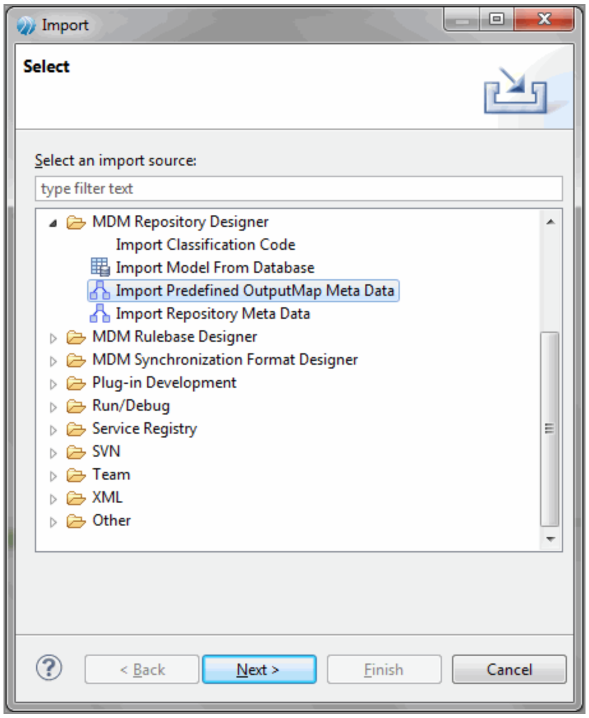
Importing Predefined Output Map

Procedure

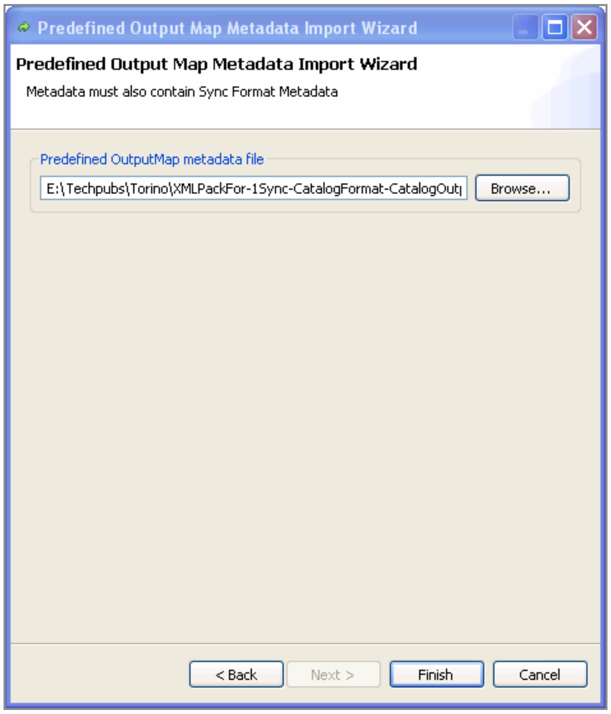
1. In the Project Explorer, right click and select **Import-- > Import.**



2. The import wizard is displayed. Select **Import Predefined Outputmap Meta Data** from **MDM Repository Designer** and click **Next**.



3. The Predefined output map metadata import wizard is displayed. Click on **Browse** and select the predefined output metadata file and click **Finish**.



Predefined Output Map Metadata Import Wizard

Predefined Output Map Metadata Import Wizard

Metadata must also contain Sync Format Metadata

Predefined OutputMap metadata file

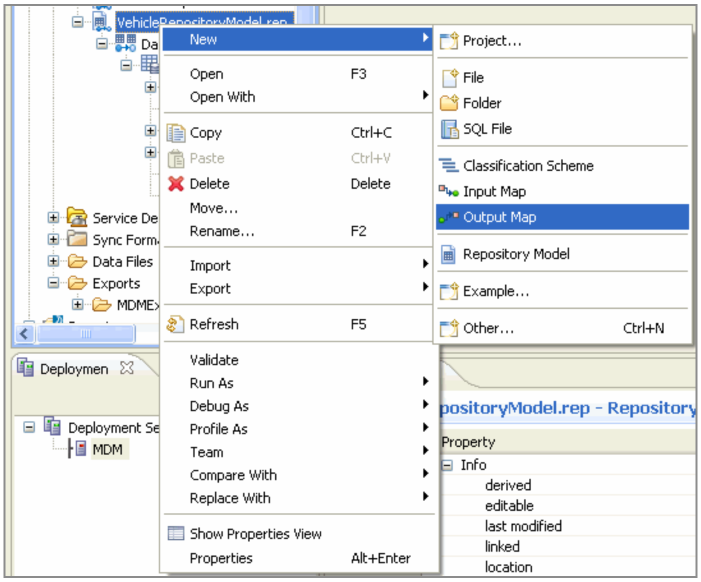
E:\Techpubs\Torino\XMLPackFor-1Sync-CatalogFormat-CatalogOut|

Associating Predefined Output Map with existing Repository

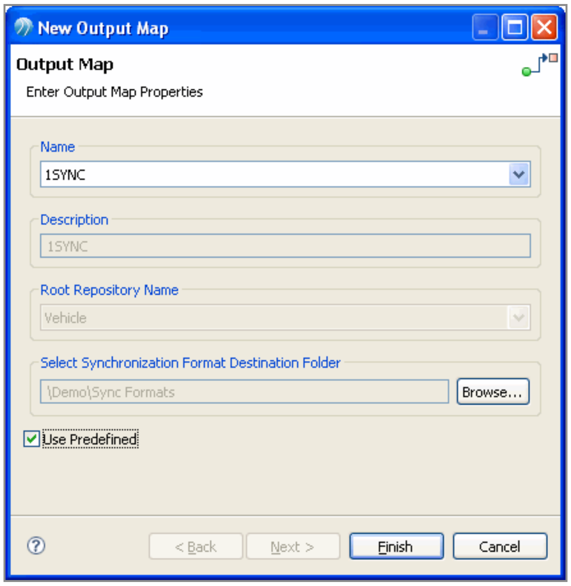
To associate the predefined output map with existing repository, the predefined out map must be imported into TIBCO MDM Studio. The same predefined output maps are populated during association.

Procedure

1. In the Project Explorer under the Repository model file, Right click on Repository on which user want to define the Output Map and select **New-- > Output Map**.



2. The import wizard is displayed. Select the **Use Predefined** check box.



Name: 1SYNC

Description: 1SYNC

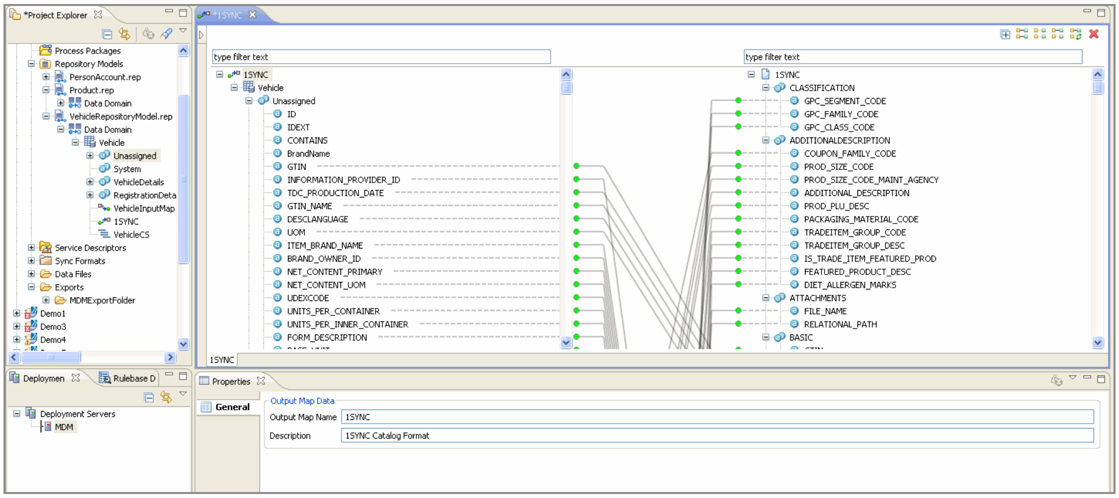
Root Repository Name: Vehicle

Select Synchronization Format Destination Folder: \Demo\Sync Formats

☑ Use Predefined

3. The predefined output map is populated in the **Name** field. Select the appropriate predefined output map and click **Finish**.

4. The Output map is associated and displayed in the Output Map editor.



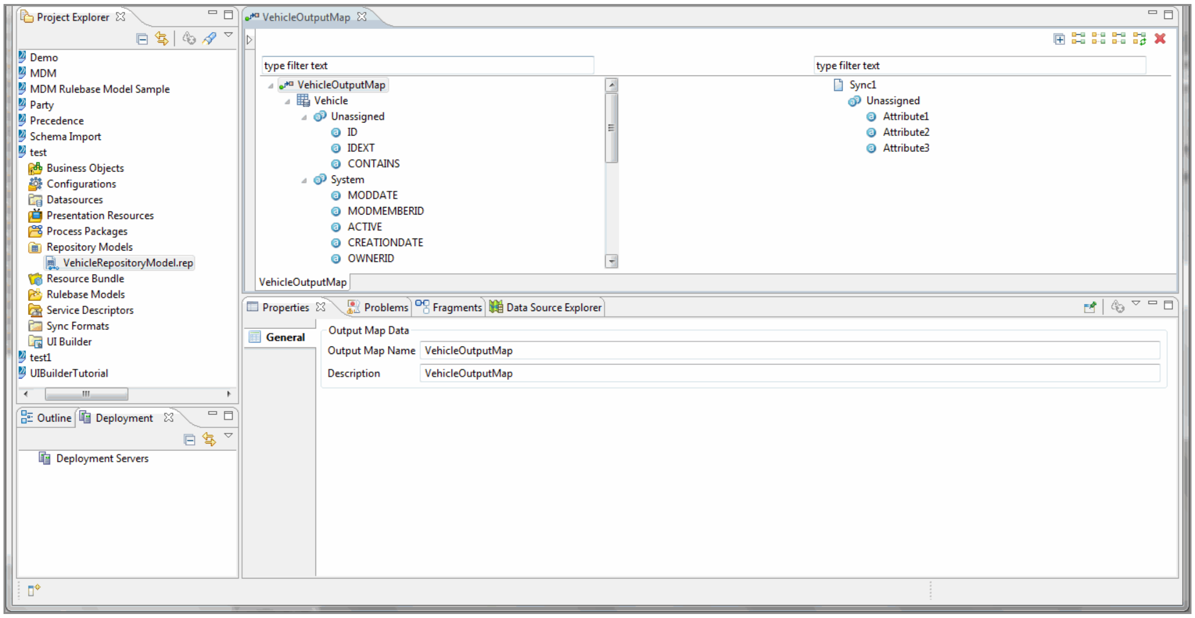
Output Map Data

Output Map Name: 1SYNC

Description: 1SYNC Catalog Format

Output Map Editor

The output map editor displays the newly created output Map. The output map editor contains repository attributes on the left hand side and sync format attributes on the right hand side shown as a tree. The type ahead filter is also provided for quick search. You can add related output map from the editor.



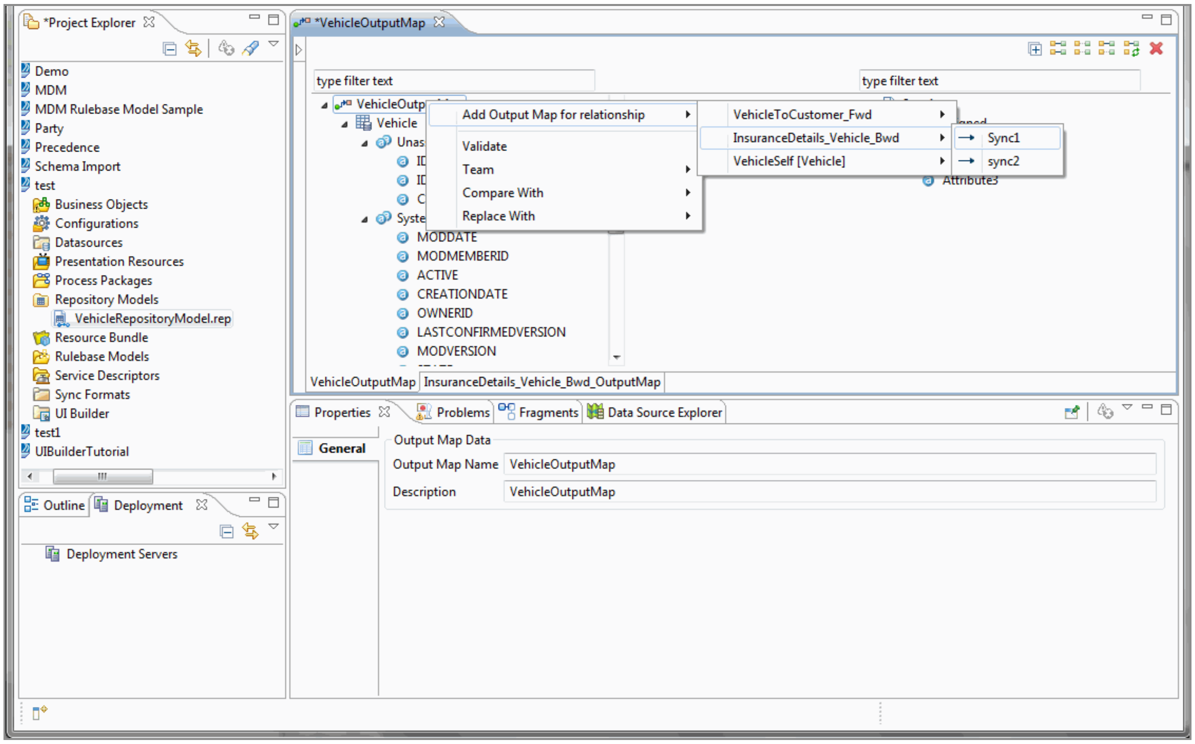
Output Map Data

Output Map Name: VehicleOutputMap

Description: VehicleOutputMap

Creating a Related Output Map

You can create related output map for a forward, reverse, and self relationship from editor. You can also create related output map from the wizard. If you are creating a related output map on relationship and if the relationship has relationship attributes then for that relationship attributes, relationship output map gets created.

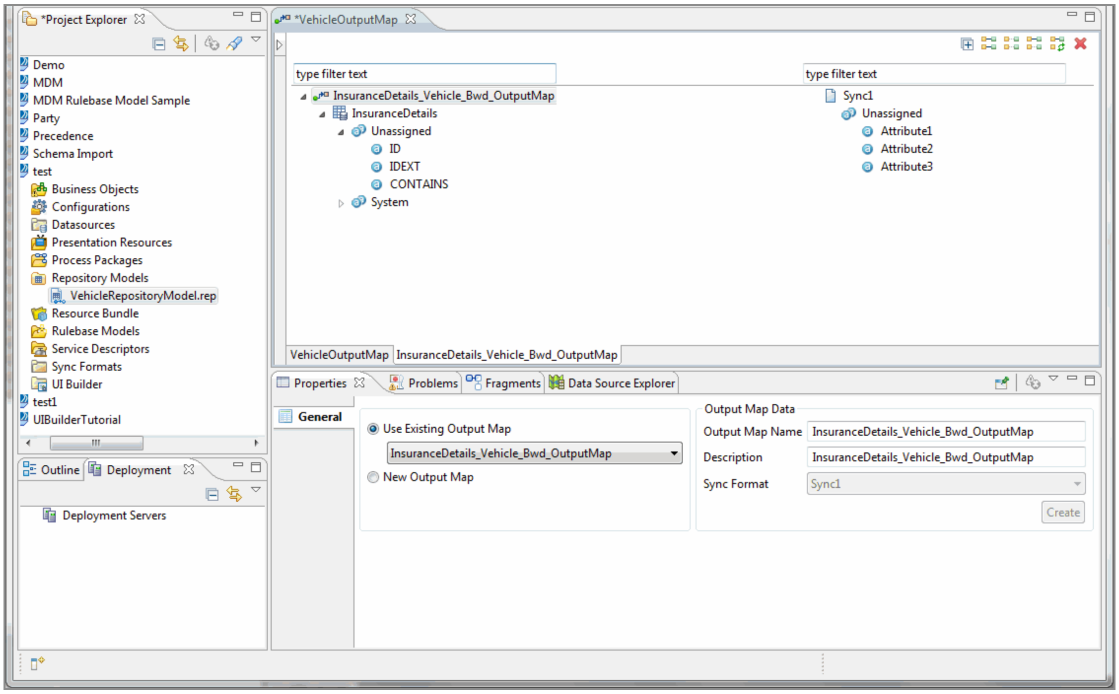


Output Map Name: VehicleOutputMap
Description: VehicleOutputMap

Procedure

1. Right-click on the parent output map, point to **Add Output Map for relationship.** The relationships available for parent output map are shown. When you select relationship, available synchronization formats are displayed to associate with related output map. For example, select the reverse relationship **InsuranceDetails_ Vehicle_Bwd**.

2. The output map for the reverse relationship is displayed.



● Use Existing Output Map: InsuranceDetails_Vehicle_Bwd_OutputMap

○ New Output Map

Output Map Data

Output Map Name: InsuranceDetails_Vehicle_Bwd_OutputMap

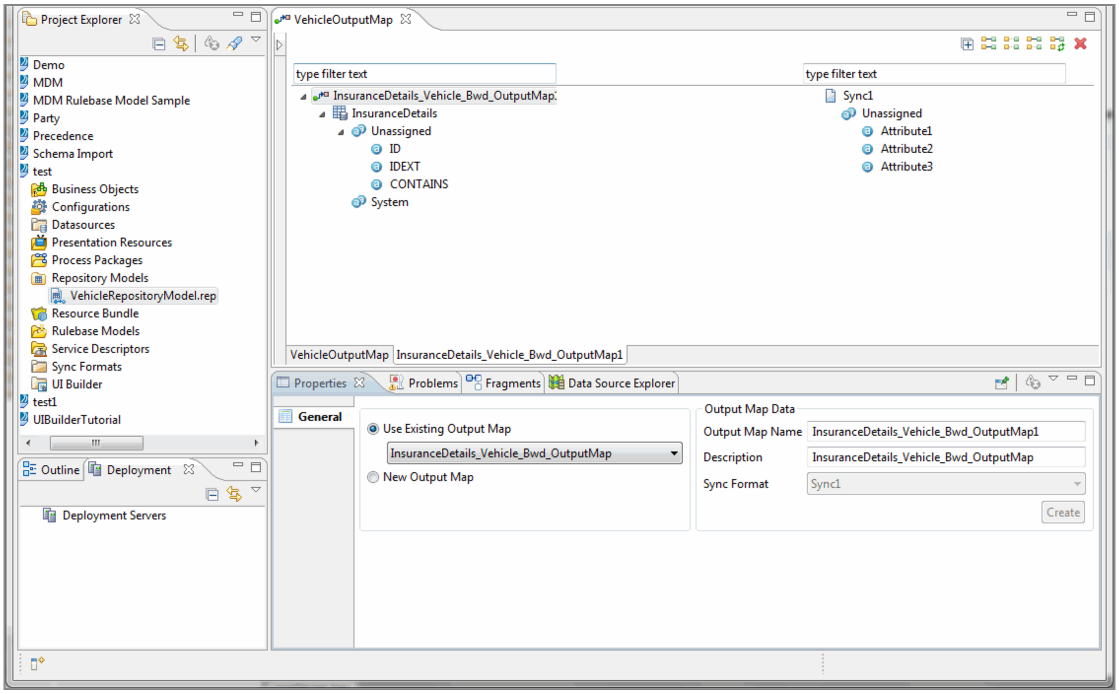
Description: InsuranceDetails_Vehicle_Bwd_OutputMap

Sync Format: Sync1

3. You can create a related output map using an existing output map or create new output map to link with the parent output map. If you have existing output maps in the related repository, select the **Use Existing Output Map** option. When you select the **Use Existing Output Map** option, the drop-down list displays the output maps available in the related repository. Select **Create New Output Map** option to create a new output map for the related repository. The new output map name and description is automatically generated with numeric suffix. Select the synchronization format and click **Create**.

> **Note:** To define a relationship between two output maps of any two repositories, there must exist a relationship definition between those two repositories.

4. You can rename the output map from the properties section. For example, you can rename the related output to **InsuranceDetails_Vehicle_Bwd_OutputMap1**



● Use Existing Output Map: InsuranceDetails_Vehicle_Bwd_OutputMap

○ New Output Map

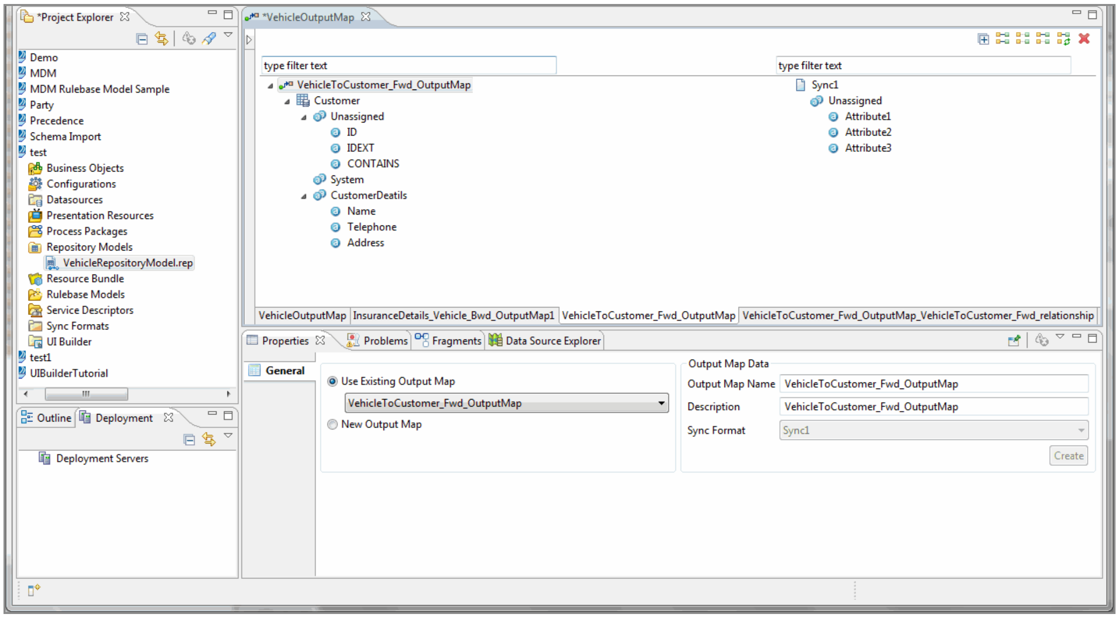
Output Map Data

Output Map Name: InsuranceDetails_Vehicle_Bwd_OutputMap1

Description: InsuranceDetails_Vehicle_Bwd_OutputMap

Sync Format: Sync1

5. Since the related output map on a relationship has relationship attributes, then for the relationship attributes, a relationship output map gets created.



● Use Existing Output Map: VehicleToCustomer_Fwd_OutputMap

○ New Output Map

Output Map Data

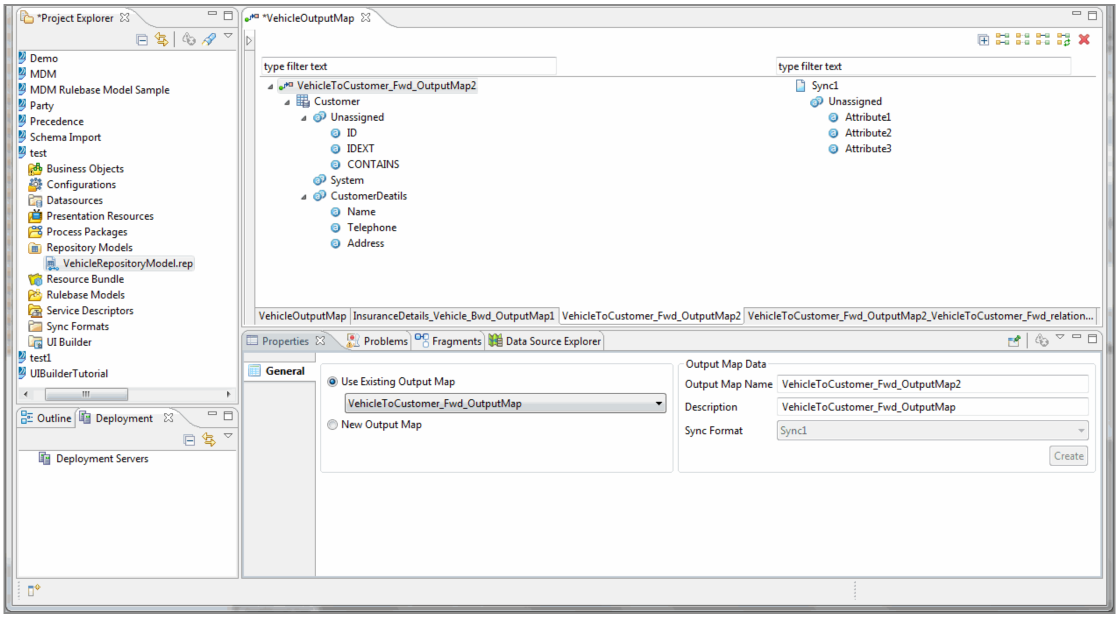
Output Map Name: VehicleToCustomer_Fwd_OutputMap

Description: VehicleToCustomer_Fwd_OutputMap

Sync Format: Sync1

6. If you rename the related output map, then the relationship output map is also renamed. For example, you can rename the related output to **VehicleToCustomer_Fwd_OutputMap2** then the relationship output map also gets renamed to

VehicleToCustomer_Fwd_OutputMap2_VehicleToCustomer_Fwd_relationship.



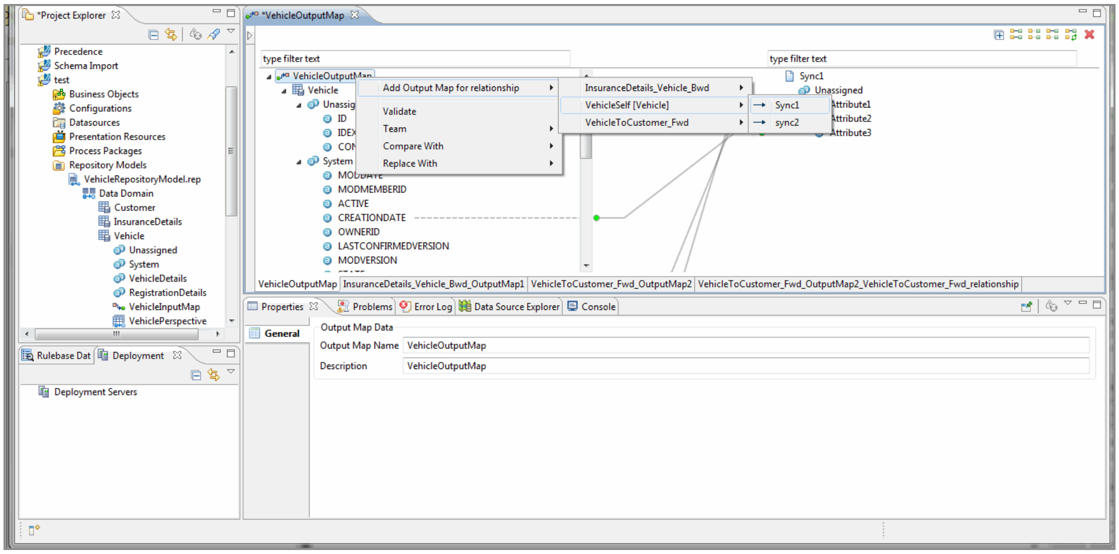
● Use Existing Output Map: VehicleToCustomer_Fwd_OutputMap

Output Map Name: VehicleToCustomer_Fwd_OutputMap2

Description: VehicleToCustomer_Fwd_OutputMap

Sync Format: Sync1

7. Similarly you can also create a related output for a self relationship.



Output Map Name: VehicleOutputMap

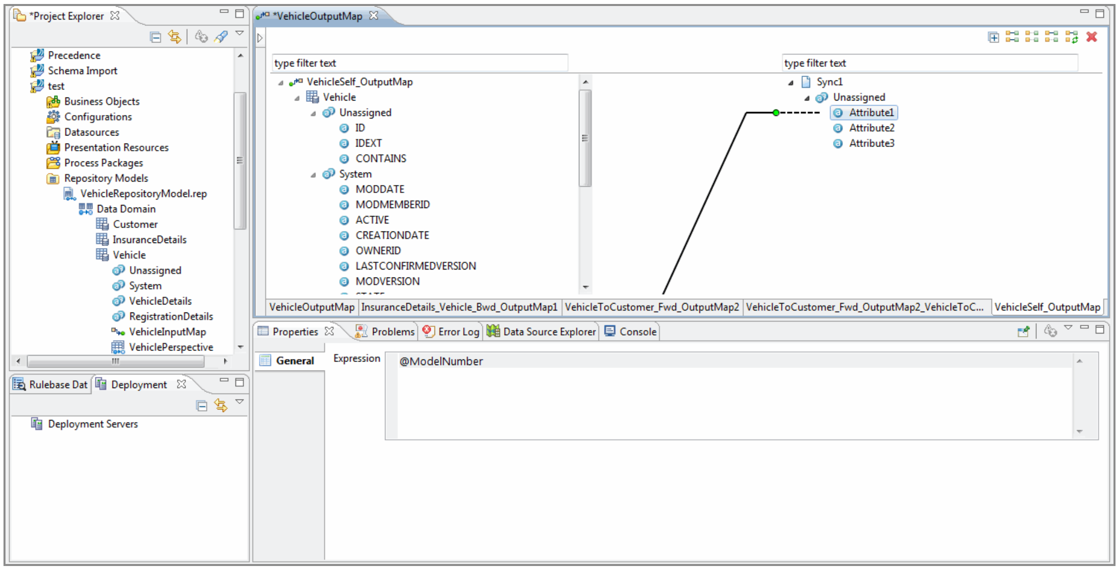
Description: VehicleOutputMap

Defining Output Map Mapping

For Repository Attributes

Procedure

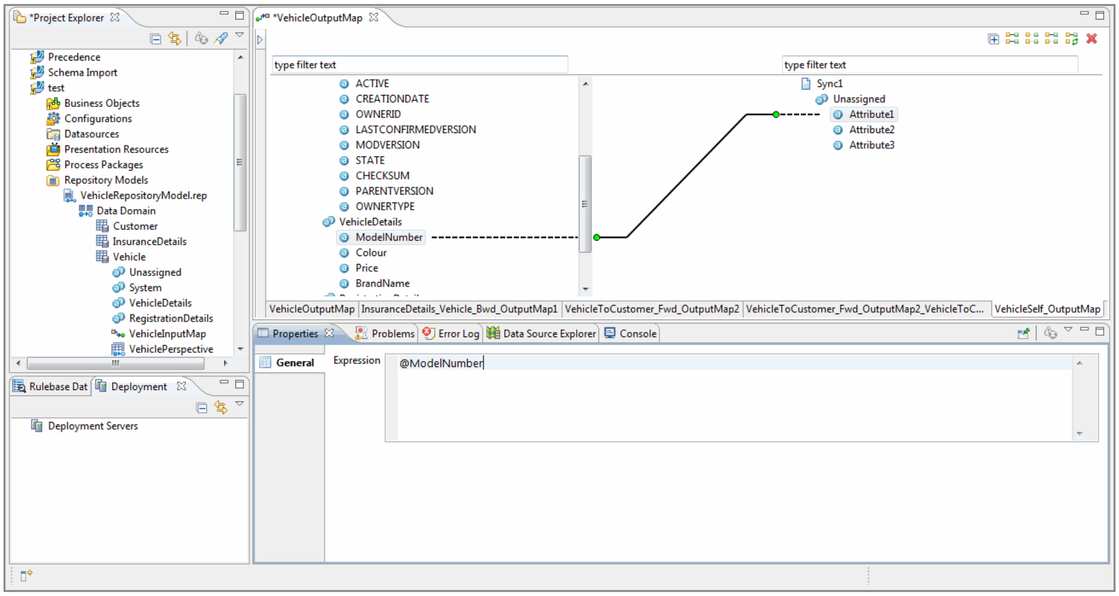
1. To manually map the output map, Drag and drop the Repository Attribute (Output Map tree on the left side of the editor) to Synchronization format attributes (Sync format tree on the right side of the editor).



Expression @ModelNumber

2. To automatically map the output map, click .

3. To enter the source expression for the Repository attribute, click on the mapping. The text expression editor will appear in the property section.



Expression: @ModelNumber

4. Enter a valid SQL expression in the **Expression** text field. The rules applicable to an expression entered are:

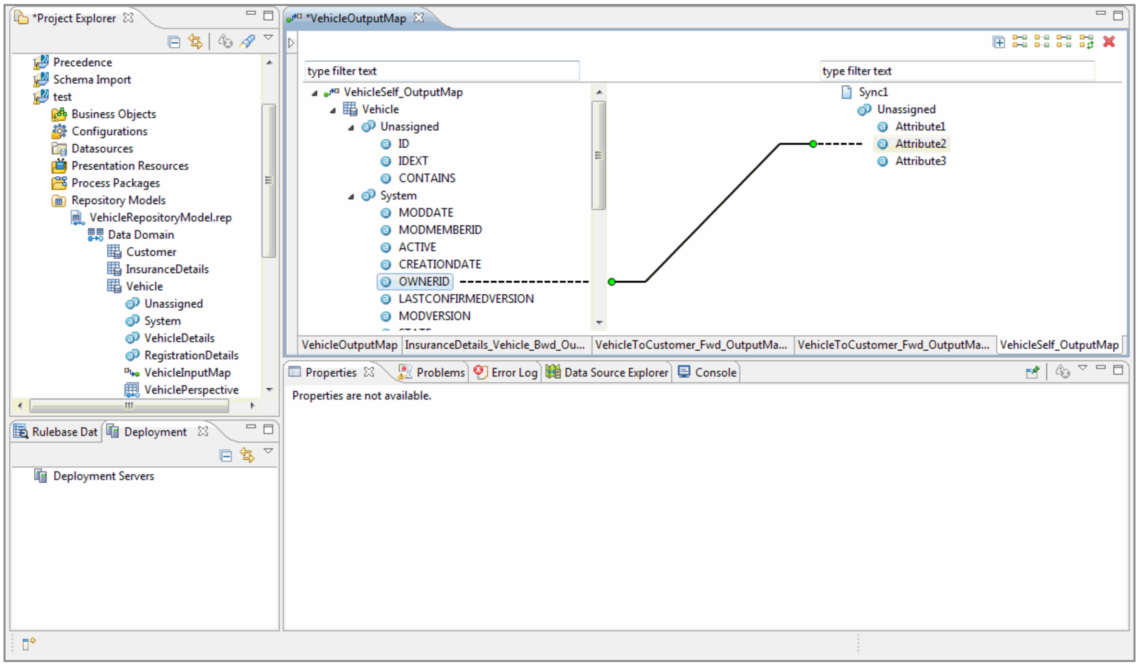
   a. All expressions entered in the expression text field must be valid SQL expressions.

   b. If you include a column name from an SQL table, ensure that the name, entered exists and is as specified in the SQL table.

   c. For constant value mapping, enter integers directly, but embed String values in single quotes (’<string’). For example, For Integer, CASE WHEN price > 50 THEN 50*3 ELSE price END. For String, CASE WHEN state = 'CONFIRMED' THEN 'CONF' ELSE state END.

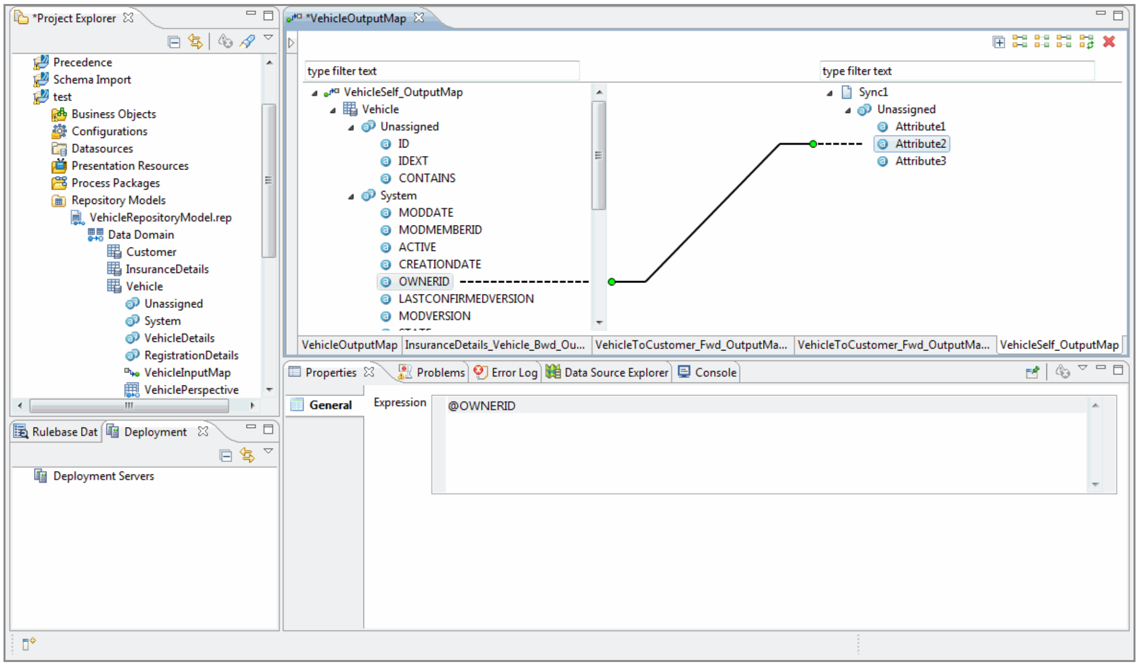
For Predefined System Attributes

To map predefined attributes, drag and drop the predefined attribute from System attribute group displayed on left side tree to synchronization format attributes (Sync format tree on the right side of the editor).



Properties are not available.

To enter the source expression for the predefined system attribute, click on the mapping. The text expression editor will appear in the property section.

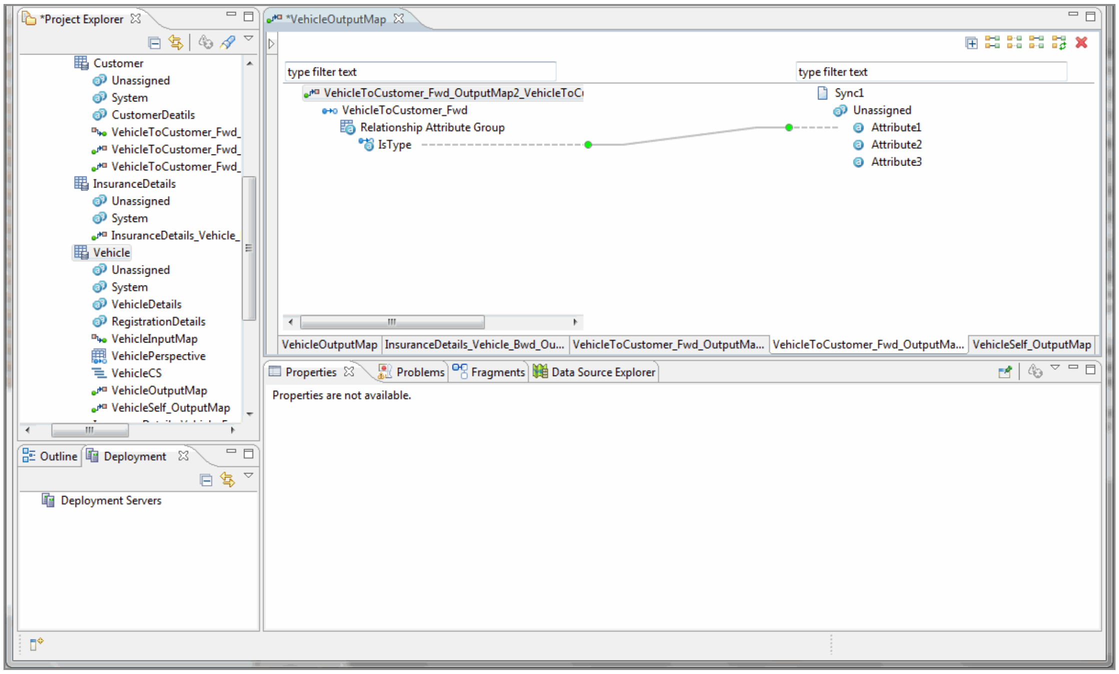


Expression: @OWNERID

For Relationship Attributes

Procedure

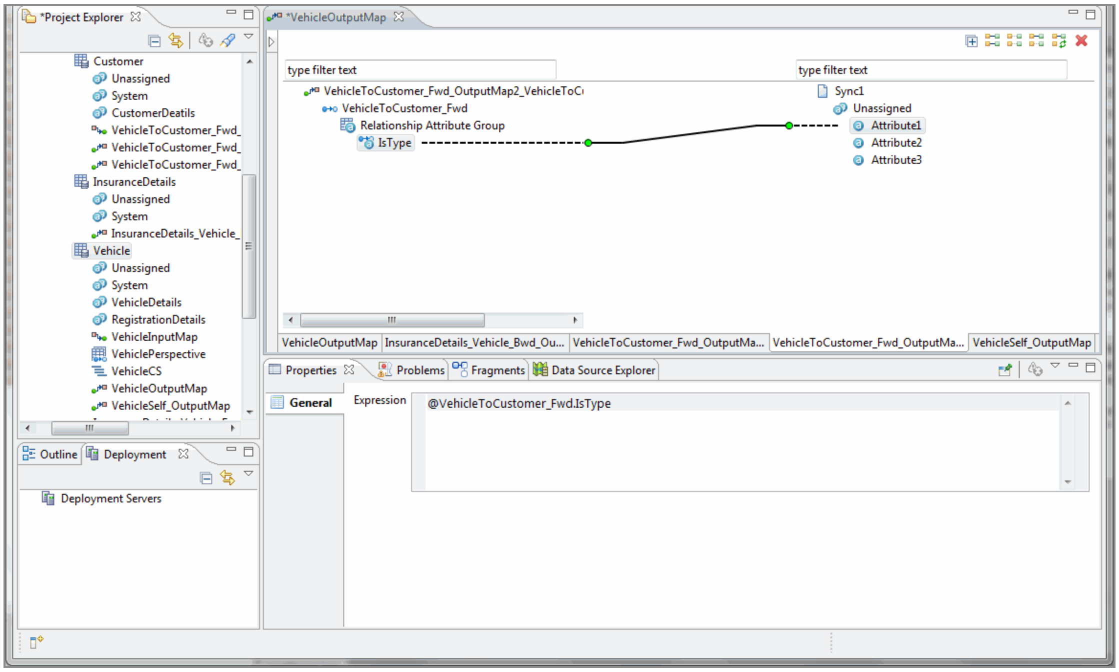
1. To manually map the relationship attribute, open relationship output map tab and drag and drop the relationship attribute (on the left side of the editor) to Synchronization format attributes (on the right side of the editor).



Properties are not available.

2. To automatically map the output map, click .

3. To enter the source expression for the Relationship attribute, click on the mapping. The text expression editor will appear in the property section.

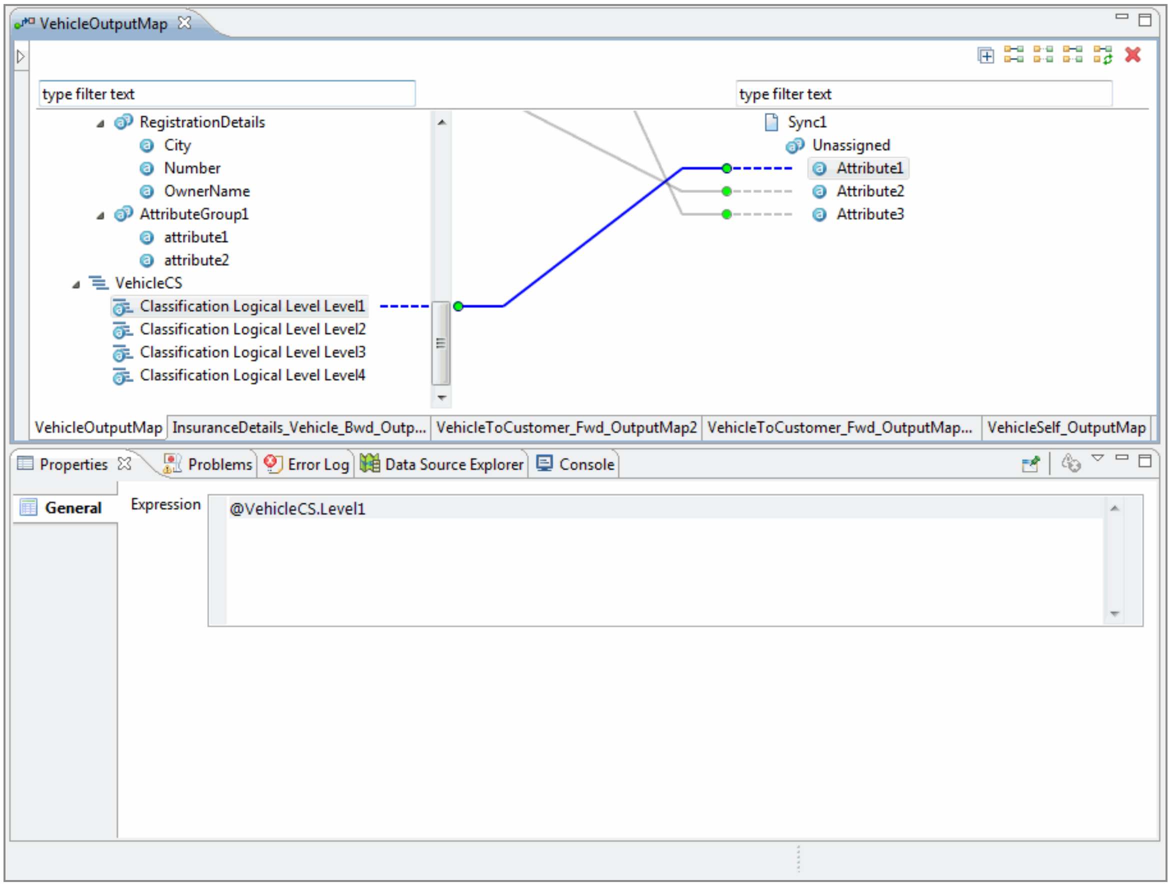


Expression: @VehicleToCustomer_Fwd.IsType

4. Enter a valid SQL expression in the **Expression** text field.

For Classification Scheme

To map classification scheme attributes, drag and drop the classification scheme attribute on the left side of the editor to synchronization format attributes on the right side of the editor.

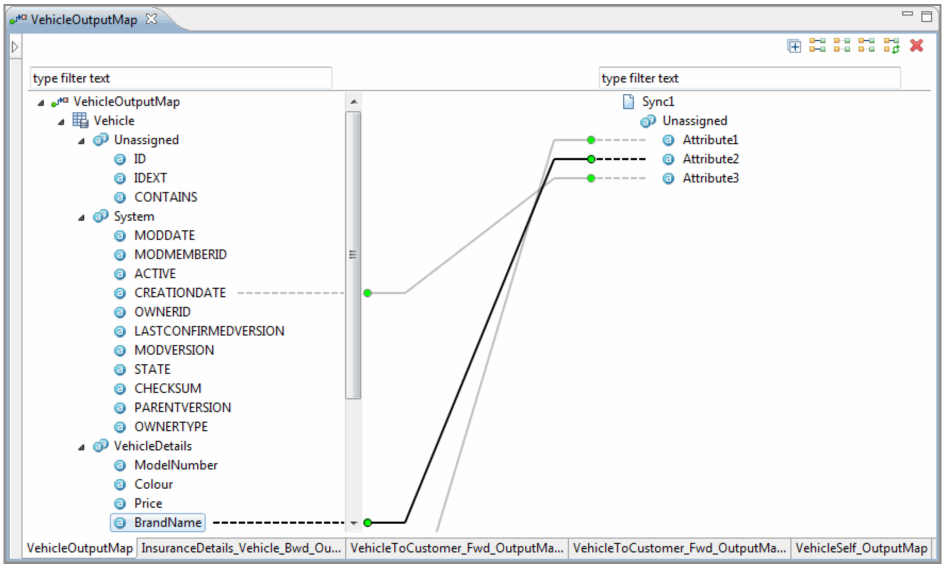


Expression @VehicleCS.Level1

View Mapped Unmapped Mappings

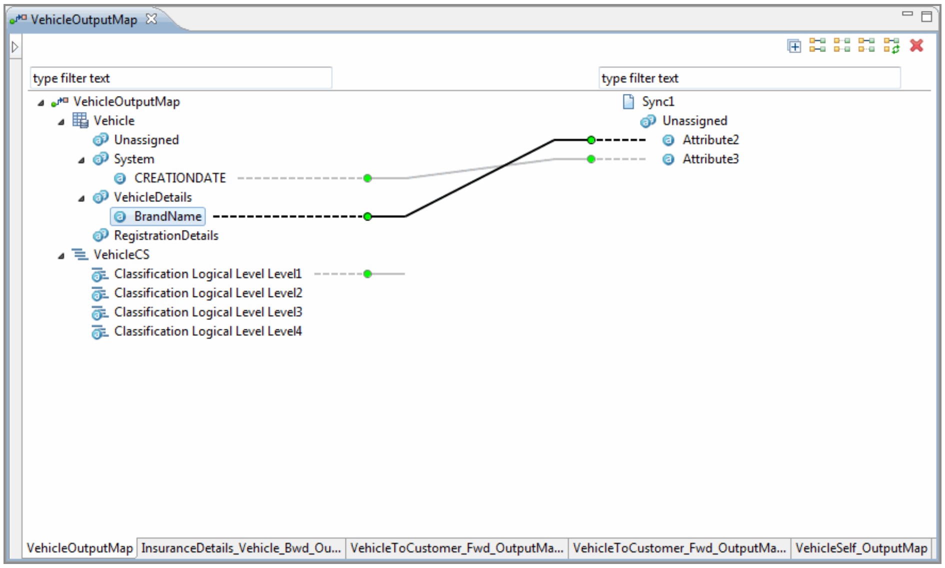
To view all mapping (both mapped and unmapped attributes) click .

The attributes values are expanded and the mapping is displayed.



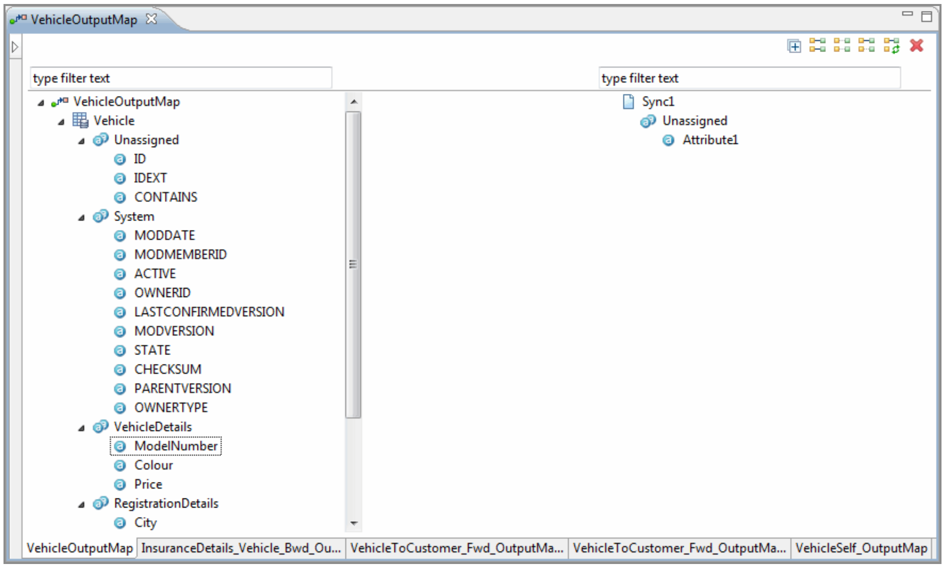
To view only mapped attributes, click .

Only the mapped attributes are displayed.


VehicleOutputMap
type filter text
type filter text
VehicleOutputMap
Vehicle
Unassigned
System
CREATIONDATE
VehicleDetails
BrandName
RegistrationDetails
VehicleCS
Classification Logical Level Level1
Classification Logical Level Level2
Classification Logical Level Level3
Classification Logical Level Level4
Sync1
Unassigned
Attribute2
Attribute3
VehicleOutputMap
InsuranceDetails_Vehicle_Bwd_Ou...
VehicleToCustomer_Fwd_OutputMa...
VehicleToCustomer_Fwd_OutputMa...
VehicleSelf_OutputMap

To view unmapped attributes, click .

All the unmapped attributes are displayed.



Deleting Mappings

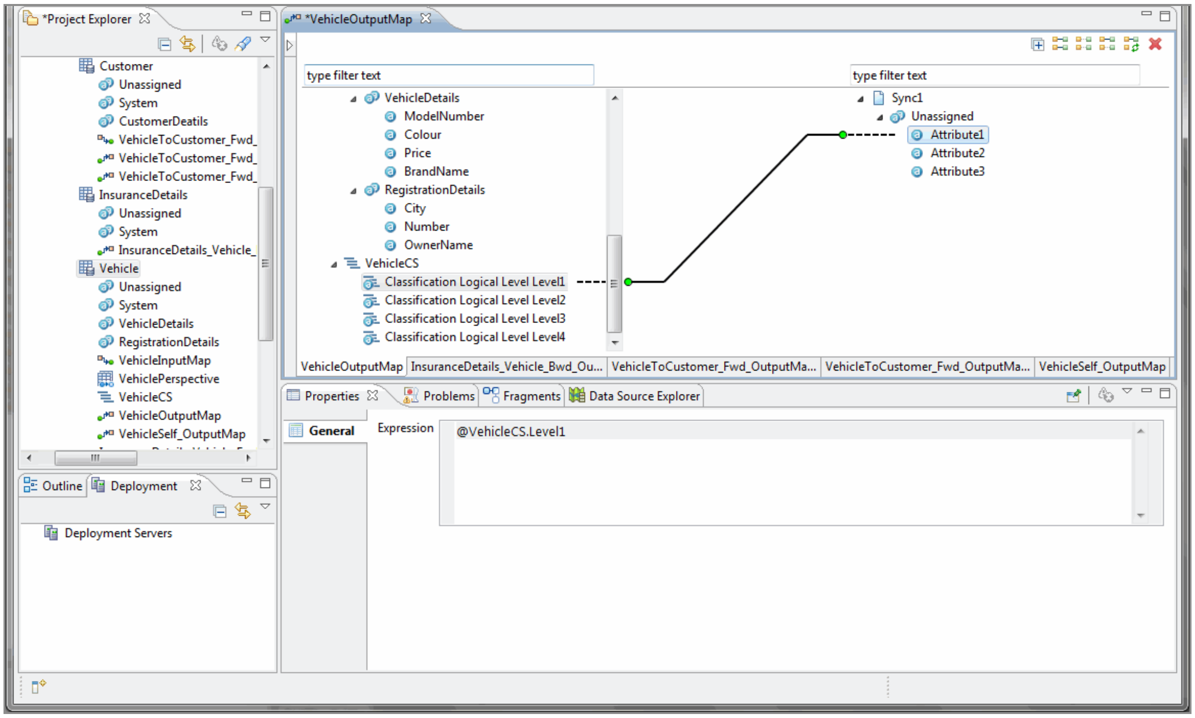
To delete mapping from the existing output map.

Procedure

1. From the Output Map Editor, select the mapping which you want to delete. Right click on the mapping and click ✖ .

View Category Hierarchy

To view the category Hierarchy, click ▷ and expand. The Category hierarchy along with the belonging attributes are displayed..



Expression: @VehicleCS.Level1

The Category Specific Attributes by default are hidden in the output map editor. Click the check box corresponding to the Category Specific Attribute to control the visibility of a particular Category Specific Attribute. To search for the Category Specific Attributes, specify the attribute in the search field. When a Category Attribute is selected, the parent category is also selected till the root level and all the inherited attributes are also selected. On selection, the Category Specify Attribute is displayed in the output map editor.

Deleting Output Maps

If you delete the related output map having related relationship attributes, all associated relationships with other output maps are deleted, but the parent output map in the related repositories remains intact.

Similarly, when a repository is deleted, all the associated relationships, input maps, and output maps across repositories are deleted.

Procedure

1. From the Output Map Editor, select the output Map from which you want to delete and click ✖ on the top right corner of the editor or right click on the output map in the project explorer and click

✖ .

Result

Note: If repository attributes are either deleted from the repository model or moved from one attribute group to another, then the mapping between synchronization format attributes and repository attributes will be deleted from the output map. Establish the mapping once again if the attributes are moved from one group to another.

Working with PatternsIndexer Configuration Editor

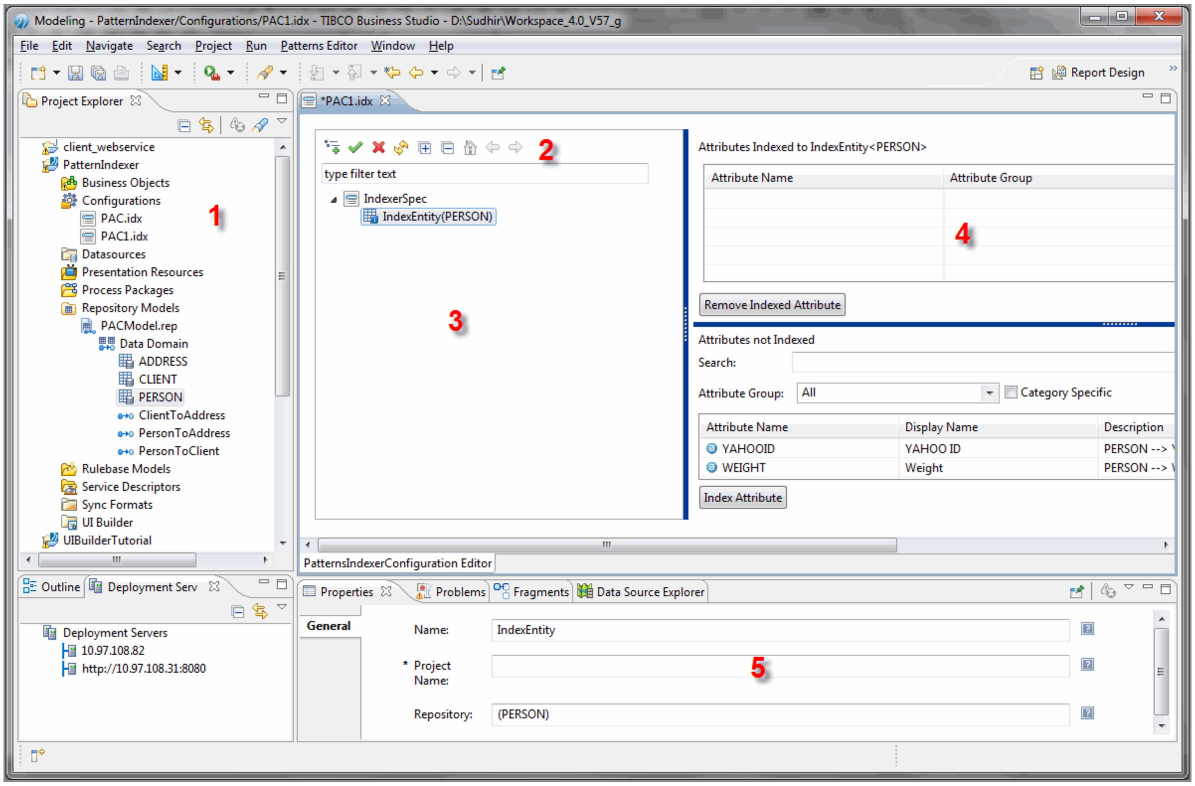
This chapter introduces you to the concept of PatternsIndexer Configuration in TIBCO MDM Studio and explains how to create and deploy the PatternsIndexer Configuration using PatternsIndexer Configuration Editor.

Overview

The PatternsIndexer Configuration allows you to define a new editor which is more intuitive and takes advantage of the repository metadata to define attributes for Text-indexing. The PatternsIndexer Editor allows to define the IndexEntity hierarchy. Some of the features supported by PatternsIndexer Configuration Editor are as follows:

- Allows adding, editing of property section of the IndexEntity.
- Allows adding related repositories for defining join entities.
- Allows selecting more than one repository attributes for Text-indexing.
- Allows indexing and unindexing of attributes to IndexEntity and TargetRepository
- Allows filtering and searching within the tree based on the IndexEntity name
- Allows filtering the attributes that are not Indexed using the attribute group, name and category specific flag.

PatternIndexer Configuration Interface



Attribute Name	Display Name	Description
YAHOOID	YAHOO ID	PERSON -->
WEIGHT	Weight	PERSON -->

Name: IndexEntity

* Project Name:

Repository: (PERSON)

Project Explorer: The project explorer displays all the folders available in the project. The folder like the datasource, repository model, project's configuration, rulebase models and so on.

IndexerSpec Tree Control Toolbar: The IndexerSpec tree control toolbar displays the toolbar icons for adding index entity, validating, deleting, re-ordering, refreshing, expanding and collapsing the tree outline.

Tree Controls	
Editing Controls	
+	Create a new IndexEntity.
✓	Validate the IndexEntity.

❌	Delete a IndexEntity.
Navigating Controls	
⟳	Refresh the IndexEntity.
⊞	Expand to view all the IndexEntites.
⊟	Collapse all the IndexEntites.
🏠	Home IndexEntity. If you are inside a child node IndexEntity and want to return the main IndexEntity.
⇦	Back IndexEntity. If you are inside a child node IndexEntity click the icon to return back.
⇨ ⇨	GoInto IndexEntity if you want view the child node IndexEntity.

IndexerSpec Tree Control: IndexerSpec Tree control shows the hierarchy of IndexEntities created for the various repositories.

Attributes Selection Area: The attributes selection area helps is adding and removing attributes for indexing.

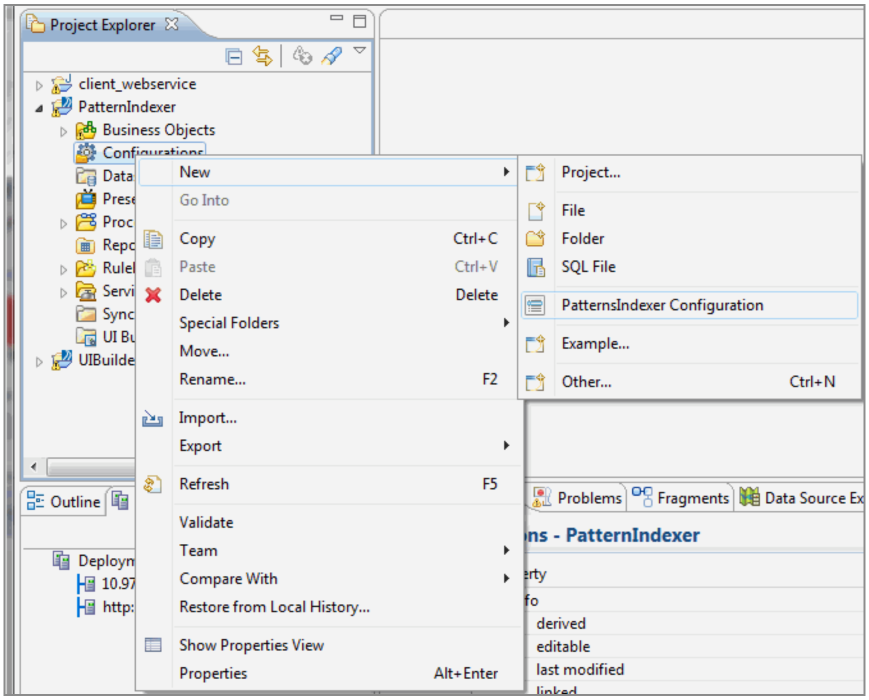
PatternIndexer Properties: The property section displays the properties of the IndexEntities.

Creating a PatternsIndexer Configuration

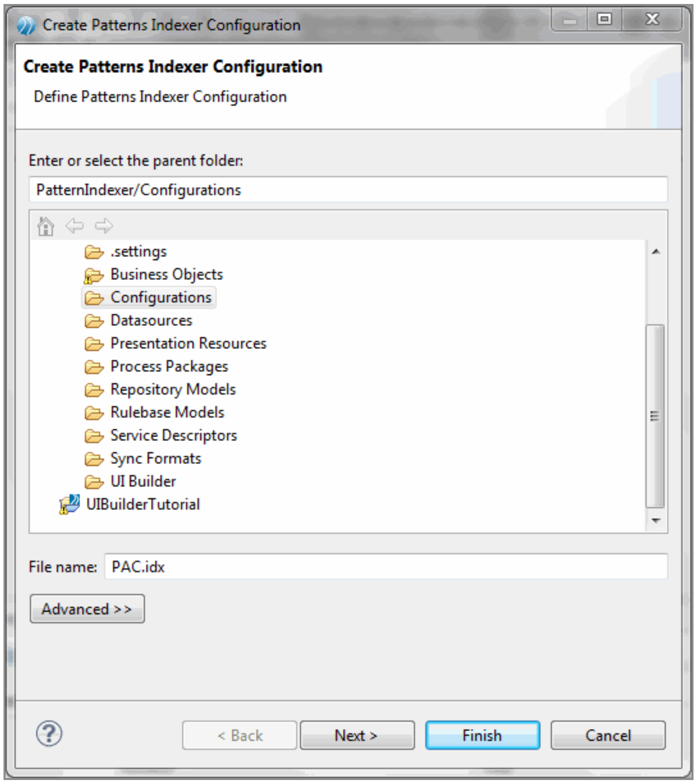
The PatternsIndexer Configuration is created using the PatternsIndexer Configuration Wizard.

Procedure

1. Select the Configurations folder and right- click on **New > PatternsIndexerConfiguration**.



2. The **Create Patterns Indexer Configuration** wizard is displayed.



Create Patterns Indexer Configuration

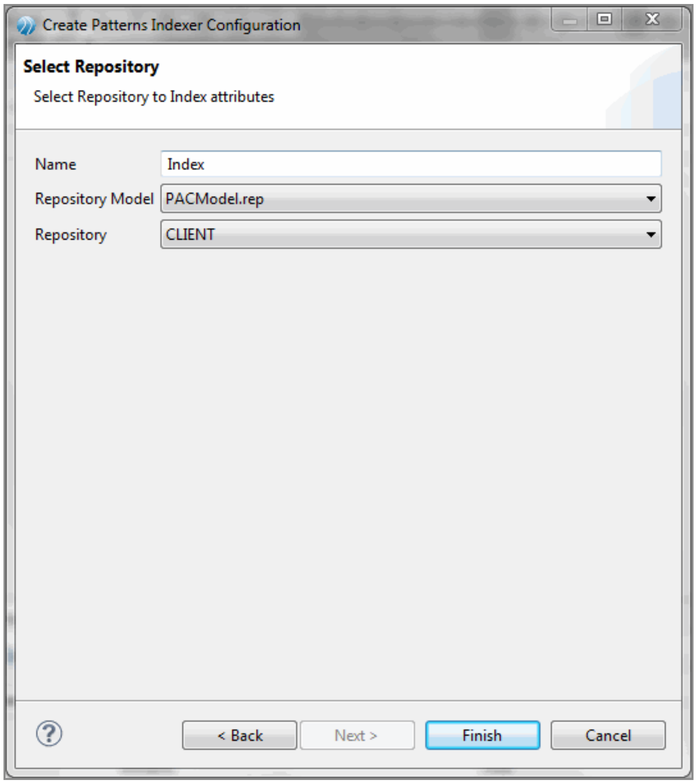
Create Patterns Indexer Configuration

Define Patterns Indexer Configuration

Enter or select the parent folder: PatternIndexer/Configurations

File name: PAC.idx

3. Enter the patternsIndexer file name in **File name** field and click **Next**.
4. The **Select Repository** wizard for selecting the repository to index attributes is displayed.


Create Patterns Indexer Configuration
Select Repository
Select Repository to Index attributes
< Back
Next >
Finish
Cancel

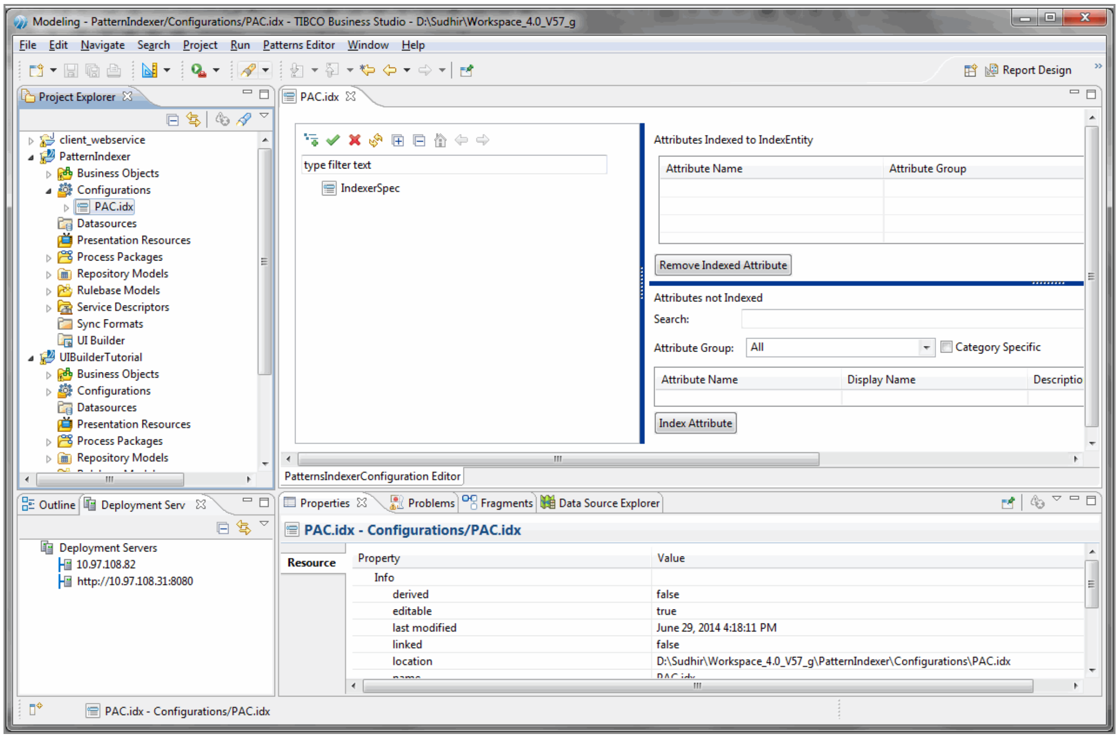
Name: Index

Repository Model: PACModel.rep

Repository: CLIENT

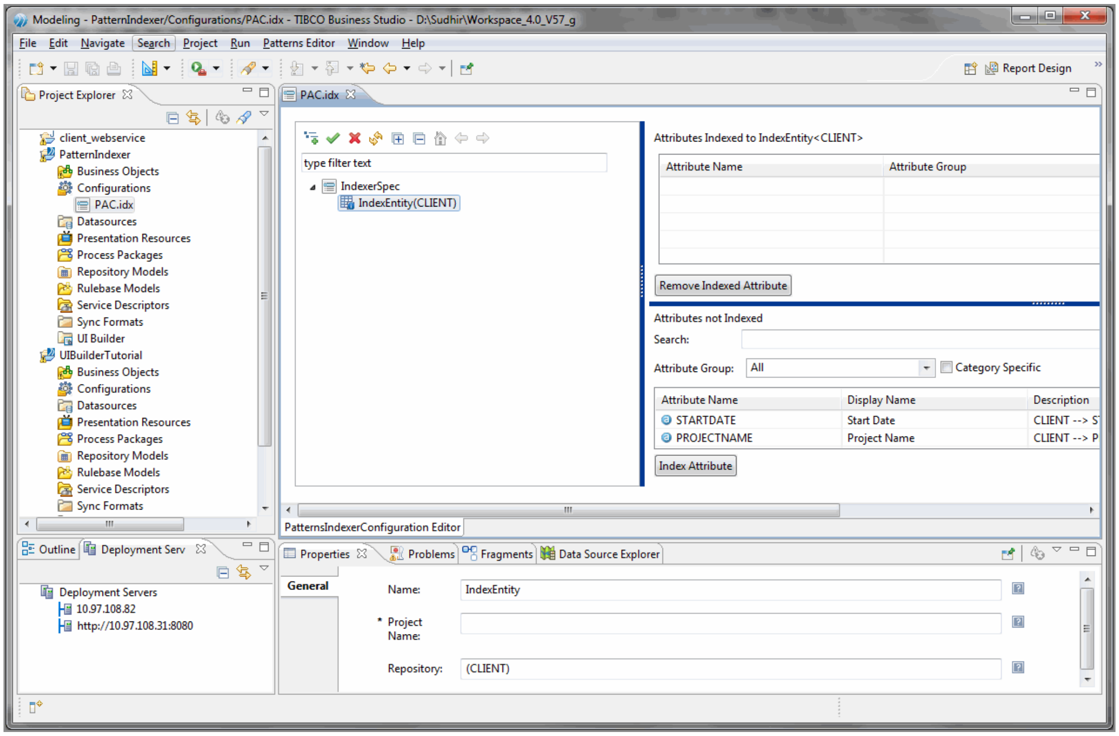
Field	Description
Name	Name of the PatternsIndexer.
Repository Model	Select the repository model from the drop-down list. Ensure that you have repository model or else create a repository model.
Repository	The list of all the repository available in the repository model is displayed. Select the repository on which you want to create the index.

5. Enter the IndexEntity name in the **Name** field.
6. Click **Finish**.
7. The **PatternsIndexer Configuration Editor** is displayed.



Property	Value
Info	
derived	false
editable	true
last modified	June 29, 2014 4:18:11 PM
linked	false
location	D:\Sudhir\Workspace_4.0_V57_g\PatternIndexer\Configurations\PAC.idx

8. Expand the **IndexerSpec** node. If IndexEntity is created without selecting a repository, you can drag and drop a repository on to index entity directly. If no IndexEnity is present you can create by clicking Create IndexEntity button on the tree control toolbar.



Attribute Name	Display Name	Description
STARTDATE	Start Date	CLIENT --> S
PROJECTNAME	Project Name	CLIENT --> P

Name: IndexEntity

* Project Name:

Repository: (CLIENT)

9. The newly created IndexEntity is displayed. The repository on which the indexer is created is displayed inside the bracket. For example, CLIENT. Using the PatternsIndexer Configuration Editor you can create the indexing attributes. For more information on the PatternsIndexer Configuration editor refer to PatternsIndexer Configuration Editor.

10. If there are multiple relationships selected for indexing, select the **Normalize** checkbox in the **Properties** section. On selecting this option, the TIBCO Patterns-Engine uses patterns join for searching the `term` across repositories.

PatternsIndexer Configuration Editor

The PatternsIndexer Configuration is divided in three section:

- Left hand side is the tree control to define IndexEntity hierarchy.
- Upper right hand side to shows the Indexed attributes selected for text indexing.
- Lower right hand side shows the UnIndexed attributes of the repository attributes.

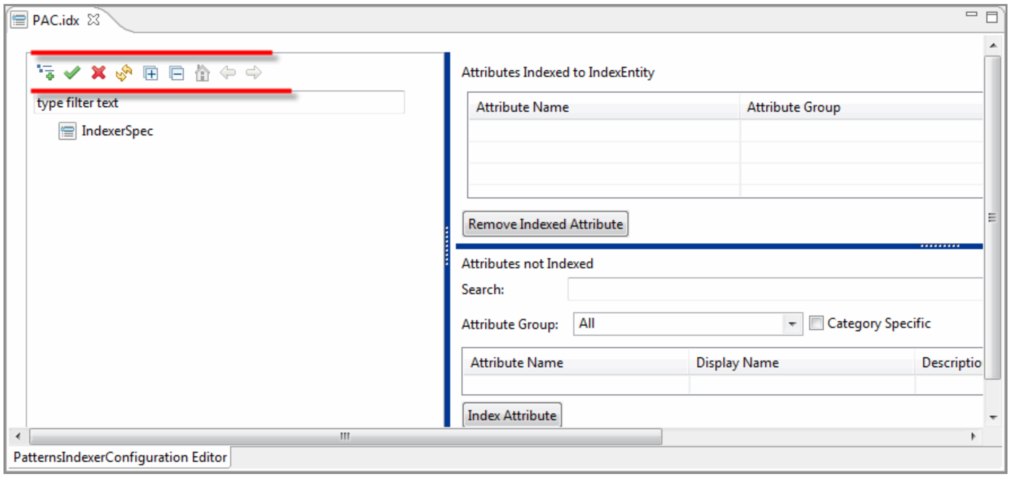
The PatternsIndexer Configuration Editor allows:

- Adding IndexEntity.

- Adding related repositories for defining join entities.
- Editing of IndexEntity name from the Property section.
- Indexing and un-indexing of attributes to IndexEntity and target repository.
- Filtering and searching within the IndexEntity tree based on the name,
- Filtering of attributes that are not Indexed using attribute group, name, and Category Specific Attributes flag.

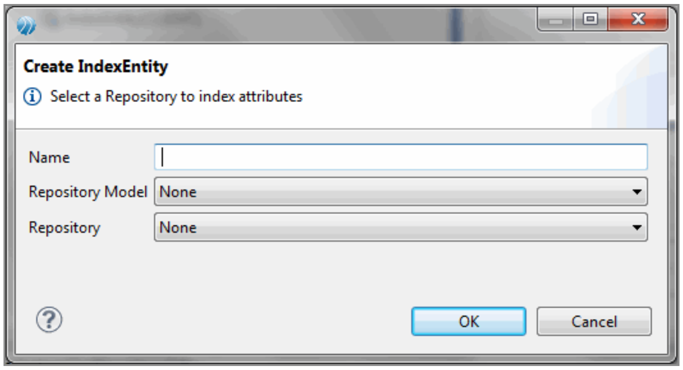
Creating IndexEntity

Using PatternsIndexer Configuration tree controls you can create new IndexEntities.



Procedure

1. Click to add a IndexEntity. The Create IndexEntity wizard is displayed.



Create IndexEntity

Select a Repository to index attributes

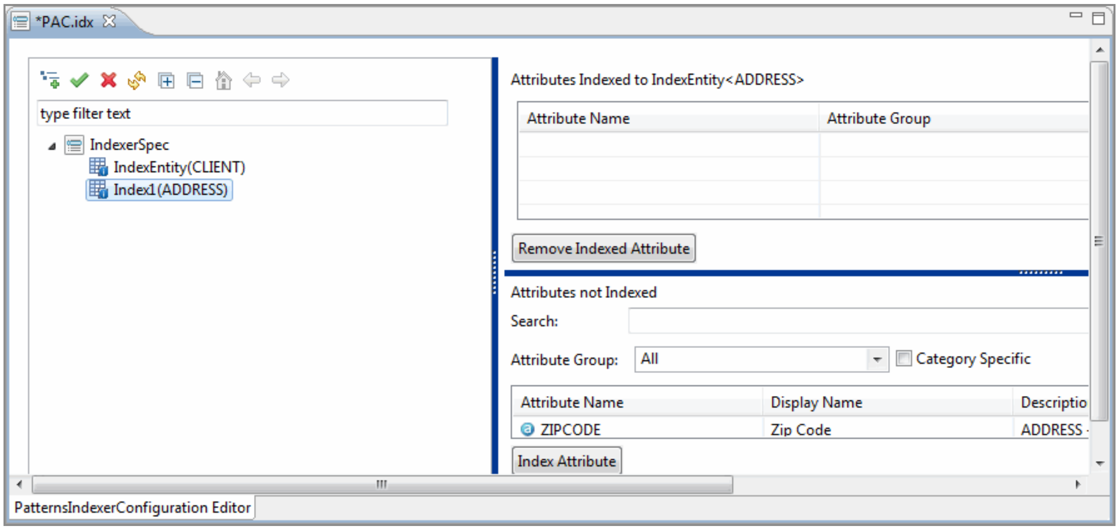
Name

Repository Model: None

Repository: None

2. Enter the IndexEntity name in the **Name** field. Select the repository model from the **Repository Model** drop-down list. Based on the repository model, the available repositories will be displayed in the Repository drop-down list. Select the appropriate repository. Repository selection and IndexEntity name are optional in the wizard.

3. Click **OK**. The newly created IndexEntity is displayed.



Attributes Indexed to IndexEntity<ADDRESS>

Attribute Name	Attribute Group

Attributes not Indexed

Search:

Attribute Group: All

□ Category Specific

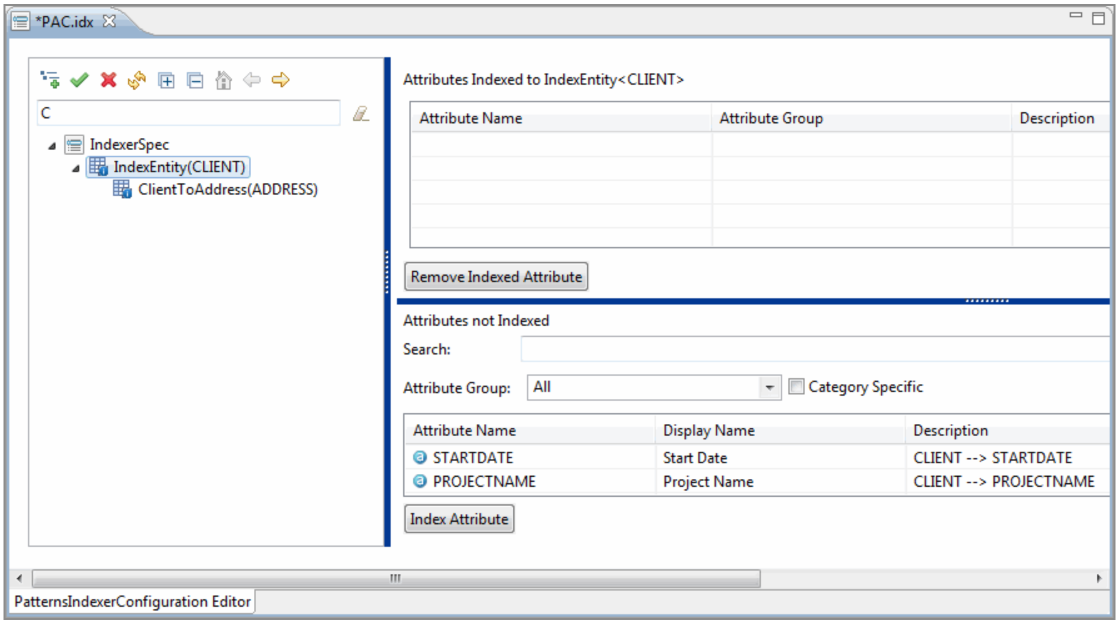
Attribute Name	Display Name	Descriptio
ZIPCODE	Zip Code	ADDRESS -

Defining Indexing Repository Attributes

Using this option you can select the repository attributes for text indexing.

Procedure

1. Select the IndexEntity in the Tree Control view, all the attributes associated with the repository are displayed in **Attributes not Indexed** section.



Attributes Indexed to IndexEntity<CLIENT>

Attribute Name	Attribute Group	Description

Attributes not Indexed

Search:

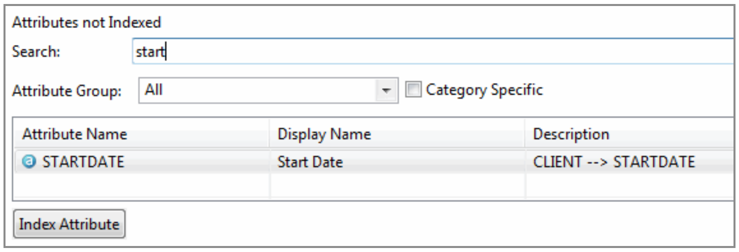
Attribute Group: All □ Category Specific

Attribute Name	Display Name	Description
STARTDATE	Start Date	CLIENT --> STARTDATE
PROJECTNAME	Project Name	CLIENT --> PROJECTNAME

2. In the **Attributes not Indexed** section, select **All** from the **Attribute Group** drop-down. This is used as a filter, all the attributes in the repository are displayed.

3. If you have category specific attributes in the repository, select the **Category > Specific** check box, all the category specific attributes in the repository are displayed.

4. Search the attributes by specifying the pattern based search in the **Search** field. For example, if you type "start" in the search field all the attributes containing the text "start" is displayed.



Attributes not Indexed

Search: start

Attribute Group: All □ Category Specific

Attribute Name	Display Name	Description
STARTDATE	Start Date	CLIENT --> STARTDATE

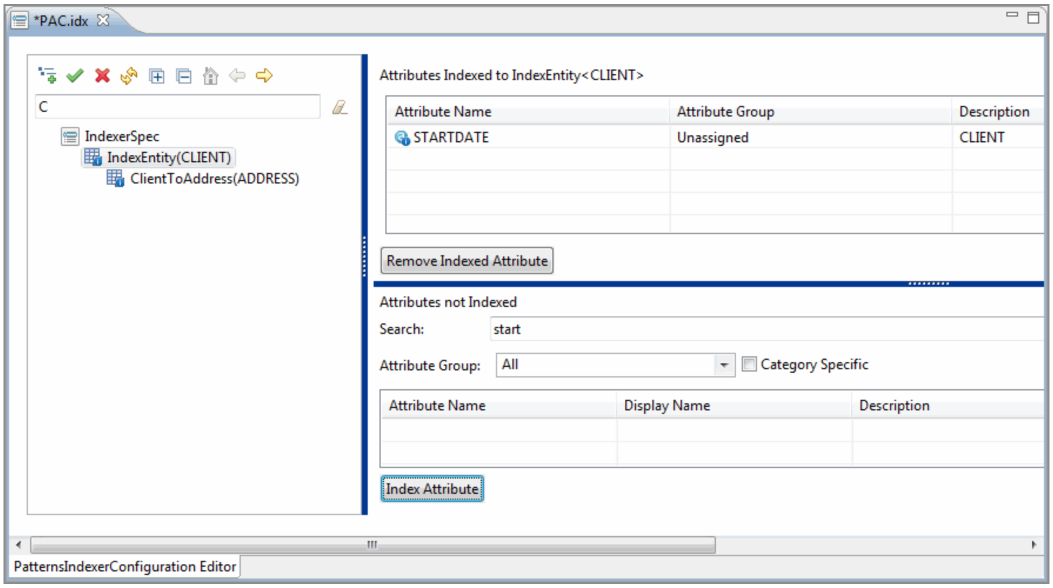
5. Select the attribute and click Index Attribute.

6. You can select one or more attributes to index using the ctrl+mouse click.

7. You can either sort or reorder the indexed attributes. Both sorting and re-ordering cannot be done together.

   - To sort the indexed attributes, click the **Attribute Name** column, the attributes are sorted in the ascending or descending order. The sorting feature is applicable only for the **Attribute Name** column.

- To reorder the indexed attribute, click any column (For example, **Attribute Group**) other than the **Attribute Name** column. Select and drag the indexed attributes to any position within the **Attribute Group** column.
- You can also sort after reordering the indexed attributes. To sort after reordering, click the **Attributes Name** column. The attribute name is sorted and the other column (**Attribute Group**) displays the reordered indexed attributes.



C

Attributes Indexed to IndexEntity<CLIENT>

Attribute Name	Attribute Group	Description
STARTDATE	Unassigned	CLIENT

Attributes not Indexed

Search: start

Attribute Group: All ☐ Category Specific

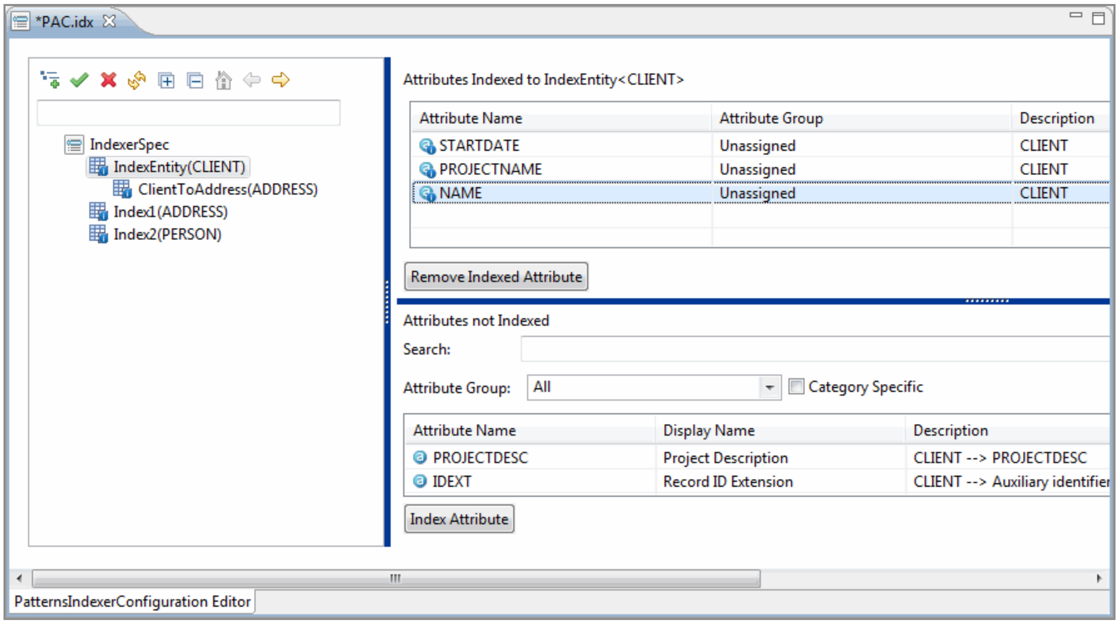
Attribute Name	Display Name	Description

UnIndexing Repository Attributes

Using this option you can unindex the attributes from the selected repository.

Procedure

1. Select the IndexEntity from which you want to unindex the repository attributes. All the attributes which are indexed are displayed.



Attributes Indexed to IndexEntity<CLIENT>

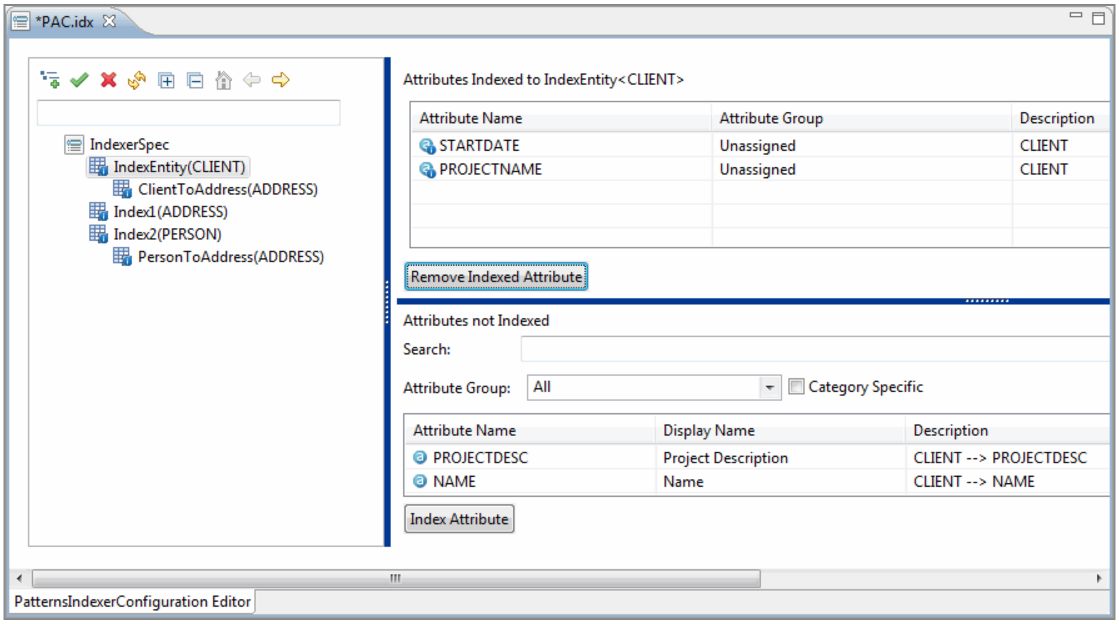
Attribute Name	Attribute Group	Description
STARTDATE	Unassigned	CLIENT
PROJECTNAME	Unassigned	CLIENT
NAME	Unassigned	CLIENT

Attributes not Indexed

Attribute Name	Display Name	Description
PROJECTDESC	Project Description	CLIENT --> PROJECTDESC
IDEXT	Record ID Extension	CLIENT --> Auxiliary identifier

2. Select the repository attributes which you want to unindex. For example, Name attribute and click Remove Indexed Attribute .

3. You can select one or more attributes to unindex.



Attributes Indexed to IndexEntity<CLIENT>

Attribute Name	Attribute Group	Description
STARTDATE	Unassigned	CLIENT
PROJECTNAME	Unassigned	CLIENT

Attributes not Indexed

Attribute Name	Display Name	Description
PROJECTDESC	Project Description	CLIENT --> PROJECTDESC
NAME	Name	CLIENT --> NAME

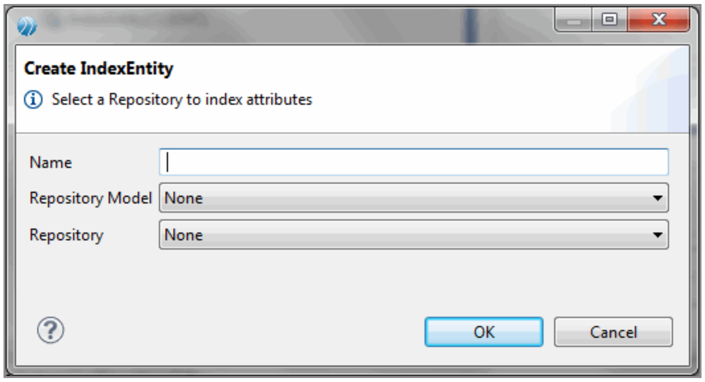
4. The unindexed attribute is removed from the **Attributes linkedto IndexEntity<Repository>** section. The unindexed attribute is displayed in table in the **Attribute not Indexed**.

Dragging and Dropping of Repository on Index Entity

You have an option to change the Repository by dragging a repository from the project explorer on to the "IndexEntity" node of the PatternsIndexer Editors Tree Control.

Procedure

1. Click ⊕ to add a IndexEntity. The Create IndexEntity wizard is displayed.



Create IndexEntity

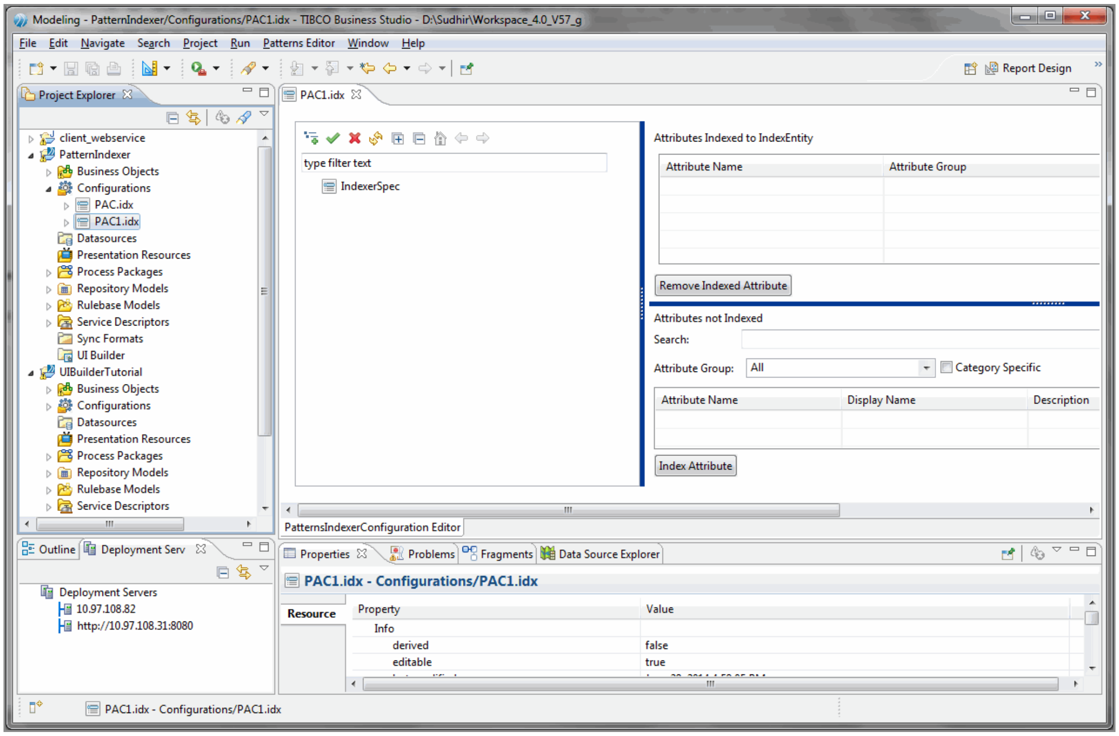
ⓘ Select a Repository to index attributes

Name:

Repository Model: None

Repository: None

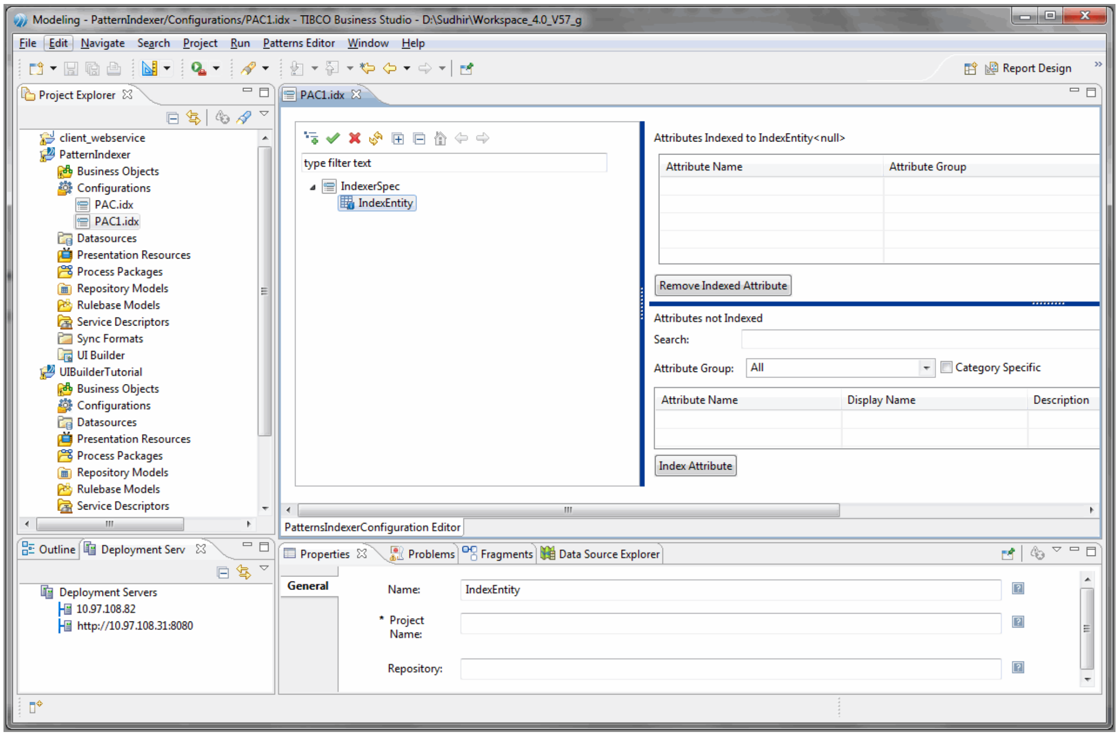
2. Click **OK** without entering the **Name**, **Repository Model**, and **Repository** fields.
3. The **PatternsIndexer Configuration Editor** is displayed.



Attribute Group: All

Property	Value
Info	
derived	false
editable	true

4. Expand the **IndexerSpec** node.



Attribute Group: All

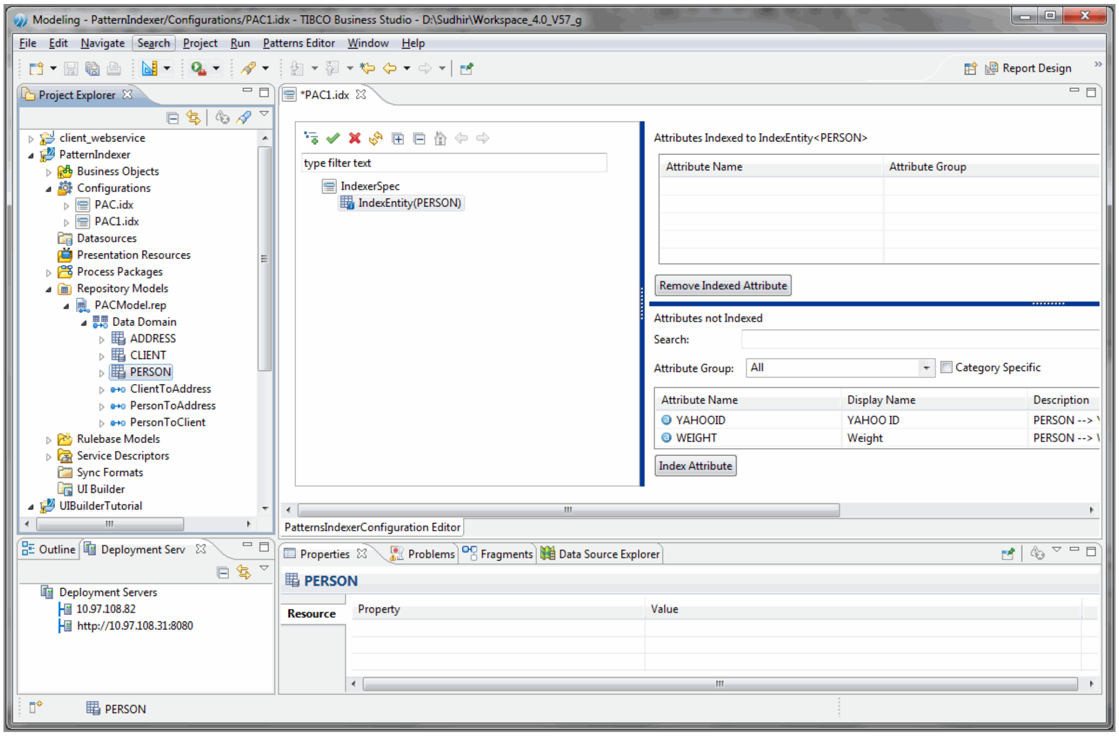
Name: IndexEntity

* Project Name:

Repository:

5. The newly created IndexEntity is displayed. Since you did not select any repository model, a blank IndexEntity is created.

6. From the explorer, select the repository for which you want to create index entities. Drag and drop the repository on the IndexEntity node.



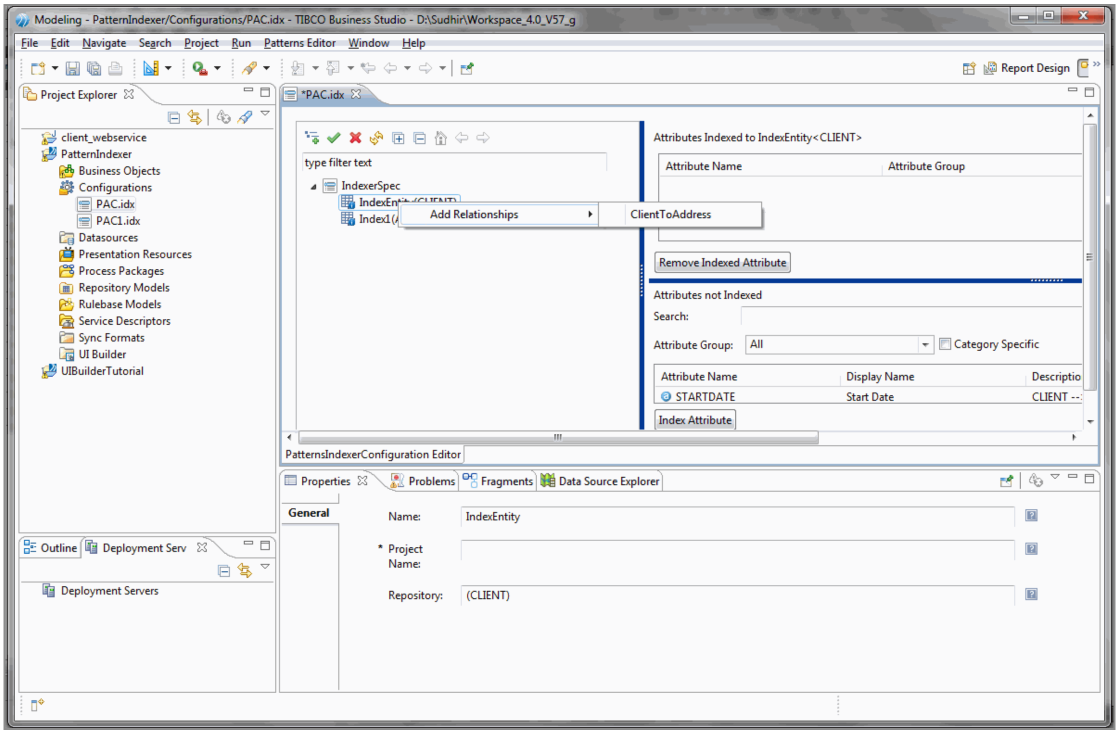
7. The repository name for which you have created the IndexEntity is displayed alongside the IndexEntity. For example, PERSON. The Repository name is not editable in the property section.

Adding Relationship (Target Repository) to IndexEntity

You can join repositories by adding relationships to IndexEntity. You must select the relationship from the parent repository to target repository.

Procedure

1. Select the IndexEntity and right-click the **Add Relationship** -<Relationship name> is displayed. Only forward relationship name is listed.

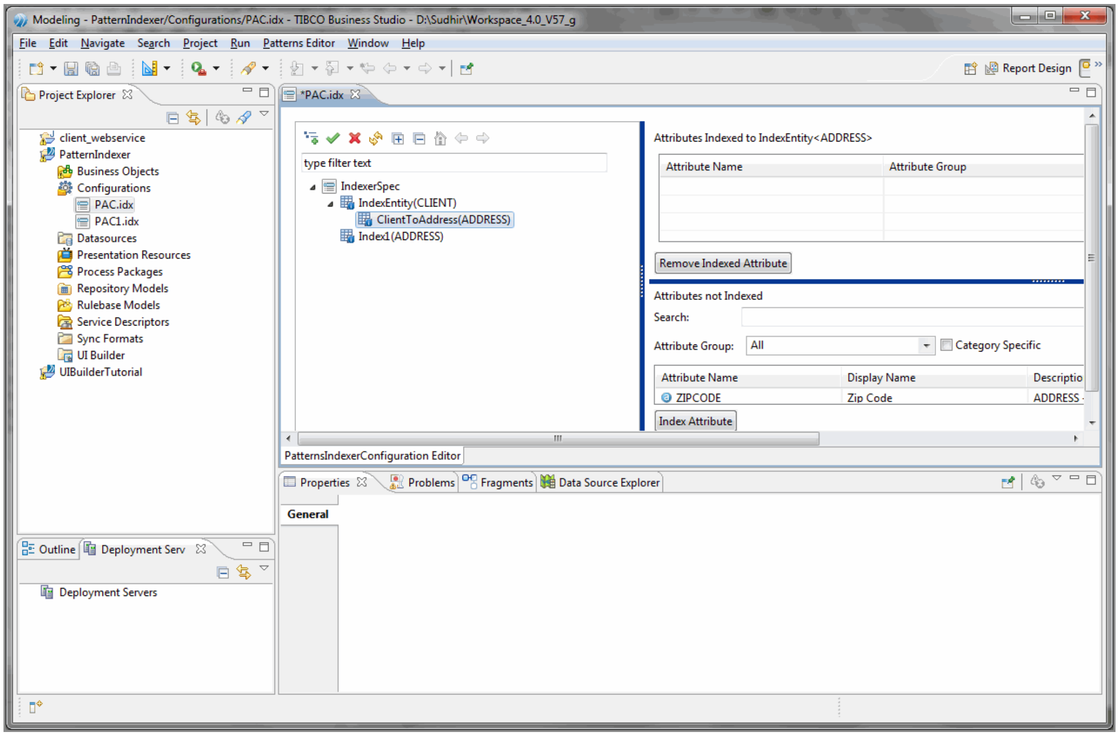


Name: IndexEntity

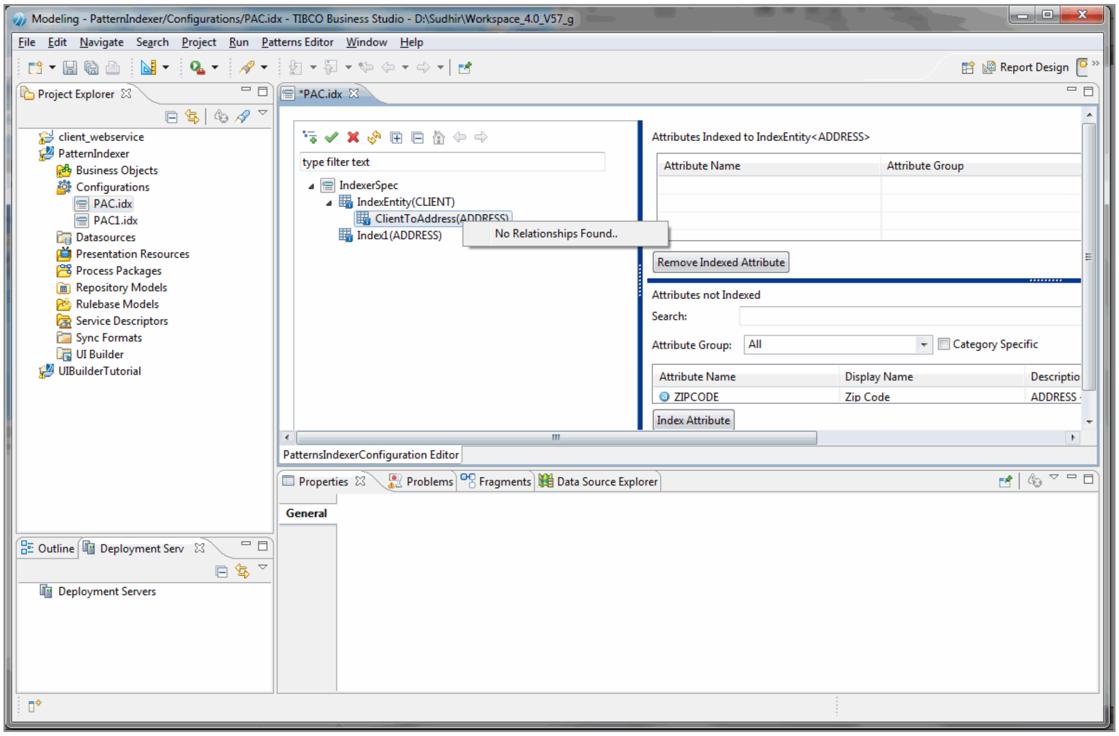
* Project Name:

Repository: (CLIENT)

2. Select any relationship. The relationship is added to the IndexEntity.

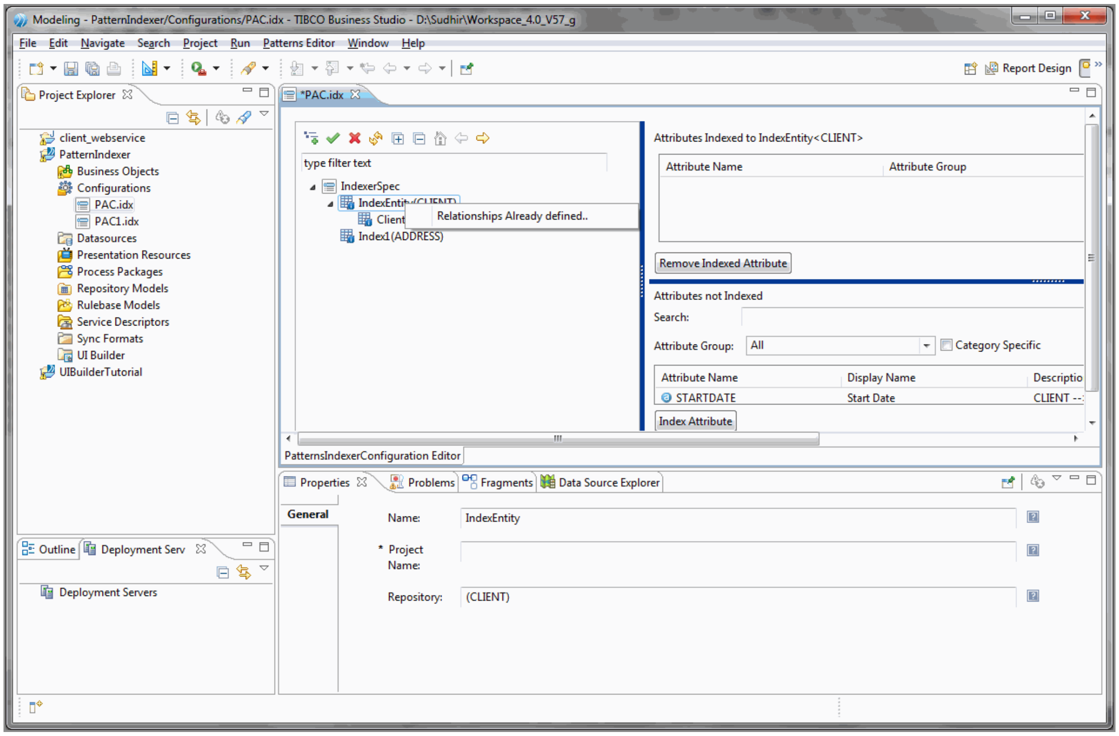


3. If there is no relationship is available to be added, the **No Relationship Found** message is displayed instead of the relationship name.



No Relationships Found..

4. Similarly if you have already added the relationship and are trying to add another then **Relationship Already Defined** message is displayed.



Relationships Already defined..

Attribute Group: All

Attribute Name	Display Name	Descriptio
STARTDATE	Start Date	CLIENT --:

Name: IndexEntity

* Project Name:

Repository: (CLIENT)

Delete IndexEntity

IndexEntity also has provision of deletion.

Select the IndexEntity or target repository (relationship name) which you want to delete and on the Tree Outline Toolbar click ✖ .

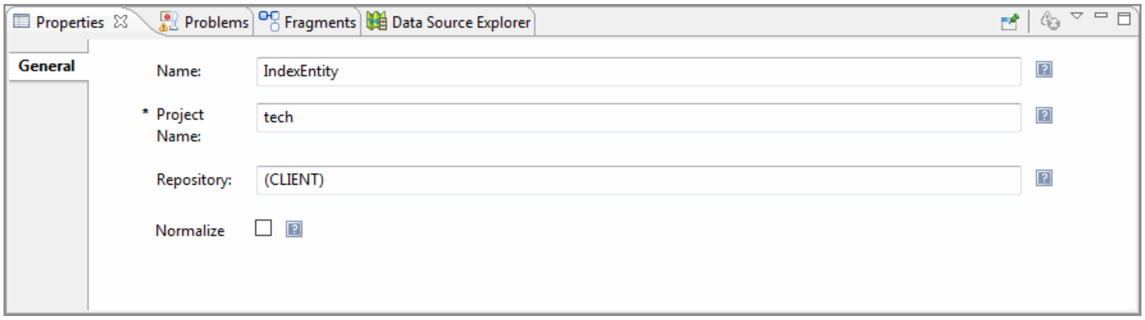
Validate IndexEntity

The PatternsIndexer Configuration Editor uses the repositories from the .rep file. You can change the repository or relationships parameters, any modification to the repository name is directly reflected in the tree-control.

However, any deletion or addition of an attribute or a relationship or a repository is not reflected. In order to sync-up the modification to the repository with the PatternsIndexer Configuration Editor you must click ✓ .

Modify the IndexEntity Properties

When you select the IndexEntity, the property-section associated with the IndexEntity is displayed, you can edit the properties of the IndexEntity.



Name: IndexEntity

* Project Name: tech

Repository: (CLIENT)

Normalize □

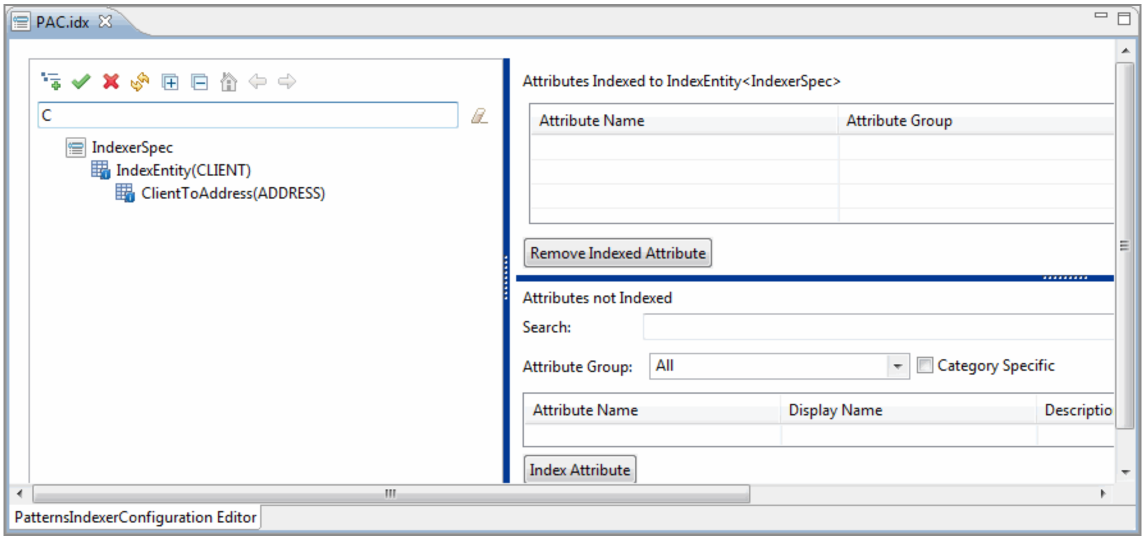
Field	Description
Name	Displays the name of the IndexEntity. You can modify the IndexEntity name.
Project Name	Displays the name of the project name associate with the Repository. You can modify the project name. This field is mandatory.
Repository	Displays the name of the repository associated with IndexEntity or TargetRepository. It is a non-editable field.
Normalize	Select the check box to normalize.

Searching Nodes in IndexConfig Hierarchy

You can search for a IndexEntity using the pattern matching string as the search criteria.

Type the pattern matching string in the search field. For example, on typing *C displays IndexEntity names starting with anything and having a letter C.

To clear the search criteria click .

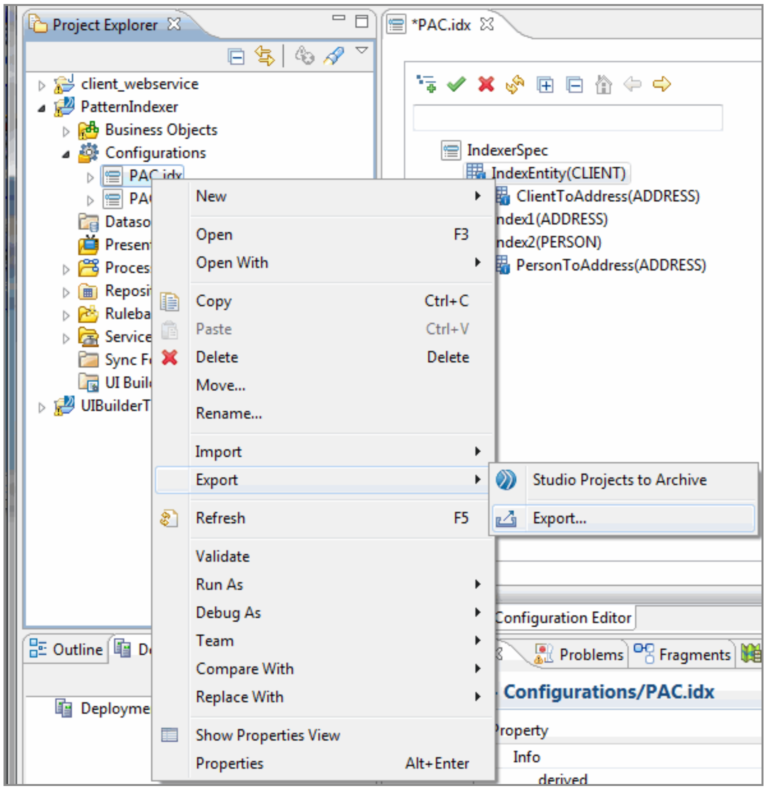


Exporting IndexEntity

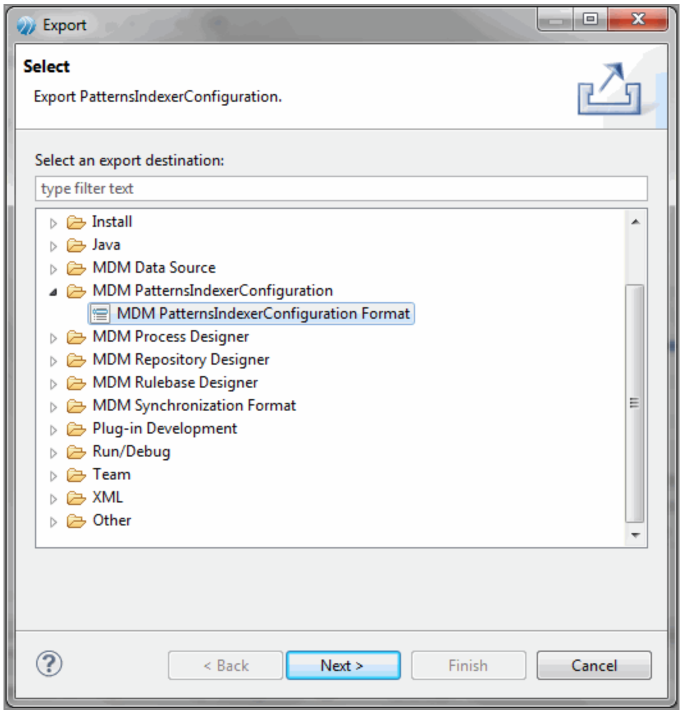
In addition to creating the IndexEntity, you can also export the PatternsIndexer. All the IndexEntities defined in the PatternsIndexer Configuration is exported.

Procedure

1. Right click on the newly created PatternsIndexer Configuration, select **Export > Export**.



2. The Export Wizard for selecting the PatternsIndexer Configuration is displayed.



Export

Select

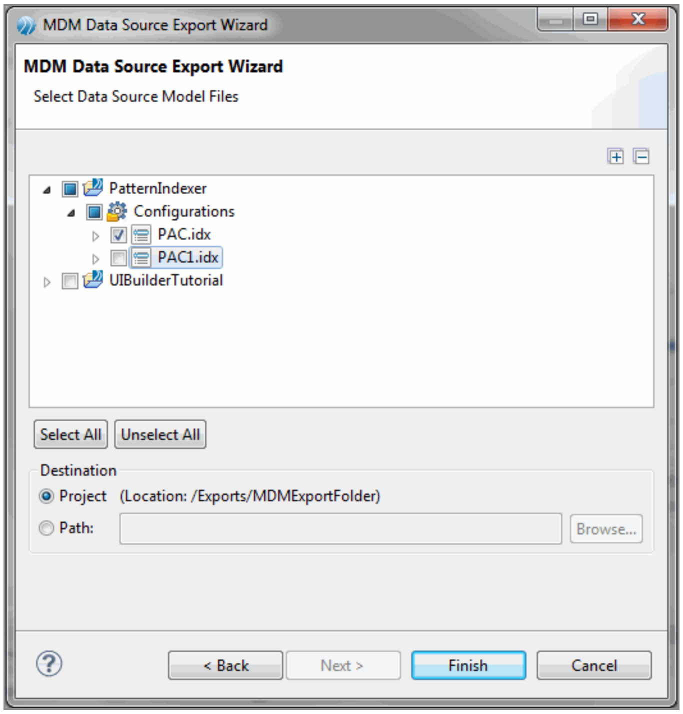
Export PatternsIndexerConfiguration.

Select an export destination:

type filter text

3. Select **MDM PatternsIndexer Configuration Format** and click **Next**.

4. The MDM Data Source Export Wizard for selecting the data source model is displayed.



☑ PAC.idx

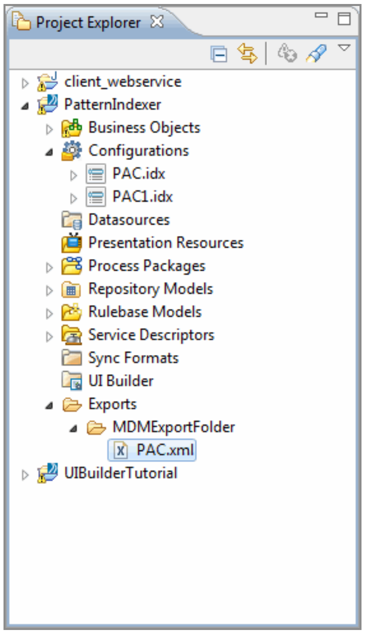
□ PAC1.idx

● Project (Location: /Exports/MDMExportFolder)

○ Path:

5. Select the PatternsIndexer which you want to export. For example, **PAC.idx** and click **Finish**.

6. The Project Export shows the exported PatternsIndexer file.



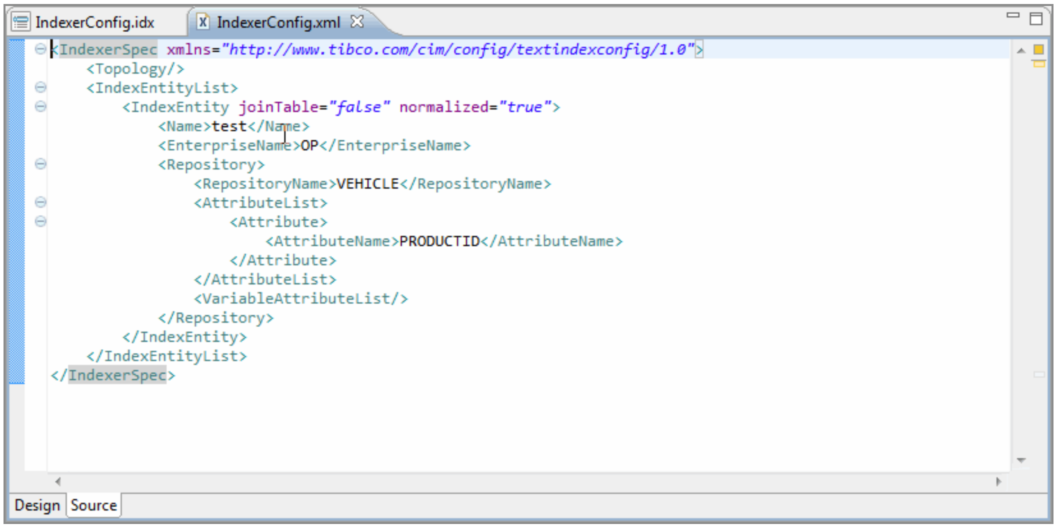
Deploy PatternsIndexer Configuration

The PatternsIndexer Configuration is created for an existing repository model and contains all the necessary information.

The Configurations folder is shown in the deployment wizard, you can select the IndexerConfig file to deploy the artifacts to the TIBCO MDM Server.

For more information on how to deploy PatternsIndexer Configuration see Deployment.

After deploying the text indexer on TIBCO MDM server. The index file can be viewed from `$MQ_HOME\config\IndexerConfig.xml`



```
<IndexerSpec xmlns="http://www.tibco.com/cim/config/textindexconfig/1.0">
    <Topology/>
    <IndexEntityList>
        <IndexEntity joinTable="false" normalized="true">
            <Name>test</Name>
            <EnterpriseName>OP</EnterpriseName>
            <Repository>
                <RepositoryName>VEHICLE</RepositoryName>
                <AttributeList>
                    <Attribute>
                        <AttributeName>PRODUCTID</AttributeName>
                    </Attribute>
                </AttributeList>
                <VariableAttributeList/>
            </Repository>
        </IndexEntity>
    </IndexEntityList>
</IndexerSpec>
```

Whenever a new `IndexerConfig.xml` is deployed, only the copy of the previous version is maintained.

You can restore the `IndexerConfig.xml` from the backup location. Import of patterns `IndexConfig.xml` file is not supported.

After deployment, the records are indexed only if the Patterns server is running.

Working with Custom Query Designer

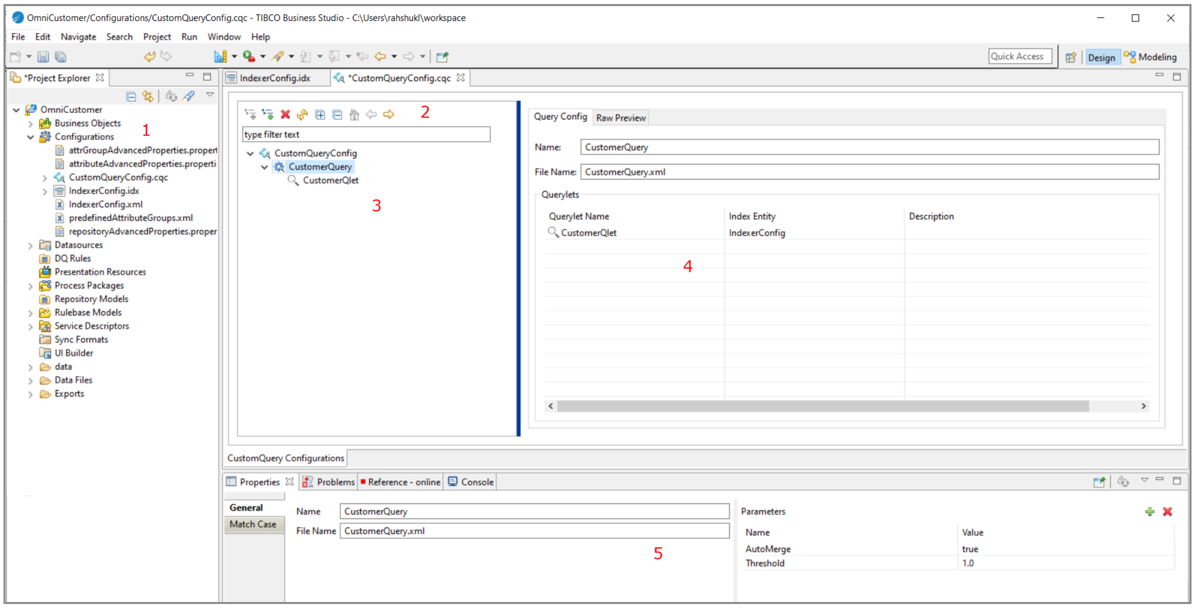
This section talks about the Custom Query Configuration Model in TIBCO MDM Studio and explains how to create and deploy the Custom Query Configuration usnig the Custom Query Designer.

Overview

In TIBCO MDM Studio you can design custom query configuration and deploy it on the TIBCO MDM server.

Using the Custom Query Configuration editor you can define a Custom Query. The intuitive user interface of the editor takes advantage of the Pattern indexer config model file to define the custom query model.

Custom Query Configuration Interface



Name: CustomerQuery

File Name: CustomerQuery.xml

Querylets

Querylet Name	Index Entity	Description
CustomerQlet	IndexerConfig	

Name: CustomerQuery

File Name: CustomerQuery.xml

Parameters

Name	Value
AutoMerge	true
Threshold	1.0

1. **Configurations folder**: All the Custom Query config files are available inside the project's configuration folder.

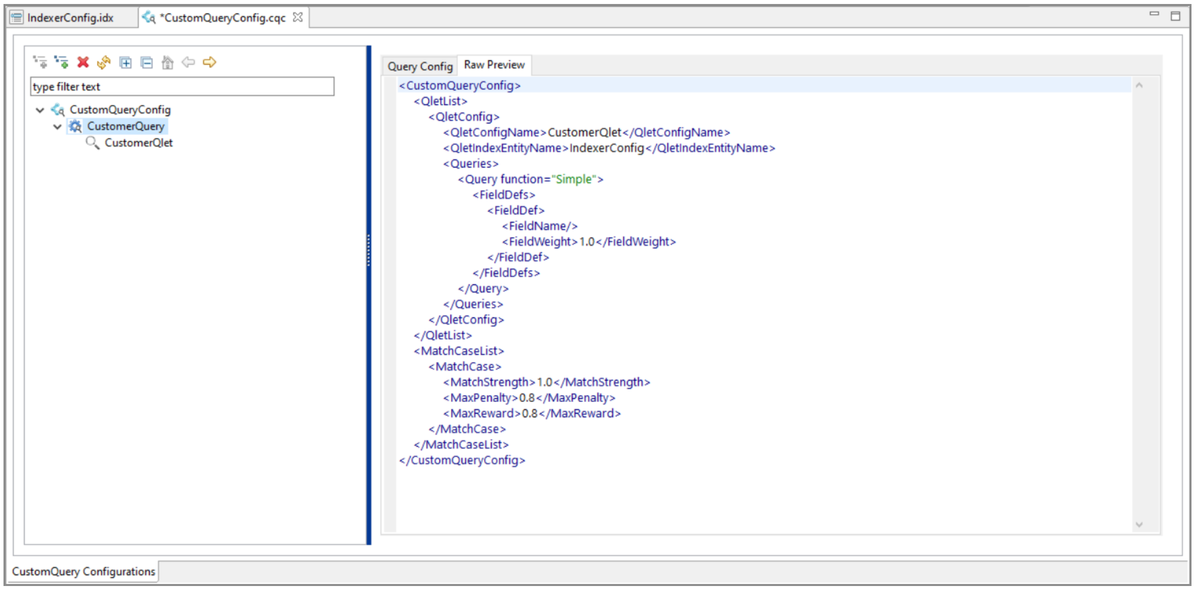
2. **Custom Query Toolbar**: The Custom Query editor toolbar displays the toolbar items for designing, deleting, reordering, refreshing, expanding, and collapsing the tree outline.

Tree Controls	
Editing Controls	
➕	Create a Query Config or Querylet.
✖	Delete a Query Config or Querylet.
Navigating Controls	
⟳	Refresh the Custom Query model.
⊞	Expand to view all constituents of a Custom Query Config.
⊟	Collapse all constituents of a Custom Query Config.
🏠	Home. If you are inside a child node of a Custom Query, click the icon to return the main Custom Query.
⇦	Back. If you are inside a child node of a Custom Query, click the icon to return back.
⇨	Go Into. Click this icon to view the child node Custom Query.

3. **Custom Query Tree Viewer**: Contains defined Match Rules that are executed in sequential order. If the first Match Rule does not find any match, the Match engine executes the next match rule, and soon.

4. **Custom Query Properties**: Panels 4 and 5 in the image above display the properties of the Custom Query Config constituents (Query Config, Querylet, Query, and Match Case rules).

You can also view a raw preview of the MatchRules or QueryConfiguration xml files.



```
<CustomQueryConfig>
  <QletList>
    <QletConfig>
      <QletConfigName>CustomerQlet</QletConfigName>
      <QletIndexEntityName>IndexerConfig</QletIndexEntityName>
      <Queries>
        <Query function="Simple">
          <FieldDefs>
            <FieldDef>
              <FieldName/>
              <FieldWeight>1.0</FieldWeight>
            </FieldDef>
          </FieldDefs>
        </Query>
      </Queries>
    </QletConfig>
  </QletList>
  <MatchCaseList>
    <MatchCase>
      <MatchStrength>1.0</MatchStrength>
      <MaxPenalty>0.8</MaxPenalty>
      <MaxReward>0.8</MaxReward>
    </MatchCase>
  </MatchCaseList>
</CustomQueryConfig>
```

Creating a Custom Query Model

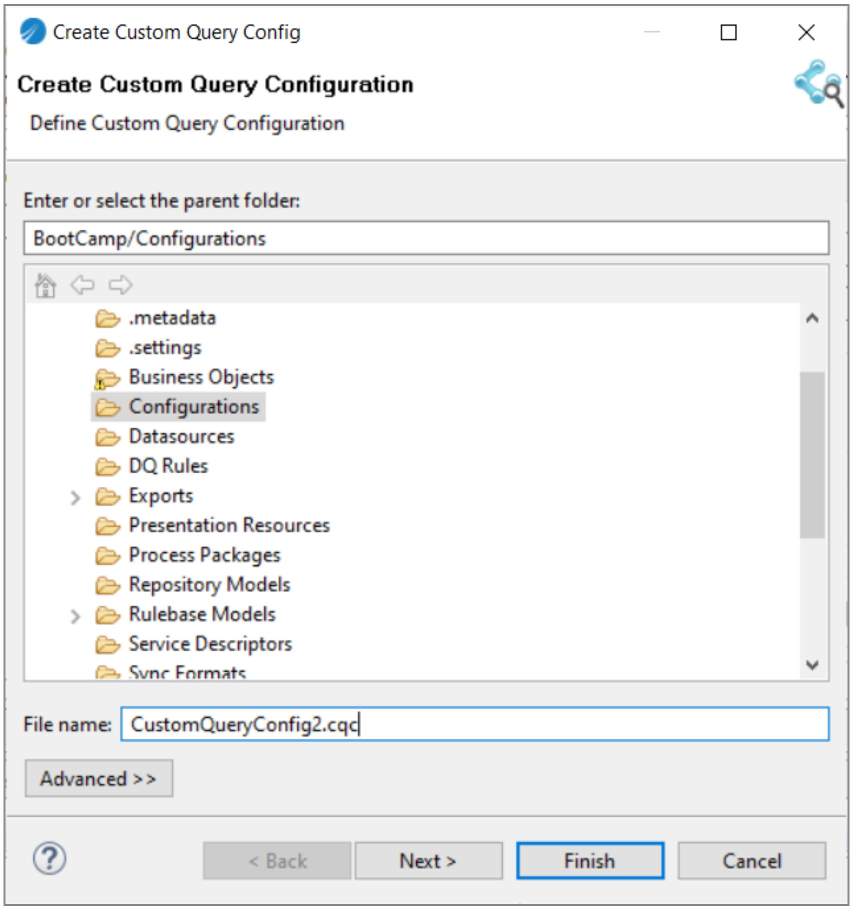
You can create a Custom Query Model by using the Custom Query Creation wizard. Follow these steps to create a Custom Query Model.

Before you begin

Ensure that the PatternsIndexer Configuration is created and all the artifacts are deployed on the TIBCO MDM Server. For more information about PatternsIndexer Configuration, see the Working with PatternsIndexer Configuration Editor section.

Procedure

1. In the Project Explorer, right click the **Configurations** folder and then select **New > Custom Query Config Model**. The Create Custom Query Configuration wizard is displayed.



Create Custom Query Config

Create Custom Query Configuration

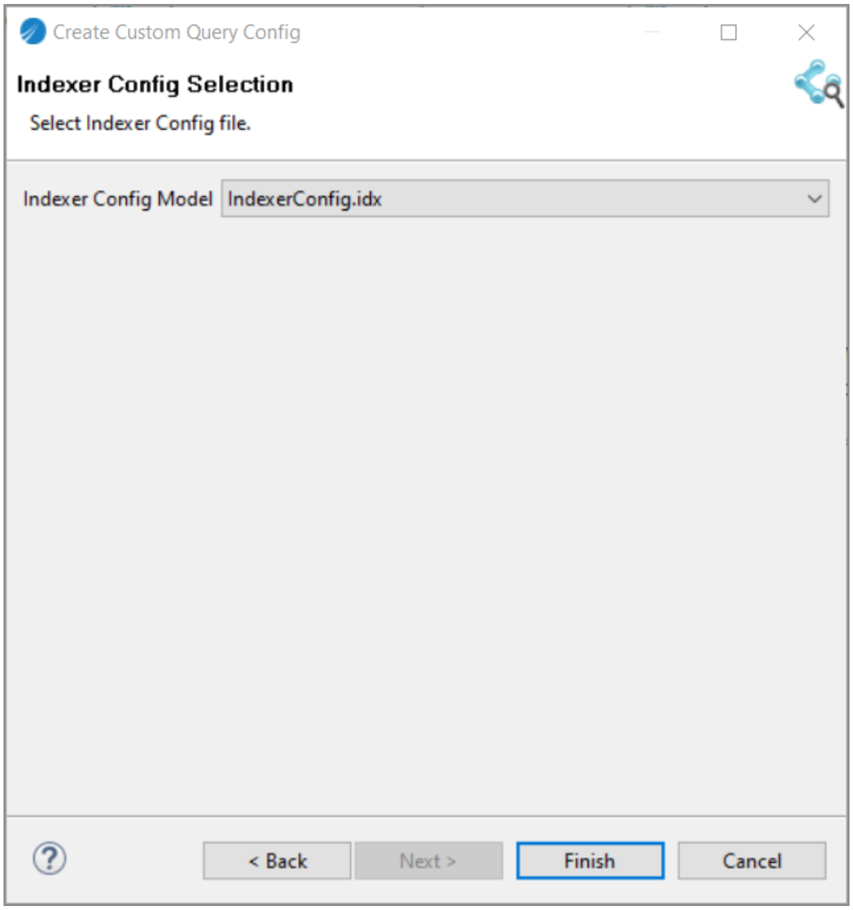
Define Custom Query Configuration

Enter or select the parent folder:

BootCamp/Configurations

File name: CustomQueryConfig2.cqc

2. Enter or select a parent folder and provide a name for the Custom Query Config. Ensure that the file extension is **.cqc**. Click **Next**. The Indexer Config Selection wizard is displayed.



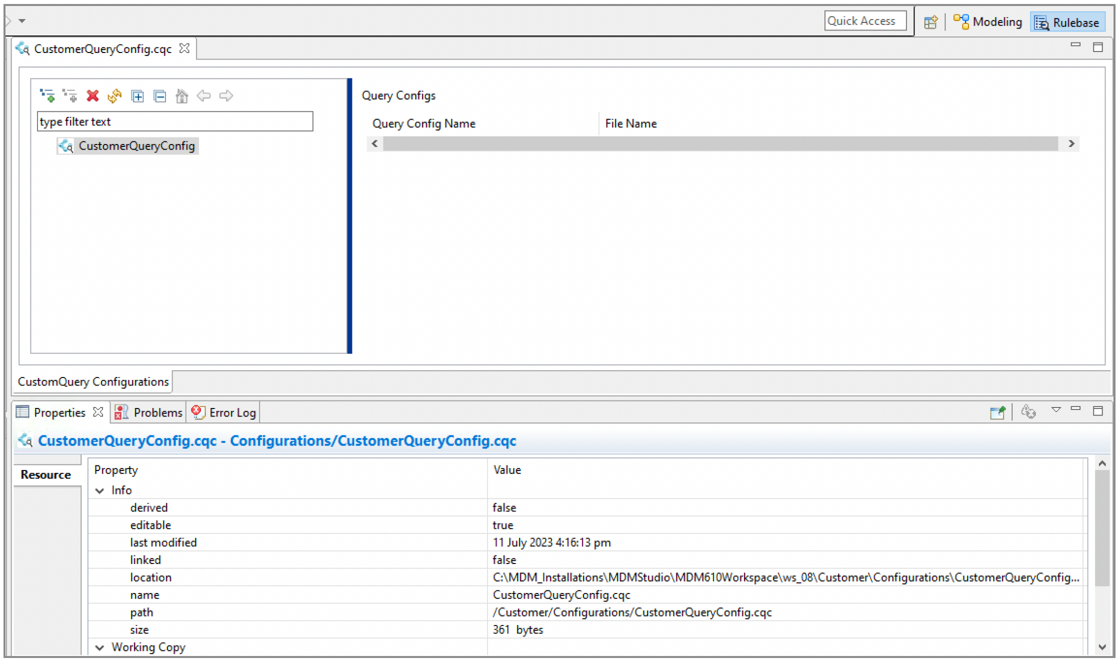
Indexer Config Selection

Select Indexer Config file.

Indexer Config Model: IndexerConfig.idx

To know more about the Indexer Configuration file, see the Creating a PatternsIndexer Configuration section.

3. Select the Indexer Configuration file and click **Finish**. The Custom Query Model is created and displayed in Custom Query Config editor.



CustomerQueryConfig.cqc - Configurations/CustomerQueryConfig.cqc

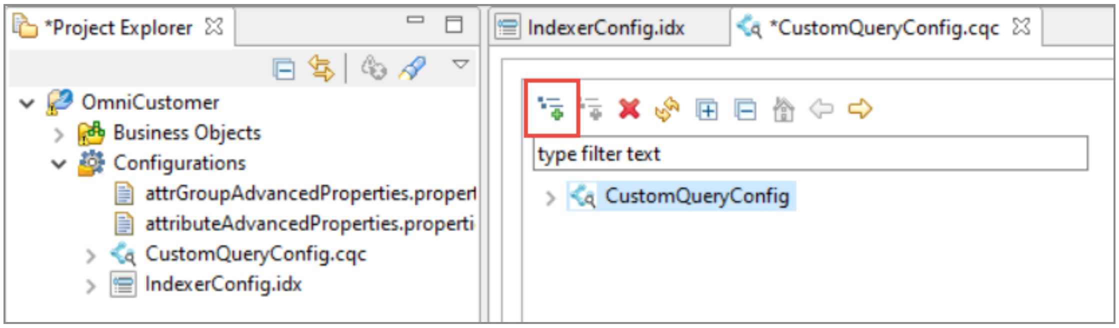
Property	Value
Info	
derived	false
editable	true
last modified	11 July 2023 4:16:13 pm
linked	false
location	C:\MDM_Installations\MDMStudio\MDM610Workspace\ws_08\Customer\Configurations\CustomerQueryConfig...
name	CustomerQueryConfig.cqc
path	/Customer/Configurations/CustomerQueryConfig.cqc
size	361 bytes
Working Copy	

Adding a Query Config

Complete the following steps to add a Query Config.

Procedure

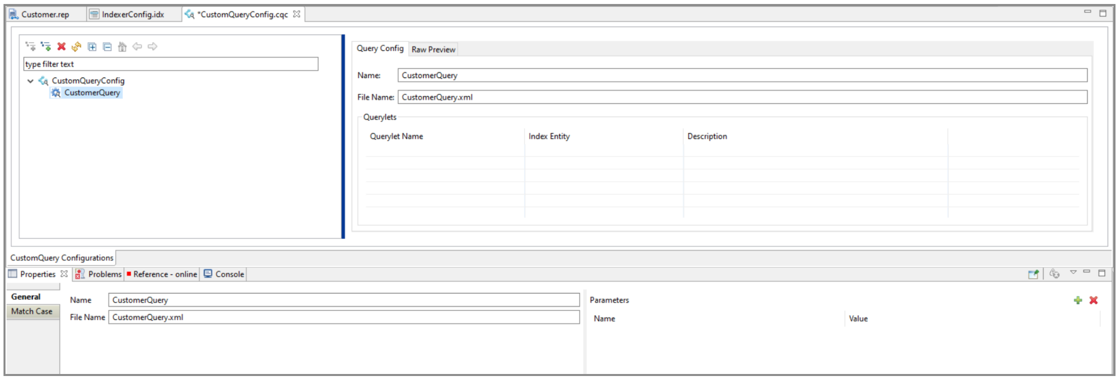
1. From the Project Explorer, navigate to **Configurations > CustomQueryConfig.cqc**. The custom query config panel is displayed.



2. Click the icon to add a new query config. The **Create Query Config** dialog is displayed.

3. Enter a Query Name and click **OK**. The new Query Config is added and displayed in

the right side panel.



Name: CustomerQuery
File Name: CustomerQuery.xml

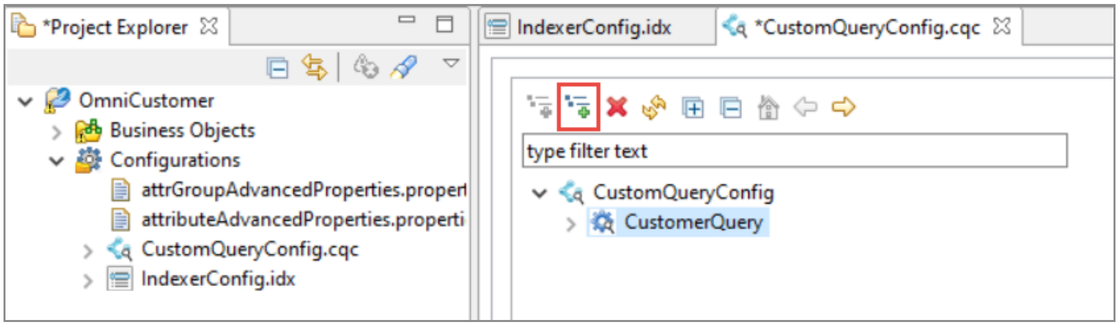
You can edit the Query Config name and file name in the right-side panel of the Custom Query editor or from the Properties tab. Additionally, you can specify parameters specific to Custom Query Config under **Properties > General > Parameters**.

Adding a Querylet

Complete the following steps to add a Querylet.

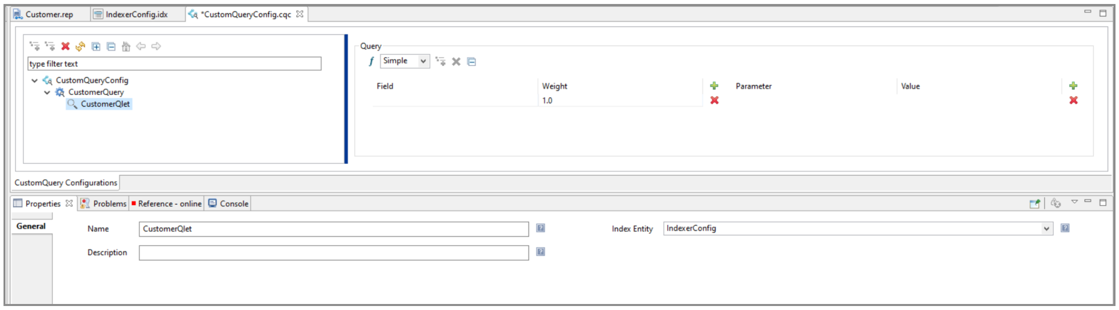
Procedure

1. From the Project Explorer, navigate to **Configurations > CustomQueryConfig.cqc**. The custom query config panel is displayed.



2. Select a query config and then click the icon to add a new Querylet. The Create Querylet Config dialog box is displayed.
3. Enter a Querylet Name and click **OK**. The Query section is displayed on the right side where you can build your query function.

4. Select the **Index Entity** from the Properties tab.



Simple

Field	Weight
	1.0

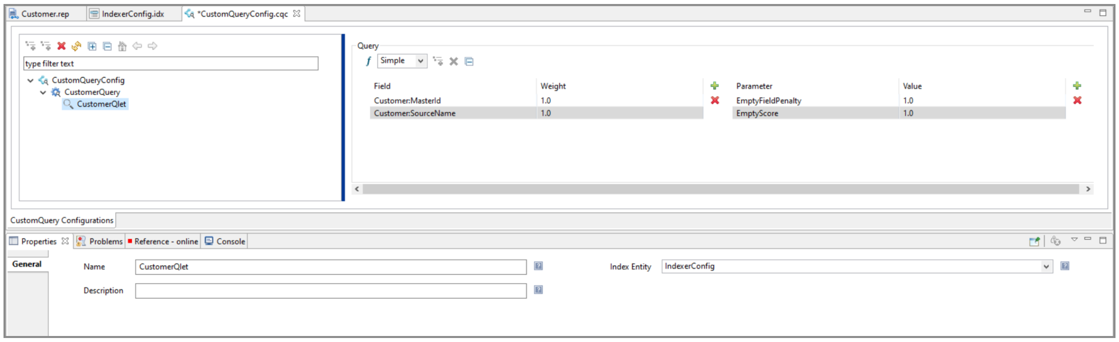
Parameter	Value

Name: CustomerQlet

Description:

Index Entity: IndexerConfig

5. In the Query section, select a function from the dropdown menu.
6. Click the ➕ button and select the Fields and Parameters from the dropdown menu to add into the query function. Click the ✖ button to remove a Field or Parameter.
7. Customize the values and weights for Fields and Parameters.



Simple

Field	Weight
Customer:MasterId	1.0
Customer:SourceName	1.0

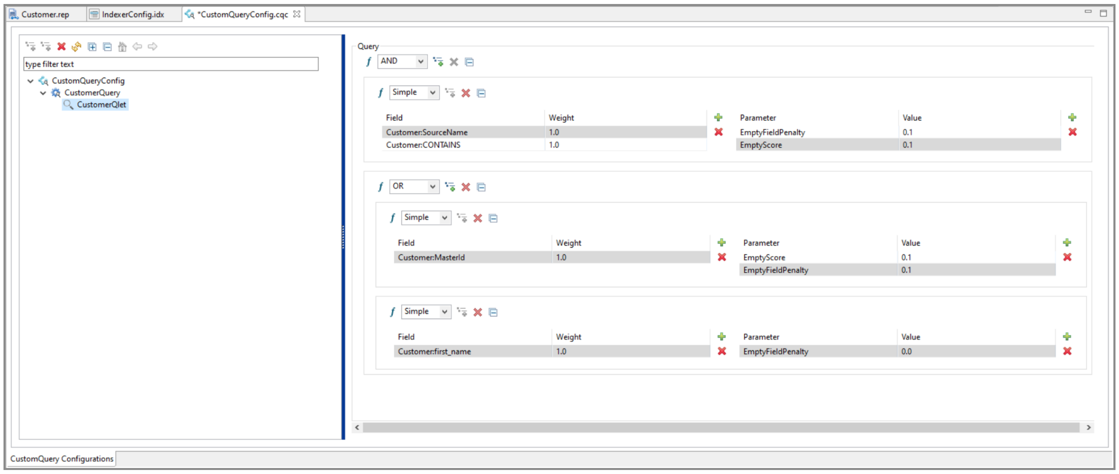
Parameter	Value
EmptyFieldPenalty	1.0
EmptyScore	1.0

Name: CustomerQlet

Description:

Index Entity: IndexerConfig

You can organize queries into nested structures when the function represents a logical group. You can create logical groups by using operators such as AND and OR. An example for a logical group function is shown below:



Simple

Field	Weight
Customer:SourceName	1.0
Customer:CONTAINS	1.0

Parameter	Value
EmptyFieldPenalty	0.1
EmptyScore	0.1

Simple

Field	Weight
Customer:MasterId	1.0

Parameter	Value
EmptyScore	0.1
EmptyFieldPenalty	0.1

Simple

Field	Weight
Customer:first_name	1.0

Parameter	Value
EmptyFieldPenalty	0.0

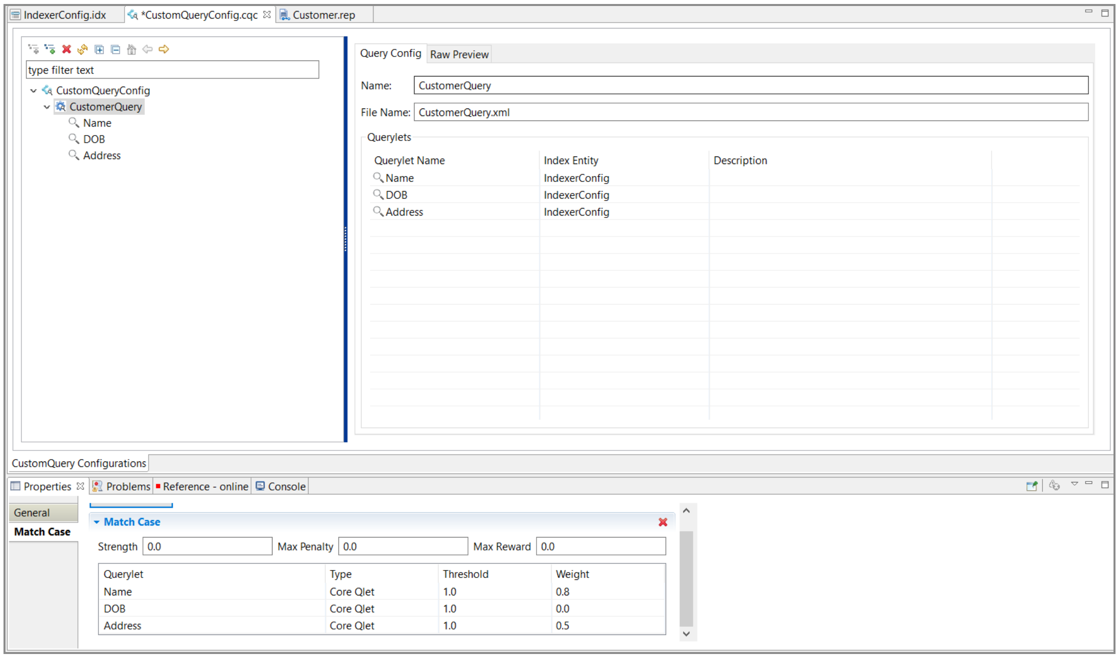
Note: You cannot add fields and parameters for logical functions.

Adding a Match Case

Complete the following steps to add a Match Case.

Procedure

1. From the Project Explorer, navigate to **Configurations > CustomQueryConfig.cqc**.
2. In the CustomeQueryConfig.cqc panel, expand the CustomQueryConfig and select a Query Config where you want to add Match Cases.
3. In the Properties tab, click **Match Case** and then click **Add Match Case** to provide the match case rule details.
4. For each Querylet, specify the **Strength**, **Max Penalty**, **Max Reward**, **Threshold**, and **Weight**.
5. Select the Querylet type from **Type** dropdown menu.



Name: CustomQuery

File Name: CustomQuery.xml

Querylets

Querylet Name	Index Entity	Description
Name	IndexerConfig	
DOB	IndexerConfig	
Address	IndexerConfig	

Match Case

Strength 0.0 Max Penalty 0.0 Max Reward 0.0

Querylet	Type	Threshold	Weight
Name	Core Qlet	1.0	0.8
DOB	Core Qlet	1.0	0.0
Address	Core Qlet	1.0	0.5

Custom Query Config Validations

There are several validations applied on Custom Query Config to ensure it's error-free and accurate export. The following are the validations that are applied to the Custom Query Config.

Validations on Custom Query Config:

- Query config name cannot be empty.
- Custom config query file should be given to custom query config.
- Each custom query confing file should have a unique file name.
- Each custom query confing should have a unique name.
- The custom query config file format must be `.xml`.

Validations on Querylet Config:

- Querylet name cannot be empty.
- Index entity cannot be empty for a querylet. There should be atleast one index entity associated with a querylet.
- Invalid index entity should not be associated with a querylet.

- Each custom querylet should have a unique name.
- The querylet type cannot be empty.

Validations on Match Case:

- Max reward value should be between 0.00 to 1.00.
- The strength value should be between 0.00 to 1.00.
- The max penalty value should be between 0.00 to 1.00.
- The decimal places in the max reward value should not be more than two.
- The decimal places in the strength value should not be more than two.
- The decimal places in the max penalty value should not be more than two.
- The threshold value for core querylet should be between -1.00 to 1.00.
- The threshold value for other querylets except the core querylet should be between 0.00 to 1.00.
- The decimal places in the threshold value should not be more than two.
- The weight value for core querylet should be between -1.00 to 1.00.
- The weight value for other querylets except the core querylet should be between 0.00 to 1.00.
- The decimal places in the weight value should not be more than two.
- At least one match case must be defined for query config.

Validations on Query:

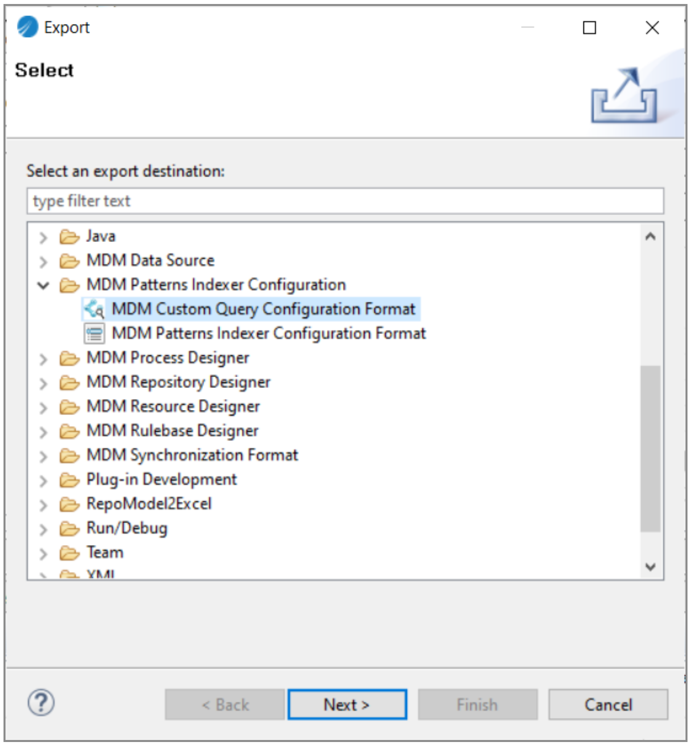
- At least one query should be defined for a querylet.
- At least two sub queries must be defined for AND & OR function. You can define more than two sub queries.
- The weight value for function definition should be between 0.00 to 1.00.
- The decimal places in the weight value for function definition should not be more than two.
- Field definition name and parameter name must be unique.

Exporting Custom Query Config

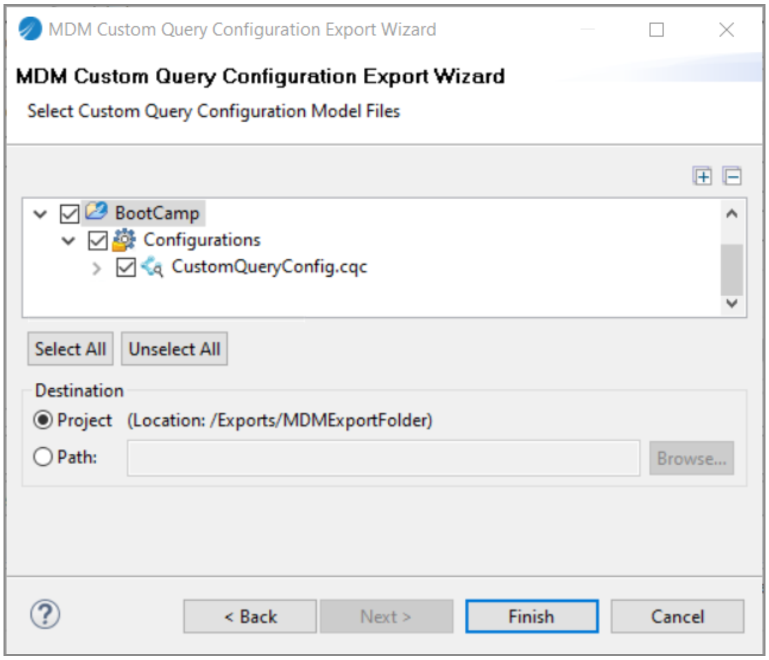
Complete the following steps to export a Custom Query Config Model.

Procedure

1. In the Project Explorer, right-click the **Configurations** folder and click **Export**. The Export wizard is displayed.

2. Expand **MDM Patterns Indexer Configuration**, select **MDM Custom Query Configuration Format**, and then click **Next**. The MDM Custom Query Configuration Export Wizard is displayed.



MDM Custom Query Configuration Export Wizard

Select Custom Query Configuration Model Files

☑ BootCamp
☑ Configurations
☑ CustomQueryConfig.cqc

Destination
● Project (Location: /Exports/MDMExportFolder)
○ Path:

3. Select the `CustomQueryConfig.cqc` file checkbox.
4. The default export destination is Project (Location:/Exports/MDMExportFolder). If you want to change the export folder, select **Path** and click **Browse** to select a different destination export folder.

   > **Note:** You can export multiple Custom Query Config at the same time.

5. Click **Finish** to export the file.

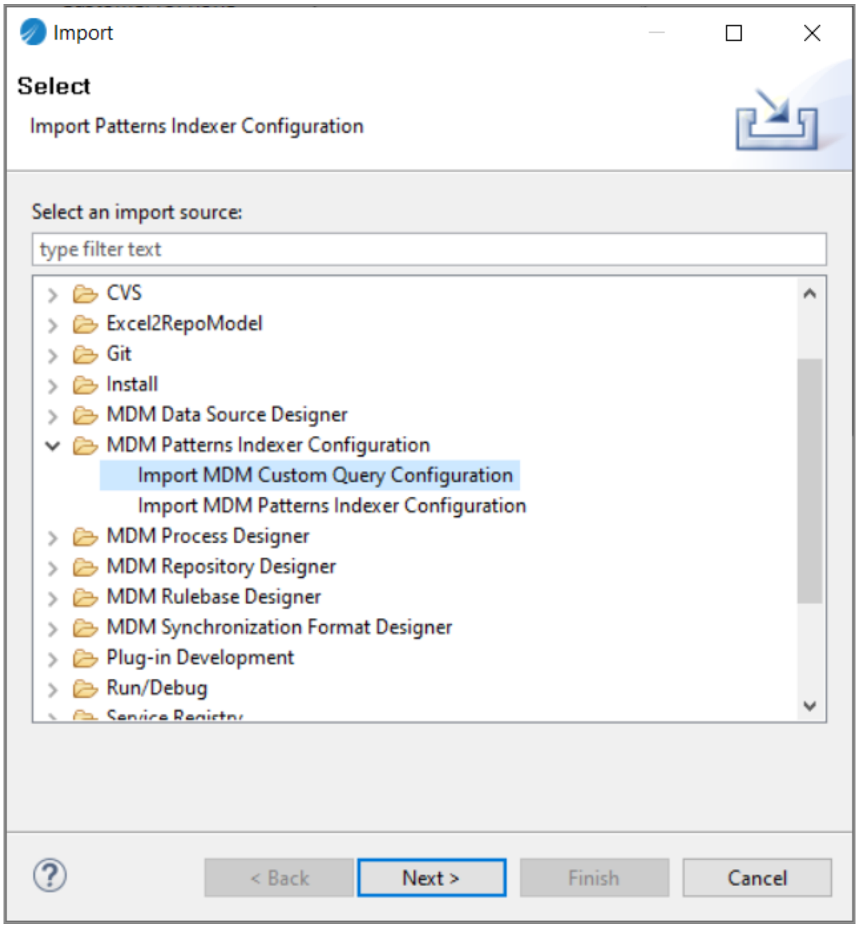
   The Custom Query Config is exported as a zip file in the Exports folder.

Import Custom Query Config Model

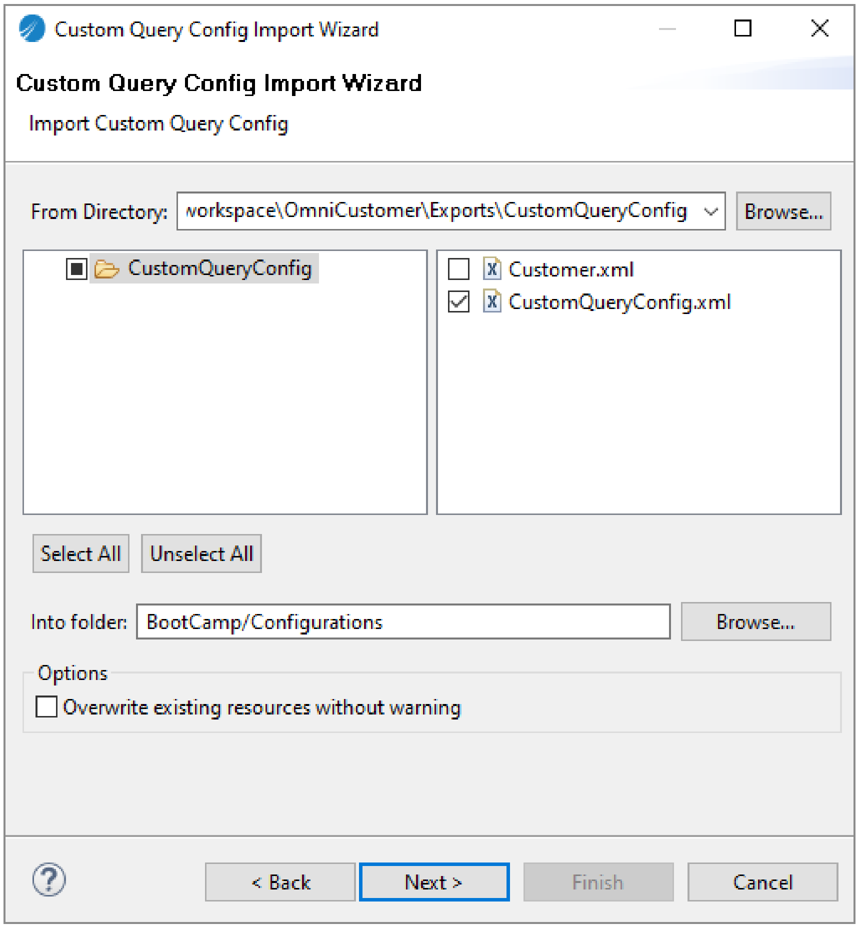
Complete the following steps to import a Custom Query Config Model.

Procedure

1. In the Project Explorer, right-click the **Configurations** folder and then click **Import**. The Import Patterns Indexer Configuration wizard is displayed.



2. Expand **MDM Patterns Indexer Configuration**, select **Import MDM Custom Query Configuration**, and then click **Next**. The Custom Query Config Import wizard is displayed.



Custom Query Config Import Wizard

Custom Query Config Import Wizard

Import Custom Query Config

From Directory: workspace\OmniCustomer\Exports\CustomQueryConfig

■ CustomQueryConfig

□ Customer.xml

☑ CustomQueryConfig.xml

Into folder: BootCamp/Configurations

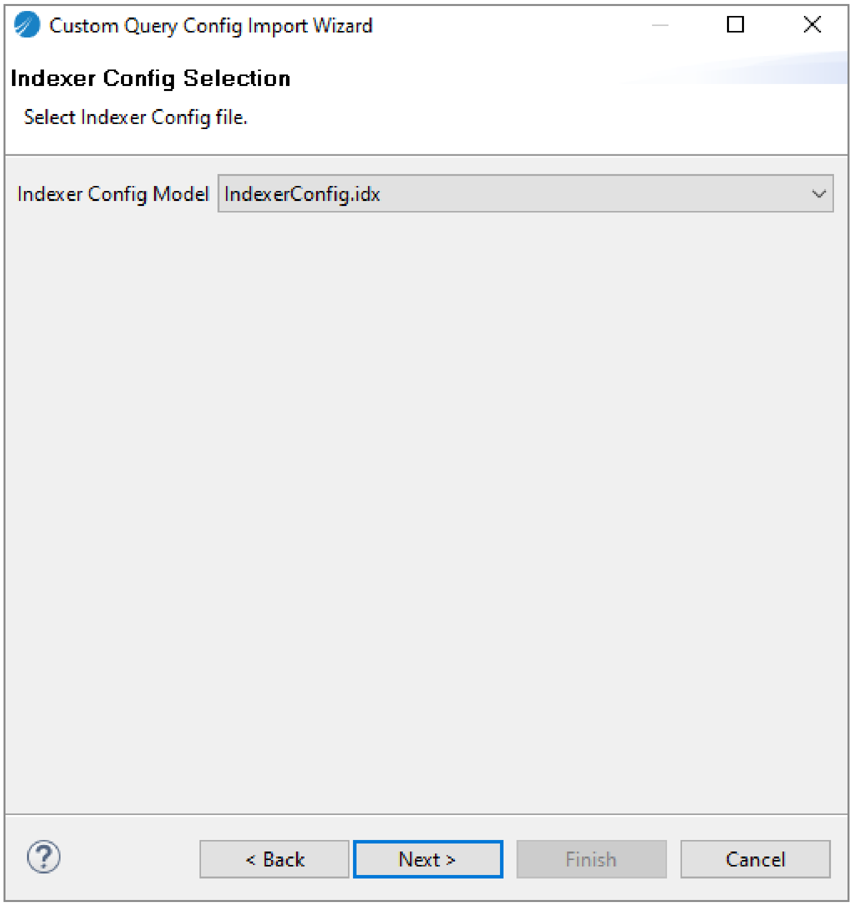
Options

□ Overwrite existing resources without warning

3. Click **Browse** to navigate to the **From Directory** where a valid `customQueryConfig.xml` file is saved.

4. Select the `customQueryConfig.xml` file and click **Next**. The Indexer Config Selection wizard is displayed.

> **Note:** Do not select referenced Query Config files as the referenced files are also saved in the same directory where the main `customqueryconfig.xml` is saved.

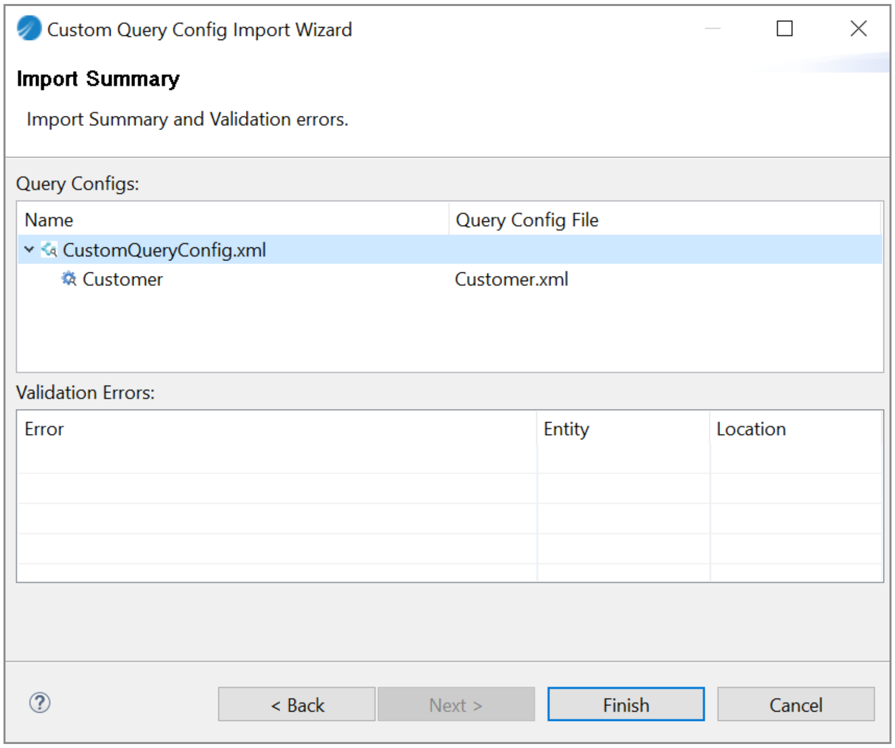


Indexer Config Selection

Select Indexer Config file.

Indexer Config Model: IndexerConfig.idx

5. Select referenced indexer configuration file and click **Next**.
6. The Import Summary wizard is displyed. This wizard diaplays the import summary and validation errors, if any. Click **Finish**.



Custom Query Config Import Wizard

Import Summary

Import Summary and Validation errors.

Query Configs:

Name	Query Config File
CustomQueryConfig.xml	
Customer	Customer.xml

Validation Errors:

Error	Entity	Location

The Custom Query Config Model is imported in the current project.

WSDL and Service Methods

The WSDL chapter covers the High level features and the implementation scope for dynamic web services. It also describes the optional generation of dynamic web services for high usability and deployment of new web services unlike static web services which are bundled with TIBCO MDM.

Overview

The current web service interface is generic and data model agnostic, but suffers from poor usability. The primary problem is generic representation of data services which makes it difficult for the user to define a web services client for it. The schema is not rigorous and has limited validation capabilities. Another problem is the use of 'special' keywords (not schema exposed) which have to be understood before proper use. For example for record services, the specific name 'PRODUCTID' has to be used in order to identify the primary record identifier.

The primary goal for the dynamic services is to make the web services simple and easy to use.

The dynamic services will especially be useful for record services based on an actual Repository, including attributes and information, to allow a more meaningful and syntactically tight schema. The number of such services can be different for each repository, hot deployed and redeployed if any changes are required.

The user will define a specific data repository, with all the needed attributes and will generate from this definition a repository specific web service in form of a WSDL definition and an associated implementation. This web service has six fundamental methods for record data management available (Add, Modify, Delete, Find, Validate, and Build Relationship). This newly generated web service can then be deployed onto an TIBCO MDM Server instance to be used.

Warning: The `ECMClasses.jar` is bundled with TIBCO MDM Studio. However, if TIBCO MDM server is upgraded and running on different or same machine, perform the following steps:

Set the `$MQ_HOME` on the machine where TIBCO MDM Studio is running

Set the environment variable for the `MQ_HOME`

Set the environment variable for the MQ_LIB

Copy the latest `ECMClasses.jar` file into `$MQ_HOME/lib/mq` folder where TIBCO MDM Studio is running (Only if TIBCO MDM Server is running on different machine or the local server is having a lower version of TIBCO MDM Studio than the compatible version).

Service Methods

The WSDL Generator Plug-in takes in a repository file and generates an output WSDL and corresponding java source code. The code is then compiled to a deployable web services entity (.aar) file.

The generation is offered through Wizard, which is provided as a 'New Project' wizard in the repository designer. The WSDL generation wizard collects the following inputs from the user:

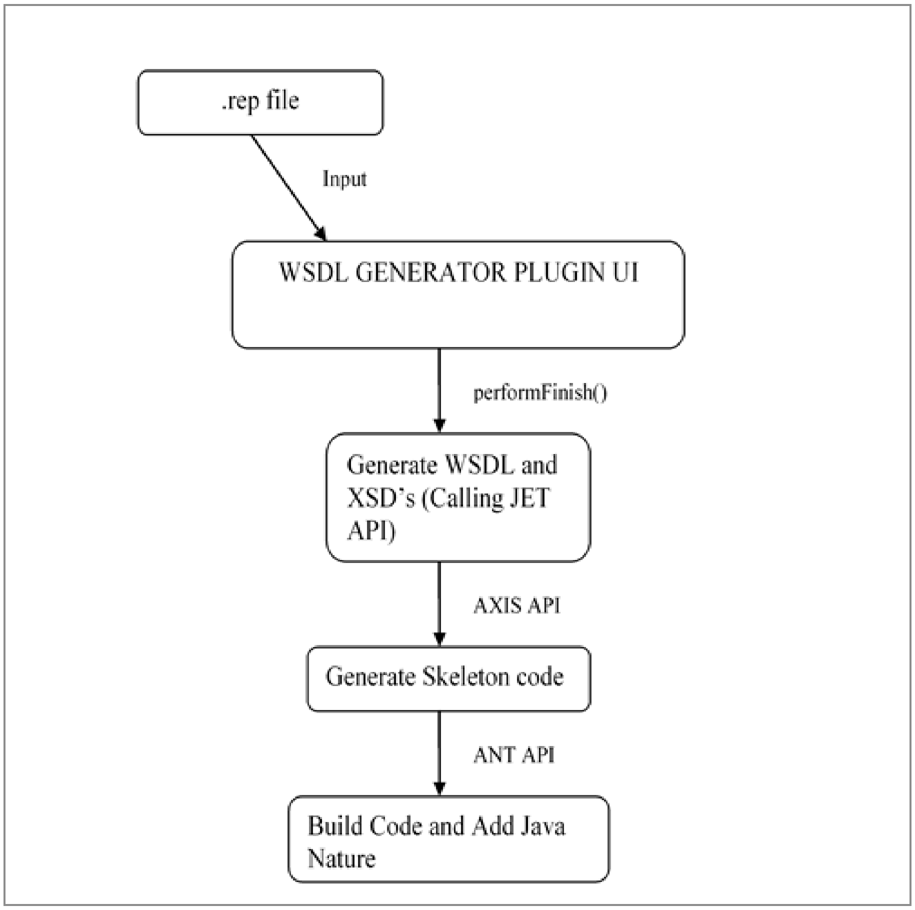
- Repository Folder name containing the Repository Models
- Repository file (*.rep)
- Repository (Repository on which you want to generate the WSDL)
- Project Name
- Selection of relationship with target repository

The WSDL should have the following properties:

- Use a single namespace for request and response
- Use the document literal flavor
- Use a SOAP fault for conditions

The schema should have the following properties:

- Define XSD types for each attribute
- Use a close mapping of XML Schema base types to TIBCO MDM base types
- Account for further type restrictions such as length
- Support multi-value attributes



Repositories with FED attributes

The implementation for performing service method operations on repositories with a Timestamp type attribute mapped to "EFFECTIVEDATE" is as follows:

- Add operation: You can add any record including future dated records to such repository.
- Update/Delete operations: You can Update and Delete all records including future effective dated records.

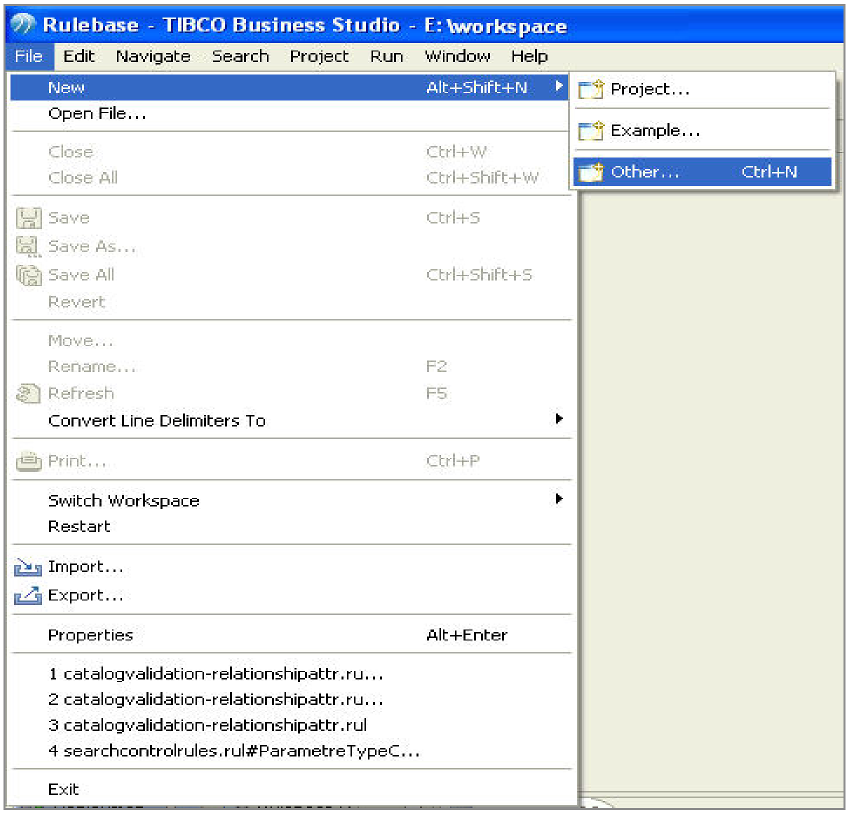
- Find operation: You can search only those records which are confirmed and already effective. The EFFECTIVEDATE attribute value should be less than current system date for Find operation to be successful. Find operation now supports options searching for FED versions only, include FED versions and exclude FED versions. The option for search is set as value for context variable 'EffectiveDateSearchOption'. The available values are, I for including FED versions, O for only FED versions, and E for excluding FED versions. If none provided FED versions are excluded from the search.

Generating WSDL

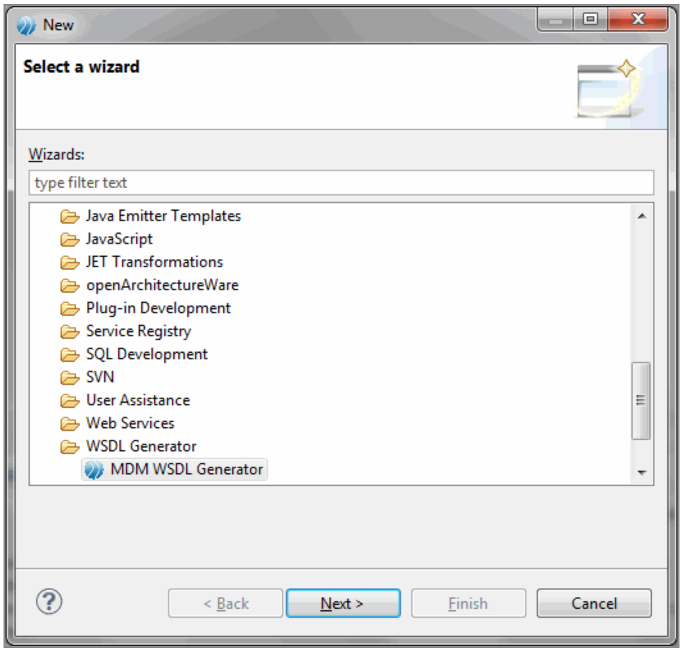
Using the eclipse wizard, you can generate the WSDL.

Procedure

1. Go to **File - > New -** and click **Other**.

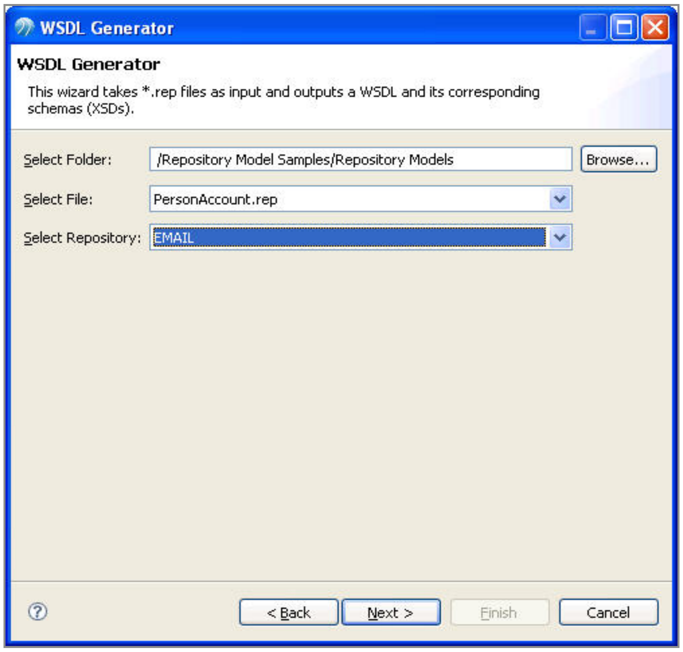


2. The **Select a wizard** screen is displayed.



3. Select **MDM WSDL Generator** from the **WSDL Generator** tree node.

4. Click **Next**. The Wizard displays the **WSDL Generator** screen.



WSDL Generator

This wizard takes *.rep files as input and outputs a WSDL and its corresponding schemas (XSDs).

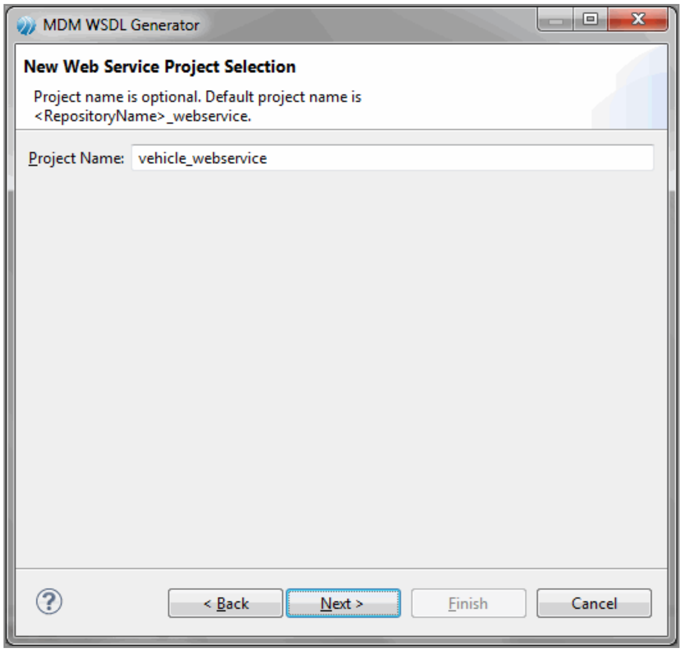
Select Folder: /Repository Model Samples/Repository Models

Select File: PersonAccount.rep

Select Repository: EMAIL

5. Click **Browse** and select the path where the input file exists. All the *.rep files are populated.

6. Select the appropriate *.rep file from the **Select File** drop-down list. Based on the selected repository file, the repositories are displayed in the **Select Repository** drop-down list.
7. Select a repository from the **Select Repository** drop-down list.
8. By default, the **Relationship** artifact is selected. You cannot deselect the relationship. The wsdl includes all the relationships in the repository.
9. Click **Next**. The **New Project** screen is displayed

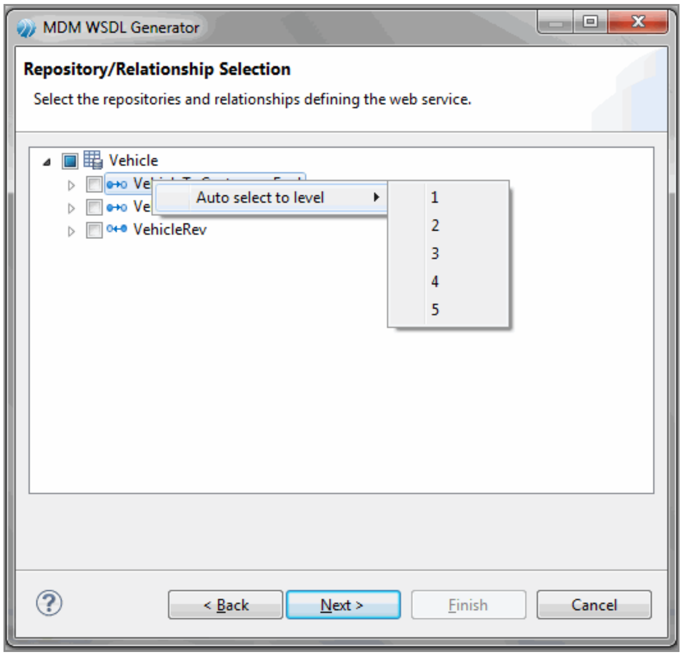


New Web Service Project Selection

Project name is optional. Default project name is <RepositoryName>_webservice.

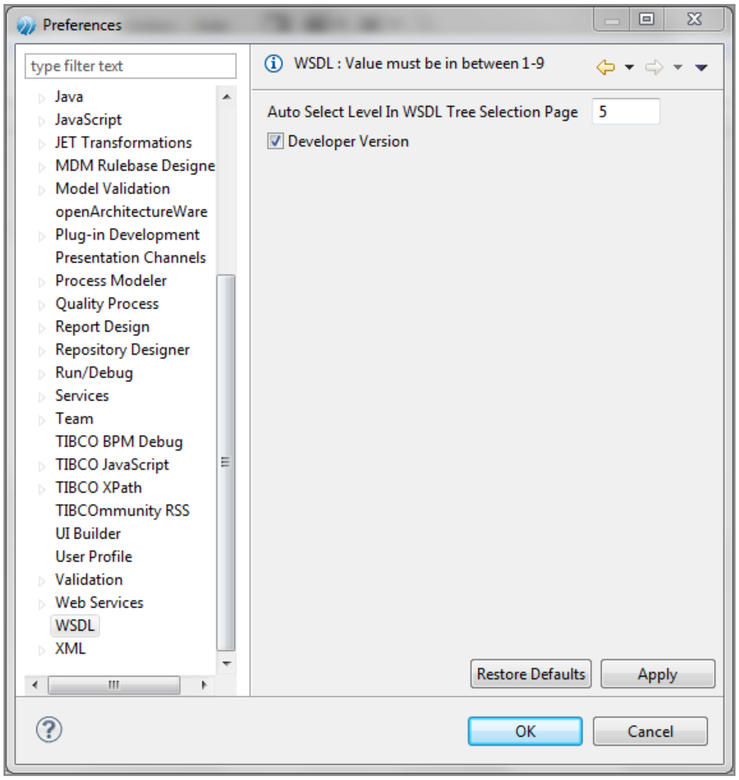
Project Name: vehicle_webservice

10. Enter the appropriate project name in the **Project Name** field. By default, the project name is <repository name_webservice.
11. Click **Next**. The **New Project** screen is displayed.


MDM WSDL Generator
Repository/Relationship Selection
Select the repositories and relationships defining the web service.
Vehicle
VehicleRev
Auto select to level
1
2
3
4
5
< Back
Next >
Finish
Cancel

12. To select the depth of the level of Repository or Relationships, right - click the relationship and point to **Auto select to level** and select the depth. For example, 3. The depth of the level is configured in the preference page. To configure the Preference perform the following.

    - From the Window, click Preference.
    - Enter the level in the **Auto Select Level in WSDL Tree Selection Page** field. The value must be between 1to 9.



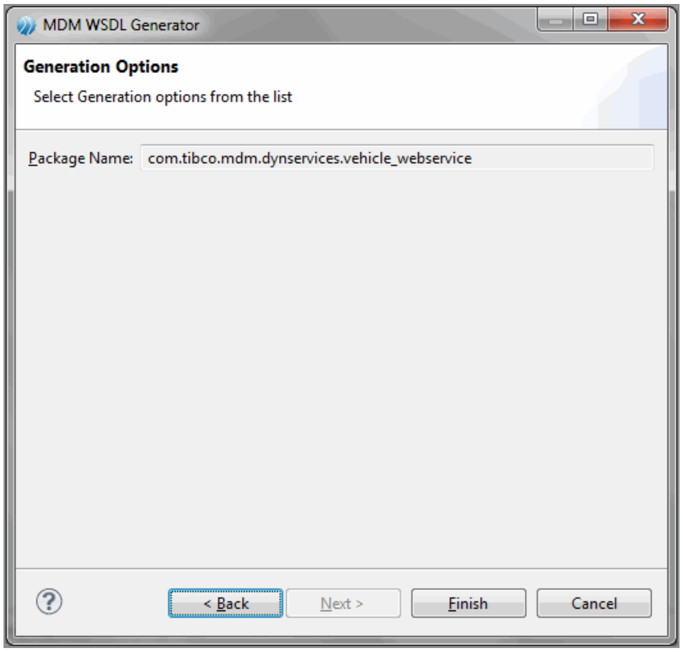
WSDL : Value must be in between 1-9

Auto Select Level In WSDL Tree Selection Page 5

☑ Developer Version

- Click **Apply**.
- Click **OK**.

13. Alternatively you can also manually select the relationship which you want and click **Next**. The **Generation Options** screen is displayed.



Package Name: com.tibco.mdm.dynservices.vehicle_webservice

14. Enter the appropriate package name in **Package Name** field. By default, a package name is provided. The package name convention is com.tibco.mdm.dynservices.<repository name>_webservice.You cannot modify the package name.

15. Click **Finish**.

> **Warning:** While generating WSDL for large repository models, TIBCO MDM Studio hangs and WSDL project gets generated with an error. To fix this issue, refer to the troubleshooting section WSDL Generation Failed due to OutOfMemoryError.

WSDL and XML Schema Generation

During the WSDL generation process, four files are created.

- WSDL file
- Request XSD file
- Response XSD file
- Generic Type file

WSDL file

The WSDL file contains the primary contract with the new service definition and the six primary operations (add, update, delete, find, validate, and build Relationship). It uses the interoperable document literal representation.

Request XSD file

The Request contains a type definition of the entire repository and for each attribute of the repository. This type should have the type mappings as described in table XSD Types, including the TIBCO MDM property length which should map for example to xsd:maxLength for a string. The three write operations (add, modify, delete) will have an entire record as input, only the read operation (find) will have a record id as input - and return the entire record as response.

Note: The types definition are defined in the repository model. If you want to change the type definition it must be changed at the repository model level.

Response XSD file

The Response file includes the request type definitions and therefore does not need to create many dynamic types. Only for the Response itself, it will create a type for all four operations. Furthermore, the ResultType and EventType is contained in there which is used by the three write operations responses.

Generic Type file

The generic type file contains the UserInfo type.

XSD Types

Type Name	Type Mapping	Type Attributes
MDM BOOLEAN	Xsd:boolean	-
`MDM Integer`	Xsd:integer	-
`MDM Decimal`	Xsd:decimal	MDM length

Type Name	Type Mapping	Type Attributes
		xsd:fractionDigits
`MDM String`	Xsd:string	MDM length, xsd:maxInclusive
`MDM File`	Xsd:string	Xsd:maxLength = 255
`MDM Amount`	Xsd:decimal	MDM length xsd:fractionDigits
`MDM Custom Decimal`	Xsd:decimal	MDM Length, Xsd:fractionDigits
`MDM Date`	Xsd:dateTime	

Service Methods

You can perform the following service methods for any repository.

- **Add Method** - This method adds a new record along with the relationship records into the TIBCO MDM repository. This supports the following:
    - Adding new Parent Record
    - Adding new child Records
- **Modify Method** - This method modifies the existing record along with the relationship records in the TIBCO MDM repository. This supports the following:
    - Modifying the Parent Record
    - Adding new child record
    - Modifying existing child record
    - Deleting existing child record
- **Delete Method** - This method deletes the record along with the relationship records from the TIBCO MDM repository. This supports the following:

    - Deleting the Parent record
    - Deleting the Parent along with all Child relationships
- **Find Method** - This method finds the record along with the relationship records in the TIBCO MDM repository. This supports the following:
    - Find All Parent records
    - Find the existing record bundle for the specified Product Id and Product ID Ext
    - Find all the relationship records for the specified parent Product ID and Product ID Ext and relationship name
    - Find a specific relationship record for specified parent and child product ID and Product ID Ext
- **Validate Method** - This method validates the record along with the relationship records in the TIBCO MDM repository. This method supports the following:
    - Validate the Record Bundle for Structural and Rule base errors
- **Build Relationship Method** - This method adds the relationship objects into the cache and it is the supporting method for add, modify and delete. The Build relationship is not recommended from Soap Client such as SOAP UI as this functionality is to improve performance in UI Builder. The Build Relationship is used for adding child relationship objects from a session into the cache. Whenever any addition or modifications (add/update/delete) are done to the parent record then the build relationship is attached to the parent object from cache to form the record bundle. The entire record bundle is saved to TIBCO MDM.
- **Copy Method** - The copy operation is currently supported only through SOAP requests using generated web services. We support only deep cloning of records. Upon successful operations the first version of the cloned record will be created and is available for view.

The following Sample depicts the service methods for the PersonAddressClient (PAC) repository model.

JSON Format

{'Envelope':{'Body':{'<servicemethodName>Request':{'UserInfo':{'UserName':'<mdm user id>','Password':'<mdmpassword>','Enterprise':'<mdm enterprise>'},

'Context':
{'Validation':'false','Process':'true','SystemAttributeReturn':'false','ReturnFileAsAttachment':'false','EffectiveDateSearchOption':'E','DeleteAllRelatedRecords':'false','PerspectiveName':'','StartCount':'1','RecordCount':20},'<RepositoryName>':{< list of repository attributes comma separated in name: value format>}}}}}

In the envelope, the <serviceMethodName> is replaced by add, update or delete followed by the repository name.

For example, addPerson/updatePerson/deletePerson.

<RepositoryName> is replaced by the repository name. For example, Person

<mdmUserid><mdmpassword><mdmEnterprise> are required for the first time.UserInfo input. It is not required for each request as these details are stored in the session from previous requests.

In the context, options are provided to apply on the record bundle.

Validation= Validates the record bundle. The return value is true or false.

Process=true/false

SystemAttributeReturn= Returns the System Attributes along with repository attributes. The return value is true or false.

ReturnFileAsAttachment= returns the file as attachment. The return value is true or false.

EffectiveDateSearchOption= Specify the value ’E’ if it is effective date search option.

DeleteAllRelatedRecords= It is used in the delete functionality for deleting all child records. The return value is true or false

PerspectiveName= Specify the perspective name

StartCount= Supports the pagination

RecordCount= Supports the pagination

Add Method

```
Request:
URL:
/eml/json/wizard/PersonService/addPerson
Content-Type:
Application/x-www-form-urlencoded
HTTP Method:
```

```
POST
Parameter Name:
JsonInput
{
  "Envelope" : {
    "Body" : {
      "AddPersonRequest" : {
        "Context" : {
          "Validation" : "false",
          "Process" : "true",
          "SystemAttributeReturn" : "false",
          "ReturnFileAsAttachment" : "false",
          "EffectiveDateSearchOption" : "E",
          "DeleteAllRelatedRecords" : "false",
          "PerspectiveName" : ""
        },
        "Person" : {
          "ID" : "99999",
          "IDEXT" : "extension-1",
          "FIRSTNAME" : "xxxxx",
          "LASTNAME" : "xxxxx"
        }
      }
    }
  }
}
```

Successful Response:

```
{
  'Envelope' : {
    'Body' : {
      'AddPersonResponse' : {
        'Result' : {
          '@code' : 'SVC-11025',
          '@severity' : 'Info',
          'Message' :
          'Service VALIDATE_PROCESS executed successfully',
          'Event' : {
            'EventId' : '61001'
          }
        }
      }
    }
  }
}
```

Failure Response:

```
{
  'Envelope' : {
    'Body' : {
      'Fault' : {
        'faultcode' : '111',
        'faultstring' : 'Cannot save record; record already exists',
        'detail' : {
          'FaultDetails' : {
            'CimErrorList' : {
              'ErrorCode' : '61001',
              'ErrorMessage' : 'Cannot save record; record already
exists.'
            }
          }
        }
      }
    }
  }
}
```

Modify/Update

```
Request:
URL:
/eml/json/wizard/PersonService/updatePerson
Content-Type:
Application/x-www-form-urlencoded
HTTP Method:
POST
Parameter Name:
JsonInput
{
  'Envelope' : {
    'Body' : {
      'updatePersonRequest' : {
        'Context' : {
          'Validation' : 'false',
          'Process' : 'true',
          'SystemAttributeReturn' : 'false',
          'ReturnFileAsAttachment' : 'false',
          'EffectiveDateSearchOption' : 'E',
          'DeleteAllRelatedRecords' : 'false',
          'PerspectiveName' : ''
        },
        'Person' : {
          'ID' : '99999',
          'IDEXT' : 'extension-1',
          'FIRSTNAME' : 'xxxxx',
          'LASTNAME' : 'xxxxx'
        }
      }
    }
  }
}
```

Success Response:

```
{
  "UpdatePersonResponse" : {
    "Result" : {
      "@code" : "SVC-11025",
      "@severity" : "Info",
      "Message" : "Service NOVALIDATE_PROCESS executed successfully.",
      "Event" : {
        "EventId" : 63025
      }
    }
  }
}
```

Failure Response:

```
{
  'Envelope' : {
    'Body' : {
      'Fault' : {
        'faultcode' : '111',
        'faultstring' : 'SVC - 11038 : Data unchanged, request ignored',
        'detail' : {
          'FaultDetails' : {
            'CimErrorList' : {
              'ErrorCode' : 'SVC - 11038 ',
              'ErrorMessage' : ' SVC - 11038 : Data unchanged, request
ignored.'
            }
          }
        }
      }
    }
  }
}
```

Delete

- Identified by ID and/or IDExt
- Exception: Record not found

```
Request:
URL:
/eml/json/wizard/PersonService/deletePerson
Content-Type:
Application/x-www-form-urlencoded
HTTP Method:
POST
Parameter Name:
JsonInput
{
  'Envelope' : {
    'Body' : {
      'DeletePersonRequest' : {
        'Context' : {
          'Validation' : 'false',
          'Process' : 'true',
          'SystemAttributeReturn' : 'false',
          'ReturnFileAsAttachment' : 'false',
          'EffectiveDateSearchOption' : 'E',
          'DeleteAllRelatedRecords' : 'false',
          'PerspectiveName' : ''
        },
        'Person' : {
          'ID' : '99999',
          'IDEXT' : 'extension-1'
        }
      }
    }
  }
}
```

Successful Response:

```
{
  'Envelope' : {
    'Body' : {
      'DeletePersonResponse' : {
        'Result' : {
          '@code' : 'SVC - 11032',
          '@severity' : 'Info',
          'Message' : 'SVC - 11032 : REASON - Record(s)deleted
successfully ',
          'Event' : {
            'EventId' : 61001
          }
        }
      }
    }
  }
}
```

Failure Response:

```
{
  'Envelope' : {
    'Body' : {
      'Fault' : {
        'faultcode' : 'CAT-1141',
        'faultstring' : 'CAT - 1141 : Record with Record ID = 12345,
Record ID Extension = not found ',
        'detail' : {
          'FaultDetails' : {
            'CimErrorList' : {
              'ErrorCode' : 'CAT-1141',
              'ErrorMessage' : 'CAT-1141: Record with Record ID=12345,
Record ID Extension= not found'
            }
          }
        }
      }
    }
  }
}
```

Find/Retrieve

- Getting a record Bundle for specific Person ID and person IDExt
- Getting all the Persons
- Getting All Addresses of a Person with id '99999' and IDExt as 'extension-1'

  ```
  Request:
  ```

```
URL:
/eml/json/wizard/PersonService/findPerson
Content-Type:
Application/x-www-form-urlencoded
HTTP Method:
POST
Parameter Name:
JsonInput
(Getting a record Bundle for specific person id and person id ext)
{
  'Envelope' : {
    'Body' : {
      'FindPersonRequest' : {
        ' Context' : {
          'Validation' : 'false',
          'Process' : 'true',
          'SystemAttributeReturn' : 'false',
          'ReturnFileAsAttachment' : 'false',
          'EffectiveDateSearchOption' : 'E',
          'DeleteAllRelatedRecords' : 'false',
          'PerspectiveName' : '',
          'StartCount' : '1',
          'RecordCount' : 20
        },
        'FindPersonID' : [{
            'ID' : '99999',
            'IDEXT' : 'extension-1'
          }
        ]
      }
    }
  }
}
Getting all the Persons
{
  'Envelope' : {
    'Body' : {
      'FindPersonRequest' : {
        ' Context' : {
          'Validation' : 'false',
          'Process' : 'true',
          'SystemAttributeReturn' : 'false',
          'ReturnFileAsAttachment' : 'false',
          'EffectiveDateSearchOption' : 'E',
          'DeleteAllRelatedRecords' : 'false',
          'PerspectiveName' : '',
          'StartCount' : '1',
          'RecordCount' : 20
        },
        'FindAllPersons' : 'true'
      }
    }
  }
}
Getting All Addresses of a Person with id '99999' and id ext as
'extension-1'
{
  'Envelope' : {
    'Body' : {
      'FindPersonRequest' : {
        ' Context' : {
          'Validation' : 'false',
          'Process' : 'true',
          'SystemAttributeReturn' : 'false',
          'ReturnFileAsAttachment' : 'false',
          'EffectiveDateSearchOption' : 'E',
          'DeleteAllRelatedRecords' : 'false',
          'PerspectiveName' : '',
          'StartCount' : '1',
          'RecordCount' : 20
        },
        'FindPersonID' : [{
            'ID' : '99999',
            'IDEXT' : 'extension-1',
            'Relationships' : {
              'PersonToAddress' : {
                'findAll' : 'true'
              }
            }
          }
        ]
      }
    }
  }
}
Response:
{
  "Envelope" : {
    "Body" : {
      "FindPersonResponse" : {
```

```
"PersonList" : {
  "Person" : {
    "ID" : "99999",
    "IDEXT" : "extension-1",
    "FIRSTNAME" : "xxxxx",
    "LASTNAME" : "xxxxx",
    "PersonToClientList" : {
      "PersonToClient" : [{
          "ParentInfo" : {
            "ParentCatalogName" : "Person",
            "ParentRecordID" : "99999",
            "ParentRecordIDEXT" : "extension-1"
          },
          "HOURLYRATE" : "100",
          "CLIENT" : {
            "ID" : "10000",
            "IDEXT" : "p",
            "CONTAINS" : "no",
            "NAME" : "",
            "DEPARTMENTNAME" : "San Fran",
            "BUSINESSTYPE" : "Retail",
            "PROJECTNAME" : "MDM",
            "DURATION" : "6"
          }
        } {
          "ParentInfo" : {
            "ParentCatalogName" : "Person",
            "ParentRecordID" : "99999",
            "ParentRecordIDEXT" : "extension-1"
          },
          "HOURLYRATE" : "100",
          "CLIENT" : {
            "ID" : "10001",
            "IDEXT" : "p",
            "CONTAINS" : "no",
            "NAME" : "",
            "DEPARTMENTNAME" : "San Fran1",
            "BUSINESSTYPE" : "Retail1",
            "PROJECTNAME" : "MDM1",
            "DURATION" : "6"
          }
        }
      ]
    }
    "PersonToAddressList" : {
```

```
              "PersonToAddress" : [{
                  "ParentInfo" : {
                    "ParentCatalogName" : "Person",
                    "ParentRecordID" : "99999",
                    "ParentRecordIDEXT" : "extension-1"
                  },
                  "Type" : "home",
                  "ADDRESS" : {
                    "ID" : "12000",
                    "IDEXT" : "p",
                    "ADDRLINE1" : "11111 san fran"
                  }
                } {
                  "ParentInfo" : {
                    "ParentCatalogName" : "Person",
                    "ParentRecordID" : "99999",
                    "ParentRecordIDEXT" : "extension-1"
                  },
                  "Type" : "office",
                  "ADDRESS" : {
                    "ID" : "12001",
                    "IDEXT" : "p",
                    "ADDRLINE1" : "2222 san fran"
                  }
                }
              ]
            }
          }
        }
      }
    }
  }
}
```

Validate Method

```
Request:
URL:
/eml/json/wizard/PersonService/validatePerson
HTTP Method:
POST
Parameter Name:
JsonInput
{
  "Envelope" : {
    "Body" : {
      "ValidatePersonRequest" : {
        "Action" : "add",
        "IncludeFailedRecordsInResponse" : "true",
        "Person" : {
          "ID" : 1,
          "IDEXT" : "p",
          "FIRSTNAME" : "Jon",
          "LASTNAME" : "S",
          "RELIGIOUS" : "Jay",
          "GENDER" : "Male",
          "HEIGHT" : 5.5,
          "WEIGHT" : 130,
          "EYECOLOR" : "Black",
          "EMAIL" : "jon@gmail.com",
          "LINKEDINID" : "joh"
        }
      }
    }
  }
}
Validation failure response:
{
  "ValidatePersonResponse" : {
    "ValidationResult" : {
      "ErrorList" : {
        "Error" : [{
            "ErrorMessage" : "'Weight' has failed validation. The
value ('130') was not present in the valid value list.",
            "RelationshipName" : "PERSON",
            "ID" : 1,
            "IDEXT" : "p",
            "AttributeName" : "WEIGHT"
          }, {
            "ErrorMessage" : "Attribute 'First Name' is transformed
from ('Jon') to ('1-NEWVAL')",
            "RelationshipName" : "PERSON",
            "ID" : 1,
            "IDEXT" : "p",
            "AttributeName" : "FIRSTNAME"
          }, {
            "ErrorMessage" : "Cannot save record; record already
exists. If you cannot find a confirmed version of this record, it
may currently be in add or delete approval process.",
            "RelationshipName" : "PERSON",
            "ID" : 1,
            "IDEXT" : "p",
            "AttributeName" : "PRODUCTID"
          }
        ]
      },
      "Person" : {
        "ID" : 1,
        "IDEXT" : "p",
        "FIRSTNAME" : "Jon",
        "LASTNAME" : "S",
        "RELIGIOUS" : "Jay",
        "GENDER" : "Male",
        "HEIGHT" : 5.5,
        "WEIGHT" : 130,
        "EYECOLOR" : "Black",
        "EMAIL" : "jon@gmail.com",
        "LINKEDINID" : "jon"
      }
    }
  }
}
Validation success response:
{
  "ValidatePersonResponse" : {
    "ValidationResult" : {
      "Info" : "Record Bundle validation is successful"
    }
  }
}
```

Build Relationship

- This creates a new relationship record of PersonToClient for person id 200 and extension tibco. The mode can be Add/Edit/Delete.

```
Request:
URL:
/eml/json/wizard/PersonService/buildRelationship
HTTP Method:
POST
Parameter Name:
JsonInput
{
  'Envelope' : {
    'Body' : {
      'RelationshipRequest' : {
        'Mode' : 'Add',
        'PersonToClient' : {
          'ParentInfo' : {
            'ParentCatalogName' : 'Person',
            'ParentRecordID' : '200',
            'ParentRecordIDEXT' : 'tibco'
          },
          'HOURLYRATE' : '100',
          'CLIENT' : {
            'ID' : '10000',
            'IDEXT' : 'p',
            'CONTAINS' : 'no',
            'NAME' : '',
            'DEPARTMENTNAME' : 'San Fran',
            'BUSINESSTYPE' : 'Retail',
            'PROJECTNAME' : 'MDM',
            'DURATION' : '6'
          }
        }
      }
    }
  }
}
Successful Response
{
  'Envelope' : {
    'Body' : {
      'RelationshipResponse' : {
        'Result' : {
          'Message' : 'Success',
          'Event' : {
            'EventId' : 'xxx'
          }
        }
      }
    }
  }
}
Failure Response:
{
  'Envelope' : {
    'Body' : {
      'Fault' : {
```

```
    'faultcode' : ' xxx',
    'faultstring' : 'xxxx' : 'Unable to add relationship into
cache',
    'detail' : {
     'FaultDetails' : {
      'CimErrorList' : {
       'ErrorCode' : 'CAT-1141',
       'ErrorMessage' : 'Unable to add relationship into
cache .'
      }
     }
    }
   }
  }
 }
}
```

Copy Method

- For Copy operation JSON sample can not be provided as it is not supported by MDM UI builder . However, an XML sample is as follows:

Request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://tibco.com/mdm/dynservices/parent_webservice1/wsdl/2.0">
   <soapenv:Header/>
   <soapenv:Body>
      <ns:CopyParentRequest>
         <!--Optional:-->
         <ns:UserInfo>
            <ns:UserName>a</ns:UserName>
            <ns:Password>a</ns:Password>
            <ns:Enterprise>party</ns:Enterprise>
         </ns:UserInfo>
         <!--Optional:-->
         <ns:Context>
            <!--You may enter the following 15 items in any order-->
            <!--Optional:-->
            <ns:Validation>true</ns:Validation>
            <!--Optional:-->
            <ns:Process>true</ns:Process>
         </ns:Context>
         <ns:Parent>
            <!--You may enter the following 13 items in any order-->
            <!--Optional:-->
            <ns:RecordIdentity>
               <ns:ID operator="eq">A3</ns:ID>
               <!--Optional:-->
               <ns:IDEXT operator="eq">333</ns:IDEXT>
            </ns:RecordIdentity>
            <!--Optional:-->
            <ns:ID>A3A</ns:ID>
            <!--Optional:-->
            <ns:IDEXT>33A</ns:IDEXT>
            <!--Optional:-->
            <ns:PtoCAList>
               <!--1 or more repetitions:-->
               <ns:PtoCA>
                  <!--You may enter the following 3 items in any order--
>
                  <ns:ParentInfo>
                     <!--You may enter the following 4 items in any
order-->
                     <ns:ParentCatalogName>PARENT</ns:ParentCatalogName>
                     <ns:ParentRecordID>A3</ns:ParentRecordID>
                     <!--Optional:-->
                     <ns:ParentRecordIDEXT>333</ns:ParentRecordIDEXT>
                  </ns:ParentInfo>
                  <!--Optional:-->
                  <ns:RecordIdentity>
                     <ns:ID operator="eq">B3</ns:ID>
                     <!--Optional:-->
                     <ns:IDEXT operator="eq">333</ns:IDEXT>
                  </ns:RecordIdentity>
                  <ns:CHILDA>
                     <!--You may enter the following 7 items in any
order-->
                     <!--Optional:-->
                     <ns:ID>B3A</ns:ID>
                     <!--Optional:-->
                     <ns:IDEXT>33A</ns:IDEXT>
                  </ns:CHILDA>
               </ns:PtoCA>
            </ns:PtoCAList>
         </ns:Parent>
      </ns:CopyParentRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

```
<soapenv:Body>
<ns2:CopyParentResponse xmlns:ns2="http://tibco.com/mdm/dynservices/parent_webservice1/wsdl/2.0">
<ns2:Result code="SVC-11025" severity="Info">
<ns2:Message>Service ''VALIDATE_PROCESS'' executed successfully.</ns2:Message>
<ns2:Event>
<ns2:EventId>76045</ns2:EventId>
</ns2:Event>
<ns2:ErrorList/>
<ns2:ValidationResult>
<ns2:RecrodDeatils>
<ns2:CatalogName>PARENT</ns2:CatalogName>
<ns2:ID>A3A</ns2:ID>
<ns2:IDEXT>33A</ns2:IDEXT>
<ns2:PIDKEY>-1</ns2:PIDKEY>
</ns2:RecrodDeatils>
<ns2:ErrorList/>
<ns2:RelationShipsList/>
</ns2:ValidationResult>
</ns2:Result>
</ns2:CopyParentResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Note:
The specified format for Date type attribute is "yyyy-mm-dd". If the format is not in the specified format then "invalid date format (<date in request xml>) with out - s at correct place" error occurs.

The ’RecordIdentity’ tag is supported mainly in update and copy operations. This tag is used in update operation to provide the identity of the record to be found in the system so that it could be mutated. The ID and IDEXT tags are used to provide the new value.

In copy operation record identity tag is used to find the record and the related record to be cloned. The ID, IDEXT tags are used to provide the identity of the cloned version.

Deployment of Services

The outcome of the generation process is a deployable entity <service name>.aar. The .aar file can be deployed into the TIBCO MDM Server.

For more information on how to deploy WSDL see Deployment

On the TIBCO MDM server, a directory named dynservices/services has been created under `MQ_HOME`. The <Repository Name>Service.aar file is deployed into the `MQ_HOME/dynservices/services` directory.

Note: When the TIBCO MDM server is running on a remote machine you must refer to a local copy of `%MQ_HOME%`.

Validation

This chapter explains validation that is performed on your repository model.

Repository Model Validation

The Repository Designer performs validation at several levels to ensure that your repository details and relationships are compliant with the TIBCO MDM format.

Display of validation errors

All validation errors are displayed in the **Problems** tab below the drawing pane.

Warnings are also displayed here but you can ignore these, since they are Business Studio specific. You only need address errors prefixed with Repository Modeler.

The errors are categorized based on:

- Description - A description of the validation error.
- Resource - The name of the Resource model that has a validation error.
- Path - The path to the model being validated.
- Type - The type of error.

Types of Validation

The following validations are performed:

Repository Level Validation

ID and IDEXT Attributes

Repositories must have a single ID and IDEXT attribute. These are created by default when you add a new repository to your design, and they should not be deleted. If you attempt to add more than one ID or IDEXT attribute per repository, you will get a validation error.

Valid Repository Name

All Repositories should be named, and the name cannot contain spaces. Repository Name can not contain invalid characters. It allows only alphanumeric character, _*.-@<>/\ characters, characters from other languages and case insensitive.

Unique Repository Name

Repository names should be unique within a repository model (.rep file).

Valid Table Name

Table name cannot contain spaces. Table Name cannot contain invalid characters. It allows only alphanumeric character and $#_ characters.

Attribute Level Validation

Valid and Unique Attribute Names

Attribute names cannot be blank, contain any spaces, or use any reserved names. Each attribute in a repository must have a unique name.

If you provide the same name to two attributes, you will get a validation error. Attribute Name can not contain invalid characters.

It allows only alphanumeric character and _ characters and are case insensitive. It should contain less than 30 characters.

Attribute Reserved Names

Some of reserved names are "STATE","MODVERSION","MODDATE", "MODMEMBERID", "CATALOGVERSIONNUMBER","CHECKSUM","PRODUCTKEYID", "LASTIMPORTTIME","ACTIVE","ENDEFFECTIVEDATE","CATALOGID", "BATCHID","PROCESSLOGID","CREATIONDATE","LASTCONFIRMEDVERSION", "OWNERID","OWNERTYPE","PARENTVERSION"

Attribute Positions

If there is one attribute where the attribute property position is set to a valid non-null value then all of the attribute must have the attribute property set to a valid value.

Attribute Column Positions

The column position for an attribute cannot be negative or zero. Define a positive number as column position.

Attribute Column Name

Column name for attribute should contain less than 28 characters.

Column name for Multi- Value attributes should be less than 26 characters.

Column Name for attributes can not contain invalid characters and white space. It allows only alphanumeric character and $#_ characters.

String Attribute Length

Length for attributes of type string cannot be negative (smaller than zero) or more than 4000 characters.

Attribute Group Name

Attribute Group Names cannot contain spaces. Attribute group name should contain less than or equal to 30 characters.

Effective Date Attribute

There cannot be more than one attribute as Effective date. The Effective date must of “TIMESTAMP” type.

Input Map

Input Map name in repository cannot be Undefined.

Input Map should have at least one common column in Input Map for joining Data Source.

Input Map cannot contain spaces.

Input Map name and description cannot contain more than 80 characters.

Input Map descriptions cannot be blank.

Relationship Validation

Reverse Relationship Name

You must provide a reverse relationship name and description.

No spaces in Relationship and Reverse Relationship Name

The relationship name and the reverse relationship name cannot contain any spaces.

Unique Relationship and Reverse Relationship Names

The relationship name cannot be the same as the reverse relationship name.

Unique Duplicate Relationship and Reverse Relationship Names

The duplicate relationship name cannot be the same as the duplicate reverse relationship name.

Relationship Attribute and Relationship Attribute Group Validation

Relationship Attribute Group

A Relationship Attribute Group must contain a relationship attribute, it cannot be empty.

Relationship Attribute Group connection to relationship

A Relationship Attribute Group once created must be connected to a relationship using a connector.

Single connector between Relationship and Relationship Attribute Group

Only a single connector should be used between a Relationship Attribute Group and a relationship.

Datasource Validation

Valid and Unique Datasource Names

Datasource name cannot be blank, contain any spaces.

Each Datasource name must have a unique name.

Length of Datasource name cannot be more than 80 characters.

If you provide the same name to two Datasource, an validation error occurs.

It allows only alphanumeric character and _ characters.

New Datasource will be created.

Datasource name already exists and will be modified.

Valid Datasource Description

Datasource description cannot be empty.

Valid Datasource Delimiter character

Datasource delimiter character cannot be empty.

Valid and Unique Datasource Attribute Names

Datasource Attribute name cannot be blank, contain any spaces.

Each datasource Attributes name must have a unique name.

Attribute name for datasource should contain less than 28 characters.

If you provide the same name, an validation error occurs.

It allows only alphanumeric character and _ characters and are case insensitive.

Valid Datasource Attribute Description

Datasource description cannot be empty.

Datasource Attribute Position

Datasource Attribute Position maximum value cannot be more than 999 minimum value cannot be a negative or zero.

Datasource Attributes Limit

The maximum number of attributes in a Data Source is 999.

Datasource Attribute Length

Datasource attribute length cannot be a invalid Length.

The length specified for Data Source Attribute in Data Source must be a number.

The Length for attribute of type 'DECIMAL' must be in between '7' and '32' or set to 0.

The Length for attribute of type 'CUSTOM_DECIMAL' must be in between '1' and '29'.

The datasource attribute length cannot be negative.

The Length for attribute of type 'STRING' must be in between '0' and '4000'.

The Length for attribute of type 'BOOLEAN' must be '5'.

Output Map Validation

Valid Output Map Name

Valid and Unique Output Map Names.

Output map names cannot be blank, contain any spaces.

Each output map name in a repository must have a unique name.

Length of output map name cannot be more than 80 characters.

If you provide the same name to two output maps, an validation error occurs.

Valid Output Map Description

Output map description cannot be blank. Length of output map description cannot be more than 80 characters.

Valid Output Map Repository

Repository field cannot be empty. You must select repository.

New Output Map will be added to the repository.

Output Map already exists in repository and will be modified.

Valid Output Map sync Format

Sync Format field cannot be empty. You must select Sync Format.

Synchronization Format Validation

Valid Sync Format Name

Sync Format name cannot be blank, contain any spaces.

Each Sync Format name must have a unique name.

Length of Sync Format name cannot be more than 80 characters.

If you provide the same name to two Sync Format, an validation error occurs.

New Synchronization Format will be created.

Synchronization Format already exits and will be modified.

Catalog Type with name already exists. Provide different name.

Synchronization Format table with name already exists.

Synchronization Format cannot be modified as Synchronization has already been performed.

Valid Sync Format Description

Sync Format description cannot be blank.

Length of Sync Format description cannot be more than 80 characters.

Valid Sync Format Delimiter

Backslash character (\) constant is not allowed as Delimiter.

Delimiter cannot be empty.

Category Specific Attributes Validation

Category Code

The Category Code for a category name cannot be undefined.

The Category Name should have a category code.

The Category Code must be unique. The Category Code must not exceed 245 characters.

Category Name

The Category Name under parent category cannot be Undefined.

The Category Name must be entered and it cannot contain spaces.

The Category Name cannot have invalid character, it supports only alphanumeric and should not exceed 245 characters.

The Category Name must be unique and is case insensitive.

Classification Scheme Name

The Classification Scheme Name cannot be Undefined.

You must provide a classification scheme name.

The length of classification scheme name must not exceed 78 characters.

The Duplicate Classification name is not allowed for particular repository.

The Classification Name cannot have white spaces.

The Classification Scheme description cannot be blank.

Database Schema Objects Import

You can design a repository model by importing a database schema objects (table or Foreign Key constraints) in the TIBCO MDM Studio.

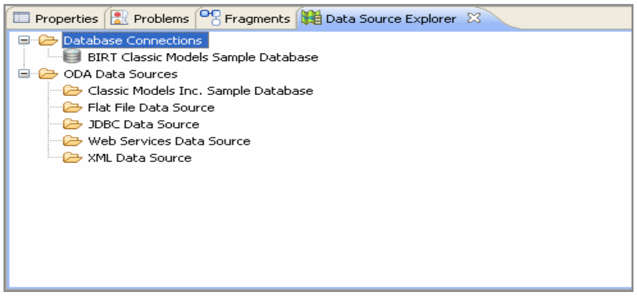
Creating a New Database Connection profile

A connection profile contains the connection property information needed to connect to a database runtime instance.

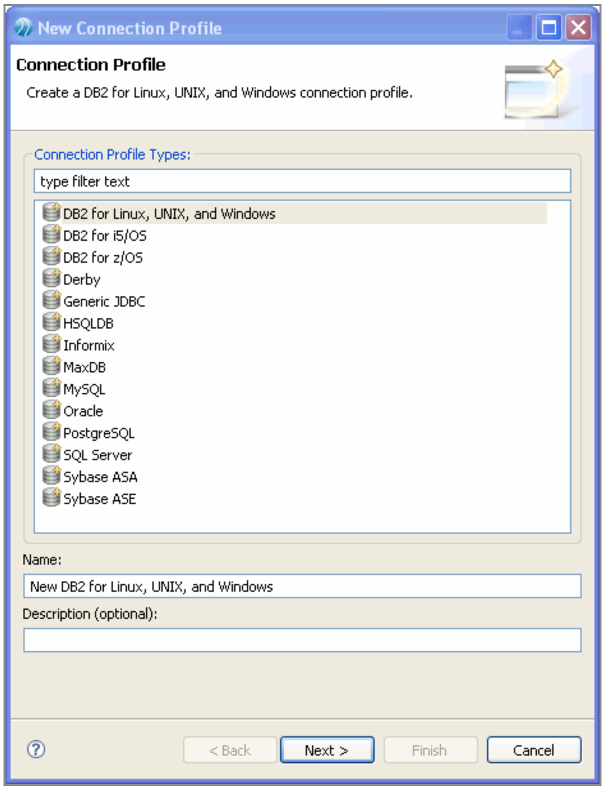
> **Warning:** Ensure that the database server is running before attempting to test the connection or to connect to the database server.

Procedure

1. In **Data Source Explorer**, Right-click on **Database Connections** and select **New**.



2. The new connection profile dialog box is displayed.



New Connection Profile

Connection Profile

Create a DB2 for Linux, UNIX, and Windows connection profile.

Connection Profile Types:

type filter text

- DB2 for Linux, UNIX, and Windows
- DB2 for i5/OS
- DB2 for z/OS
- Derby
- Generic JDBC
- HSQLDB
- Informix
- MaxDB
- MySQL
- Oracle
- PostgreSQL
- SQL Server
- Sybase ASA
- Sybase ASE

Name: New DB2 for Linux, UNIX, and Windows

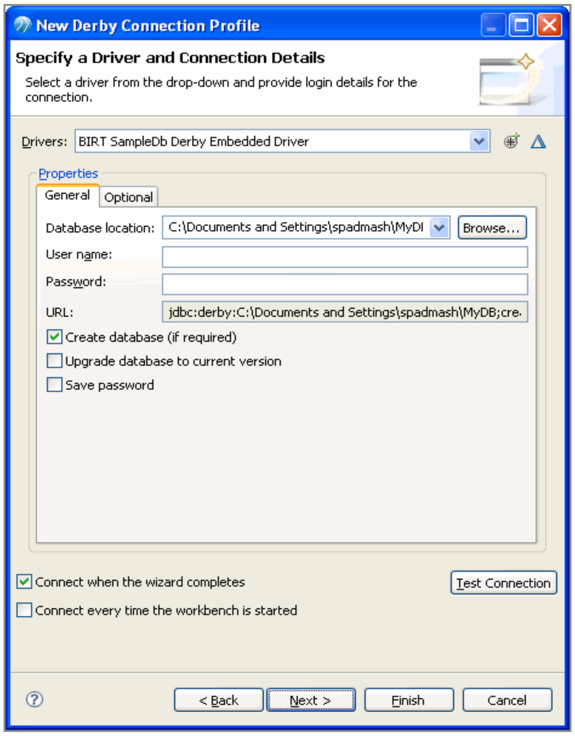
Description (optional):

3. Select the database from the filter type list for which you want to create a connection profile. For example, to list all databases that begin with the letter D, enter a d in the text box. You can also enter a text string such as JDBC or embedded to filter the list

4. Change the Name from the system-provided default, and enter a Description. You must change the name if a connection profile already exists with the same name.

5. Click **Next**.

6. The driver details dialog box for the selected database if displayed.



New Derby Connection Profile

Specify a Driver and Connection Details

Select a driver from the drop-down and provide login details for the connection.

Drivers: BIRT SampleDb Derby Embedded Driver

Properties

General | Optional

Database location: C:\Documents and Settings\spadmash\MyDI

User name:

Password:

URL: jdbc:derby:C:\Documents and Settings\spadmash\MyDB;cre

☑ Create database (if required)

□ Upgrade database to current version

□ Save password

☑ Connect when the wizard completes

□ Connect every time the workbench is started

7. Select the appropriate driver from the drop-down list. If a driver definition does not exist for the database type selected, you can create one. Click ⊕ to create a new driver definition. Select △ to edit an existing driver definition. .

8. The Properties tab displays the default template information for the selected driver. Add or modify the Properties information in the **General** or **Optional** Properties tabs.

   - Select the **Save Password** checkbox to save the password for the database server login.

   - Select **Connect When the Wizard Completes** checkbox to connect to the database server once you finish defining this connection profile.

   - Select **Connect Every Time the Workbench Is Started** checkbox to automatically connect to the database server using this connection profile when you launch Eclipse.

   - Click **Test Connection** to ping the server and to verify that the connection profile is working.

9. If you want to create database select the **Create database** checkbox and if you want to upgrade your database then select the **Upgrade database to current version** checkbox.

10. Click **Next** to view Summary information.
11. Click **Finish** to create the connection profile.

Importing Model from Database Objects

You can design a repository model by importing a database schema objects (table or Foreign Key constraints) in the TIBCO MDM Studio.

Note: To use Database Schema Import, you must have a database connection. For more information on creating a new database connection refer to Creating a New Database Connection profile.

For each selected table/view, repository would be generated and table column definitions are mapped as repository attribute. Foreign key constraints are mapped as relationship between the repositories. For example, the Table/View name is mapped to Repository Name and the Column Name is mapped to Attribute Name.

Schema Object	MDM Mapping
Table/View	Repository Name
Column	Attribute
Column Name	Attribute Name
Column Type	Attribute Type. For more information, refer to MDM data type Mapping
Column Length	Attribute Length
Foreign key constraint	Relationship

The Foreign key becomes the relationship between repositories (Selected table is source repository and R_TABLE is target repository). For repository attribute, Oracle and SQL server data types are mapped as:

MDM data type Mapping

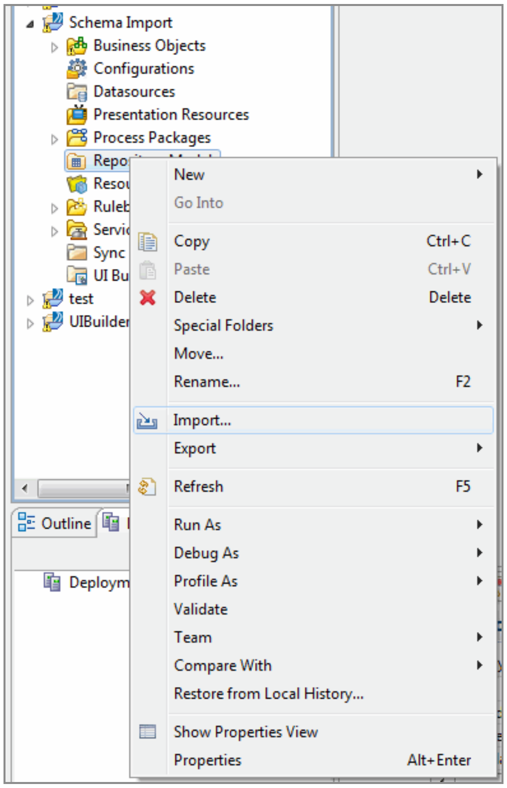
MDM Data Type	Oracle data type	SQL Server data type
String	VARCHAR2	nvarchar, varchar, sysname
Decimal	NUMBER	numeric
Integer	INTEGER	Int, tinyint, smallint, bigint
Date	DATE	datetime2
Timestamp	TIMESTAMP, TIMESTAMP(6)	datetime
File	BLOB, CLOB	varbinary
Boolean	BIT	bit

Importing a Schema Objects

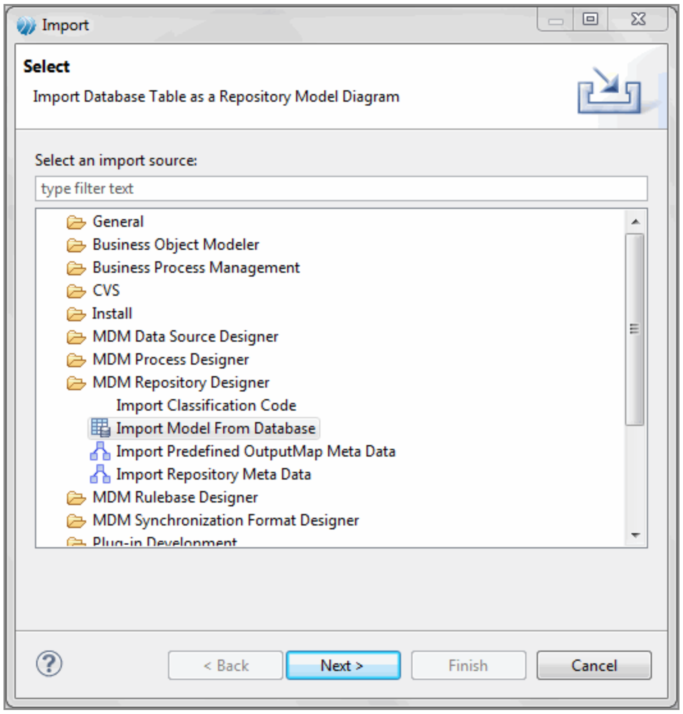
You can import the schema objects from the database connection you have in Data Source explorer.

Procedure

1. In the Project Explorer, Right click on the Repository Models folder to which you want to import and select **Import**.



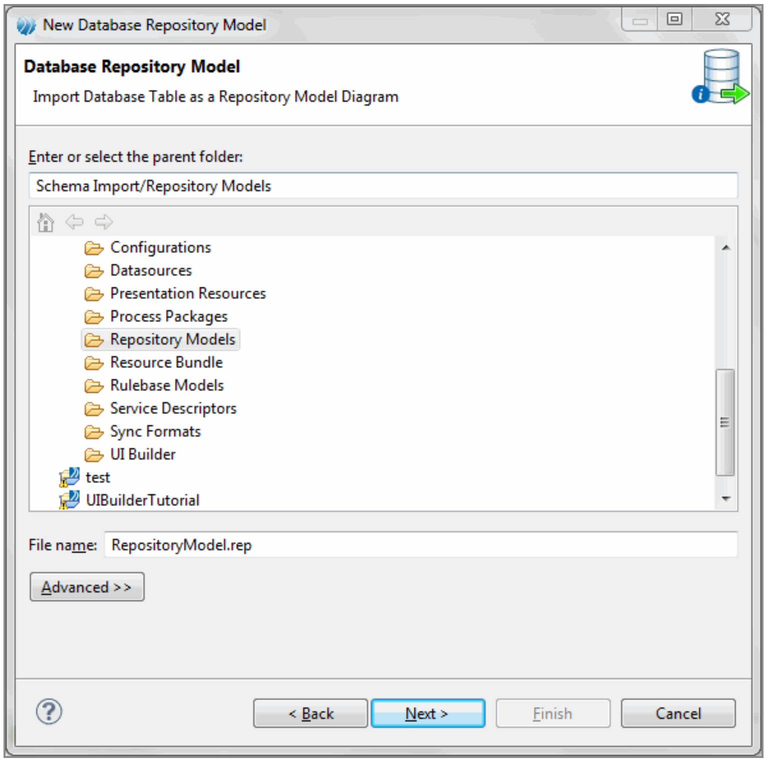
2. The Import Wizard is displayed.



Import
Select
Import Database Table as a Repository Model Diagram
Select an import source:
type filter text
General
Business Object Modeler
Business Process Management
CVS
Install
MDM Data Source Designer
MDM Process Designer
MDM Repository Designer
Import Classification Code
Import Model From Database
Import Predefined OutputMap Meta Data
Import Repository Meta Data
MDM Rulebase Designer
MDM Synchronization Format Designer
Plug-in Development
< Back
Next >
Finish
Cancel

3. Expand **MDM Repository Designer** folder, select **Import Model From Database** and click **Next**.

4. The New Database Repository Model page is displayed.



New Database Repository Model

Database Repository Model

Import Database Table as a Repository Model Diagram

Enter or select the parent folder:

Schema Import/Repository Models

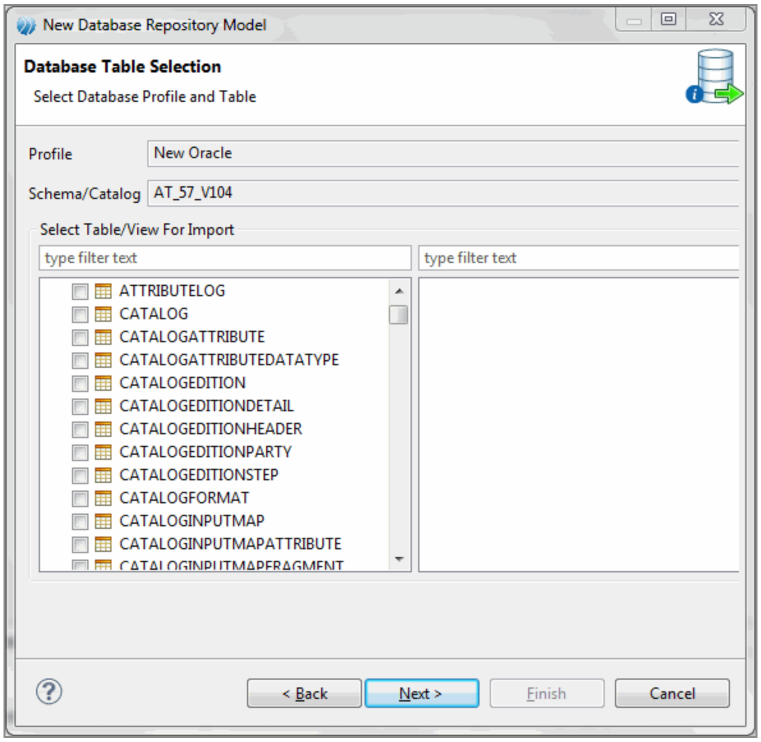
- Configurations
- Datasources
- Presentation Resources
- Process Packages
- Repository Models
- Resource Bundle
- Rulebase Models
- Service Descriptors
- Sync Formats
- UI Builder
- test
- UIBuilderTutorial

File name: RepositoryModel.rep

5. By default, the model name is displayed as **RepositoryModel.rep** in **File name** field. You can specify a different model name (*.rep) or choose the default name and click **Next**.

6. The **Database Table Selection** page is displayed. Select the profile name from the **Profile** drop-down list from which you want to select the schema.

7. Select the schema or catalog from the **Schema/Catalog** drop-down. For the selected schema, all the Tables ▦ and Views ▤ are populated on the left hand side of the table. There are checkboxes corresponding to the table or view.



New Database Repository Model

Database Table Selection

Select Database Profile and Table

Profile: New Oracle

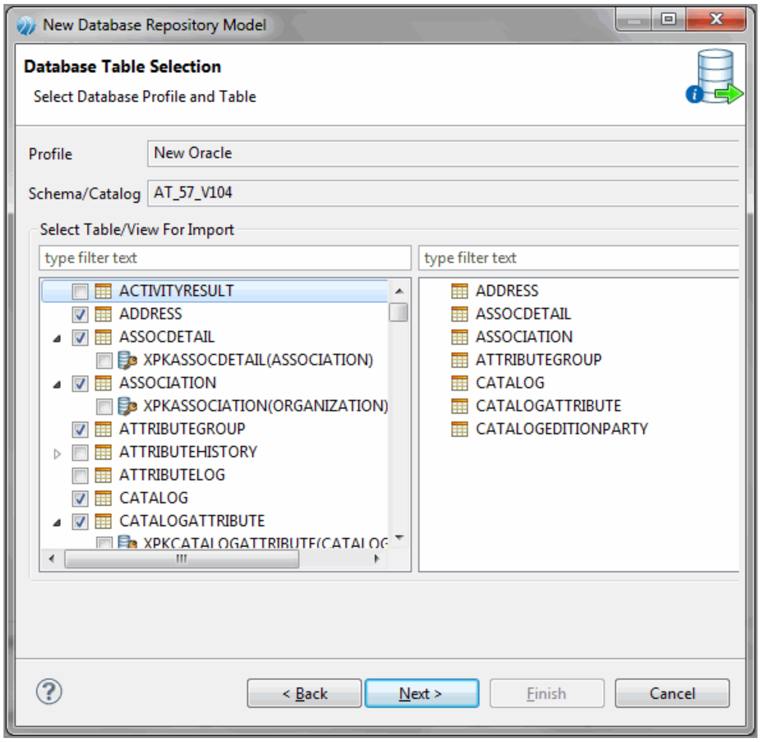
Schema/Catalog: AT_57_V104

Select Table/View For Import

type filter text | type filter text

- □ ATTRIBUTELOG
- □ CATALOG
- □ CATALOGATTRIBUTE
- □ CATALOGATTRIBUTEDATATYPE
- □ CATALOGEDITION
- □ CATALOGEDITIONDETAIL
- □ CATALOGEDITIONHEADER
- □ CATALOGEDITIONPARTY
- □ CATALOGEDITIONSTEP
- □ CATALOGFORMAT
- □ CATALOGINPUTMAP
- □ CATALOGINPUTMAPATTRIBUTE
- □ CATALOGINPUTMAPFRAGMENT

8. Select the checkboxes corresponding to the table or view which you want to include in the repository model and then, right hand side of the table displays the selected table or view. You can use the **type filter text** to search for the Table/View. If you want to import Foreign key constraints as relationship for a particular table or view then expand the table or view and select the checkbox corresponding to that constraint.



New Database Repository Model

Database Table Selection

Select Database Profile and Table

Profile: New Oracle

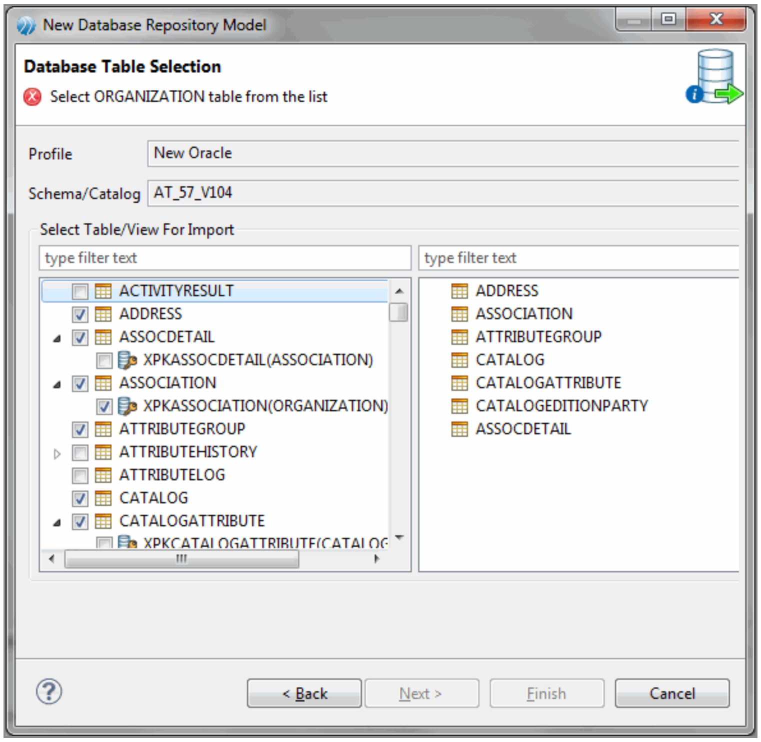
Schema/Catalog: AT_57_V104

Select Table/View For Import

- ☐ ACTIVITYRESULT
- ☑ ADDRESS
- ☑ ASSOCDETAIL
  - ☐ XPKASSOCDETAIL(ASSOCIATION)
- ☑ ASSOCIATION
  - ☐ XPKASSOCIATION(ORGANIZATION)
- ☑ ATTRIBUTEGROUP
- ☐ ATTRIBUTEHISTORY
- ☐ ATTRIBUTELOG
- ☑ CATALOG
- ☑ CATALOGATTRIBUTE
  - ☐ XPKCATALOGATTRIBUTE(CATALOG

ADDRESS
ASSOCDETAIL
ASSOCIATION
ATTRIBUTEGROUP
CATALOG
CATALOGATTRIBUTE
CATALOGEDITIONPARTY

9. If you have selected a foreign key of a table and the target table (which is mandatory to make a relationship) is not selected then an error is thrown prompting to select the target Table.



New Database Repository Model

Database Table Selection

Select ORGANIZATION table from the list

Profile: New Oracle

Schema/Catalog: AT_57_V104

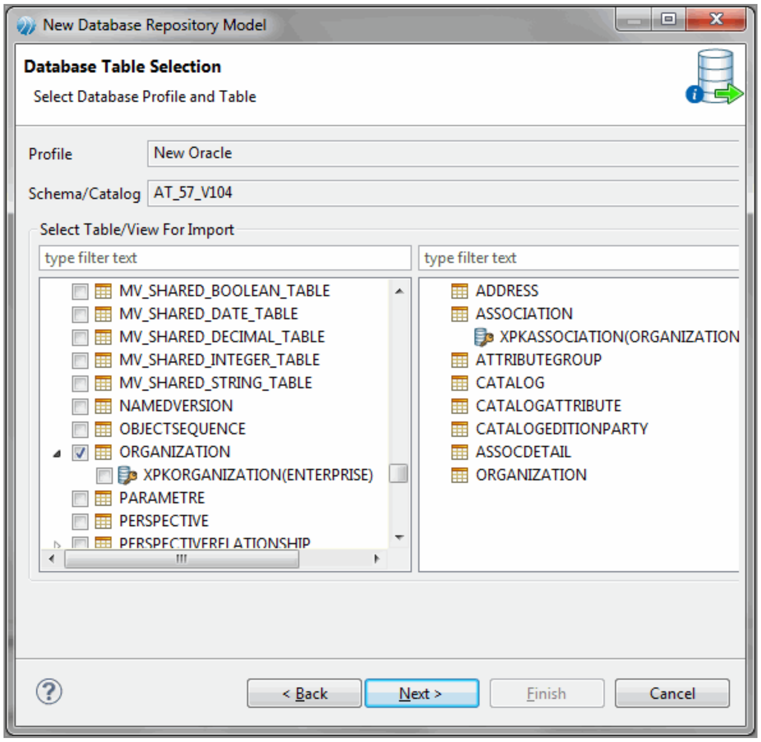
Select Table/View For Import

- □ ACTIVITYRESULT
- ☑ ADDRESS
- ☑ ASSOCDETAIL
  - □ XPKASSOCDETAIL(ASSOCIATION)
- ☑ ASSOCIATION
  - ☑ XPKASSOCIATION(ORGANIZATION)
- ☑ ATTRIBUTEGROUP
- □ ATTRIBUTEHISTORY
- □ ATTRIBUTELOG
- ☑ CATALOG
- ☑ CATALOGATTRIBUTE
  - □ XPKCATALOGATTRIBUTE(CATALOG

Selected:

- ADDRESS
- ASSOCIATION
- ATTRIBUTEGROUP
- CATALOG
- CATALOGATTRIBUTE
- CATALOGEDITIONPARTY
- ASSOCDETAIL

10. Select the target table or view.



New Database Repository Model

Database Table Selection

Select Database Profile and Table

Profile: New Oracle

Schema/Catalog: AT_57_V104

Select Table/View For Import

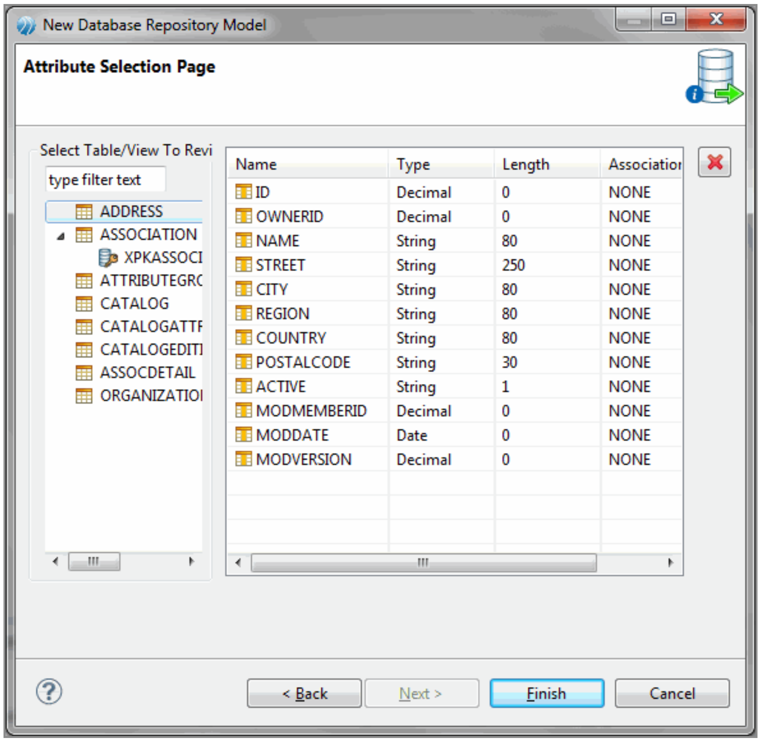
type filter text

- □ MV_SHARED_BOOLEAN_TABLE
- □ MV_SHARED_DATE_TABLE
- □ MV_SHARED_DECIMAL_TABLE
- □ MV_SHARED_INTEGER_TABLE
- □ MV_SHARED_STRING_TABLE
- □ NAMEDVERSION
- □ OBJECTSEQUENCE
- ☑ ORGANIZATION
  - □ XPKORGANIZATION(ENTERPRISE)
- □ PARAMETRE
- □ PERSPECTIVE
- □ PERSPECTIVERELATIONSHIP

type filter text

- ADDRESS
- ASSOCIATION
  - XPKASSOCIATION(ORGANIZATION
- ATTRIBUTEGROUP
- CATALOG
- CATALOGATTRIBUTE
- CATALOGEDITIONPARTY
- ASSOCDETAIL
- ORGANIZATION

11. Click **Next**. The **Attribute Selection Page** is displayed.

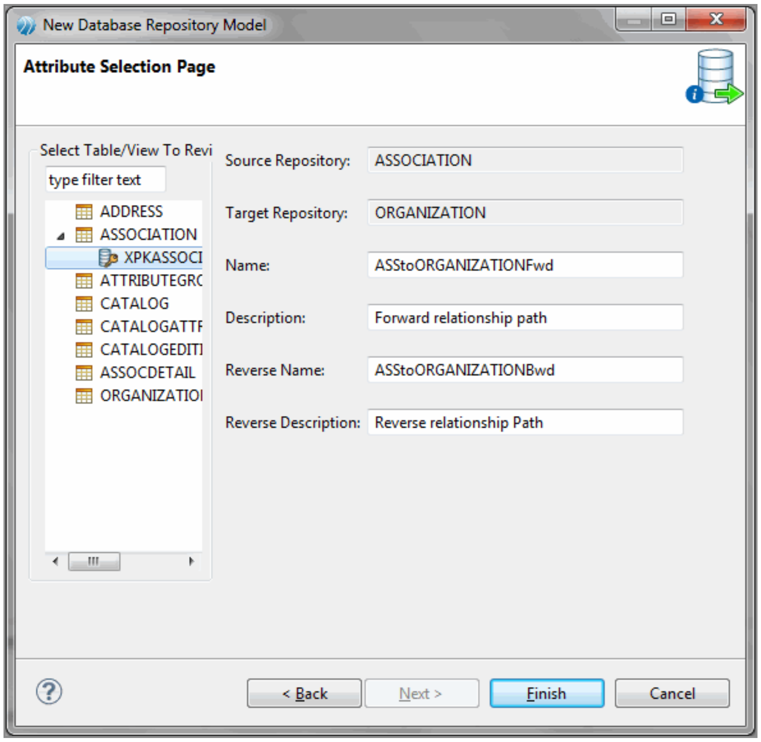


Name	Type	Length	Associatio
ID	Decimal	0	NONE
OWNERID	Decimal	0	NONE
NAME	String	80	NONE
STREET	String	250	NONE
CITY	String	80	NONE
REGION	String	80	NONE
COUNTRY	String	80	NONE
POSTALCODE	String	30	NONE
ACTIVE	String	1	NONE
MODMEMBERID	Decimal	0	NONE
MODDATE	Date	0	NONE
MODVERSION	Decimal	0	NONE

12. The repository attribute list with the default Name, Type, and Length values is displayed. You can update values from all available columns for an attribute and you can also use Association map to choose any attribute as ID (Record Id), IDEXT (Record ID Extension), or EFFECTIVEDATE.

13. To delete any attribute, select the attribute and click ☒. To delete more than one attribute, hold the CTRL button select the attributes and click ☒.

14. Similarly you can modify the relationship properties by selecting foreign key node.



Attribute Selection Page

Source Repository: ASSOCIATION

Target Repository: ORGANIZATION

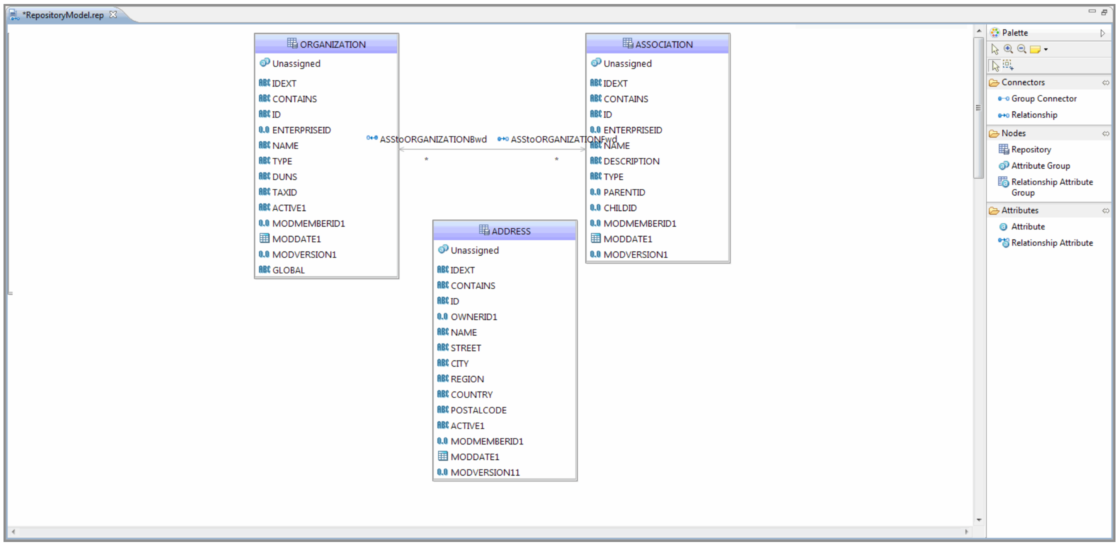
Name: ASStoORGANIZATIONFwd

Description: Forward relationship path

Reverse Name: ASStoORGANIZATIONBwd

Reverse Description: Reverse relationship Path

15. Click **Finish**. The Repository Model is created and displayed in the canvas.



Repository Model in Excel Format

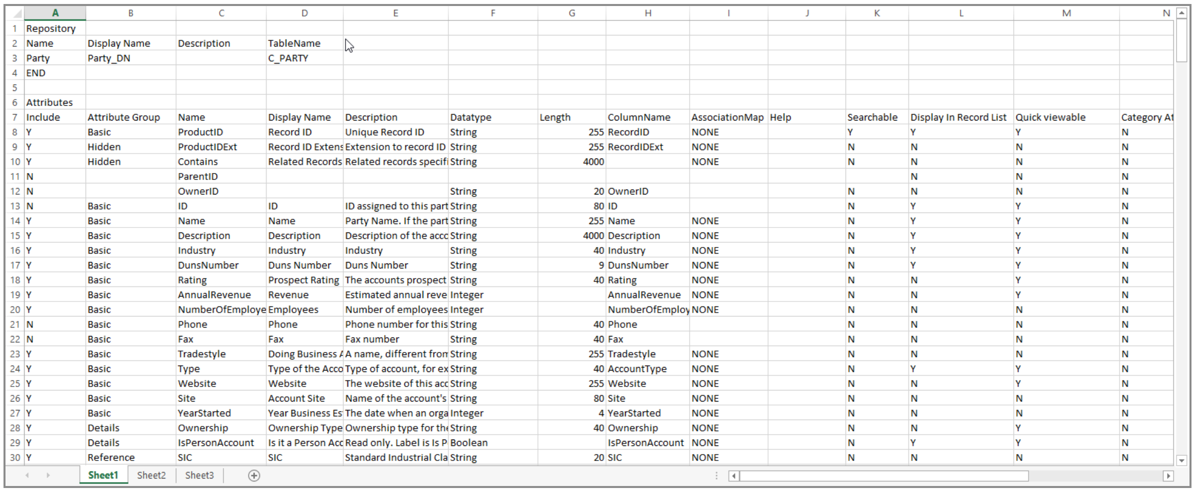
Repository Model can be designed by importing an Excel sheet with predefined format for a repository, attributes, relationships, and relationship attributes. The excel format describing the format is explained in this chapter.

Note: You can find the sample excel file `MDM Excel File Format.zip` in the TIBCO MDM Studio <version> Samples folder.

Repository Excel File Format

To import excel file into the repository model, you need to follow specified file formats and editing rules.

The sequence of the sections must be 'Repository', 'Attributes', 'Relationship' and 'RelationshipAttributes'. Each section must be terminated by specifying "END" in first column. The order of the predefined columns must also be same according to each section.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Repository													
2	Name	Display Name	Description	TableName										
3	Party	Party_DN		C_PARTY										
4	END													
5														
6	Attributes													
7	Include	Attribute Group	Name	Display Name	Description	Datatype	Length	ColumnName	AssociationMap	Help	Searchable	Display In Record List	Quick viewable	Category At
8	Y	Basic	ProductID	Record ID	Unique Record ID	String	255	RecordID	NONE		Y	Y	Y	N
9	Y	Hidden	ProductIDExt	Record ID Extens	Extension to record ID	String	255	RecordIDExt	NONE		N	N	N	N
10	Y	Hidden	Contains	Related Records	Related records specifi	String	4000		NONE		N	N	N	N
11	N		ParentID									N	N	N
12	N		OwnerID			String	20	OwnerID			N	N	N	N
13	N	Basic	ID	ID	ID assigned to this part	String	80	ID			N	Y	Y	N
14	Y	Basic	Name	Name	Party Name. If the part	String	255	Name	NONE		N	Y	Y	N
15	Y	Basic	Description	Description	Description of the accc	String	4000	Description	NONE		N	Y	Y	N
16	Y	Basic	Industry	Industry	Industry	String	40	Industry	NONE		N	Y	Y	N
17	Y	Basic	DunsNumber	Duns Number	Duns Number	String	9	DunsNumber	NONE		N	Y	Y	N
18	Y	Basic	Rating	Prospect Rating	The accounts prospect	String	40	Rating	NONE		N	N	Y	N
19	Y	Basic	AnnualRevenue	Revenue	Estimated annual reve	Integer		AnnualRevenue	NONE		N	N	Y	N
20	Y	Basic	NumberOfEmploye	Employees	Number of employees	Integer		NumberOfEmploy	NONE		N	N	N	N
21	N	Basic	Phone	Phone	Phone number for this	String	40	Phone			N	N	N	N
22	N	Basic	Fax	Fax	Fax number	String	40	Fax			N	N	N	N
23	Y	Basic	Tradestyle	Doing Business A	A name, different from	String	255	Tradestyle	NONE		N	N	N	N
24	Y	Basic	Type	Type of the Acco	Type of account, for ex	String	40	AccountType	NONE		N	Y	Y	N
25	Y	Basic	Website	Website	The website of this acc	String	255	Website	NONE		N	N	Y	N
26	Y	Basic	Site	Account Site	Name of the account's	String	80	Site	NONE		N	N	N	N
27	Y	Basic	YearStarted	Year Business Es	The date when an orga	Integer	4	YearStarted	NONE		N	N	N	N
28	Y	Details	Ownership	Ownership Type	Ownership type for the	String	40	Ownership	NONE		N	N	Y	N
29	Y	Details	IsPersonAccount	Is it a Person Acc	Read only. Label is Is P	Boolean		IsPersonAccount	NONE		N	Y	Y	N
30	Y	Reference	SIC	SIC	Standard Industrial Cla	String	20	SIC	NONE		N	N	N	N

Repository Section

- This section starts with the text 'Repository' in first column.

- The **Name** field is mandatory whereas the **Display Name**, **Description** and the **Table Name** are optional fields.
- Columns A-D must be used to specify repository fields.
- End the section with the text 'END' in the first column

Attribute Section

- This section starts with 'Attributes' in the first column.
- The Attribute Group and Name fields are mandatory, if left blank error occurs.
- Columns A-S must be used to specify attribute fields.
- The **Display Name**, **Description**, **Column Name**, **Help**, **AssociationMap**, **MultiValued** table names are optional fields.
- The valid values for the Searchable, Multivalued, Use Common Table, Display in Record List, Quick Viewable, Category Attribute, and Partition key are either 'Y' or 'N'.
- The **Data Type** field is mandatory and must contain valid MDM data types.
- The **Length** field is optional, however, it is recommended to specify length for each attribute. The length must be provided as a numeric value.
- The **Position** field is optional. If specified, all the attributes in the repository must have their position specified. The position must be provided as a numeric value.
- End the section with the text 'END' in the first column.
- The **Include** column indicates whether the row should be skipped or not. Specify 'Y' if you do not want to skip this row or specify 'N' to skip the row.

Relationship Section

- This section starts with 'Relationships' in the first column.
- Columns A-F must be used to specify relationship fields.
- The **Name**, **Description**, **Reverse Name** and **Reverse Description** fields are optional. If fields are not provided then it assigns default names and descriptions with respect to target repository name. For example, if the Target Repository Name is **Contact** then the default relationship names are, **Name - ContactFwd** and **Reverse Name - ContactBwd**.
- The **Target Repository Name** field is mandatory and must contain valid Repository Name.

- End section with the text 'END' in the first column.
- The **Include** column indicates whether the row should be skipped or not. Specify 'Y' if you do not want to skip this row or specify 'N' to skip the row.

Relationship Attribute Section

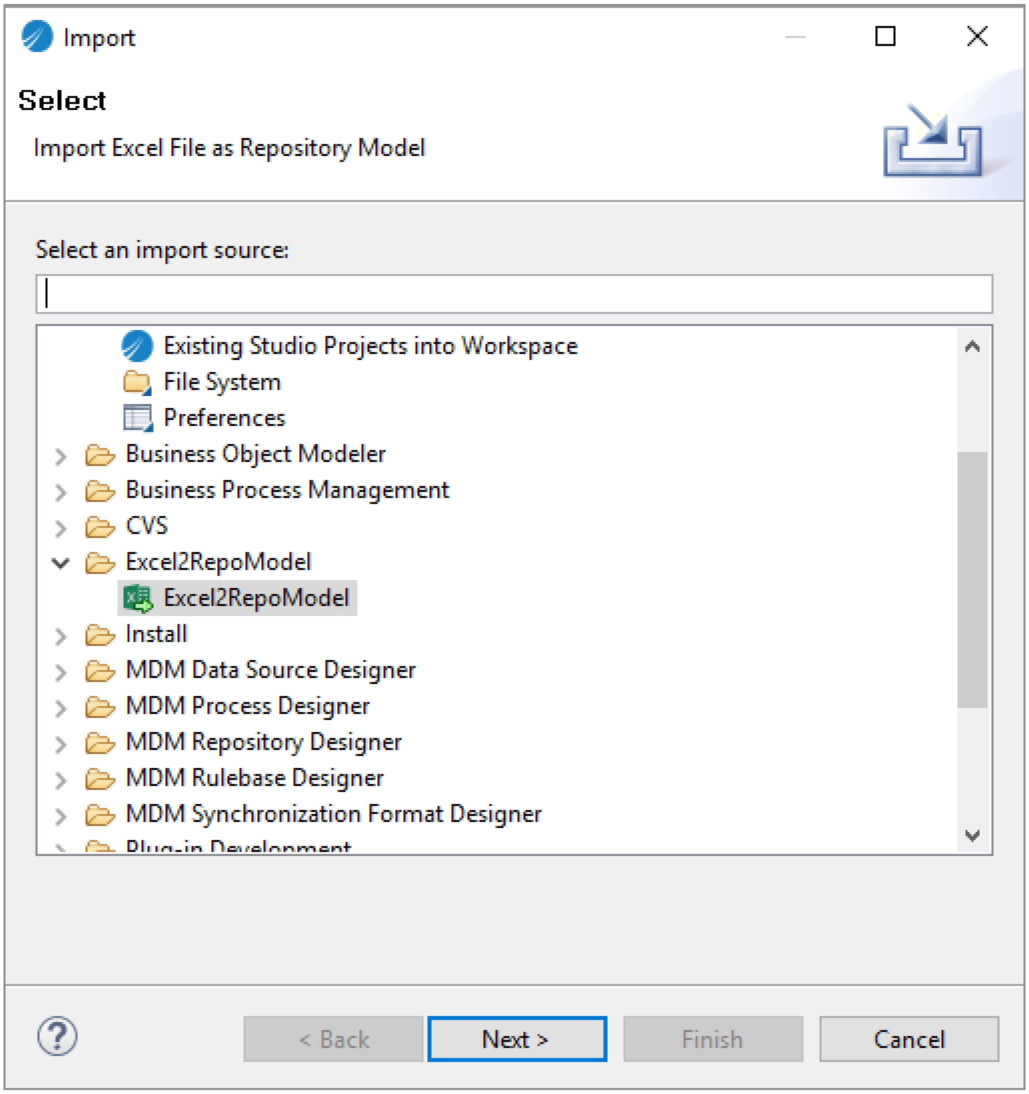
- This section starts with 'RelationshipAttributes' in the first column.
- Columns A-M must be used to specify relationship attribute fields.
- The **Relationship Name** field is mandatory and must be equal to the Relationship Name.
- The **Name** field is mandatory and must contain a unique Relationship Attribute Name.
- The **Display Name**, **Description**, **Column Name**, **Help**, **AssociationMap** properties are optional.
- The valid values for the Searchable and Display in Record List fields are either 'Y' or 'N'.
- The **Data Type** field is mandatory and must contain valid MDM data type.
- The **Length** field is optional, however, it is recommended to specify length for each attribute. The length must be provided as a numeric value.
- The **Position** field is optional. If specified, all the attributes in the repository must have their position specified. The position must be provided as a numeric value.
- End section with the text 'END' in the first column.
- The **Include** column indicates whether the row should be skipped or not. Specify 'Y' if you do not want to skip this row or specify 'N' to skip the row.

Designing Repository Model by Importing Excel File

You can design a Repository Model by importing the excel file. The supported file formats are `.xlsx` and `.xls`.

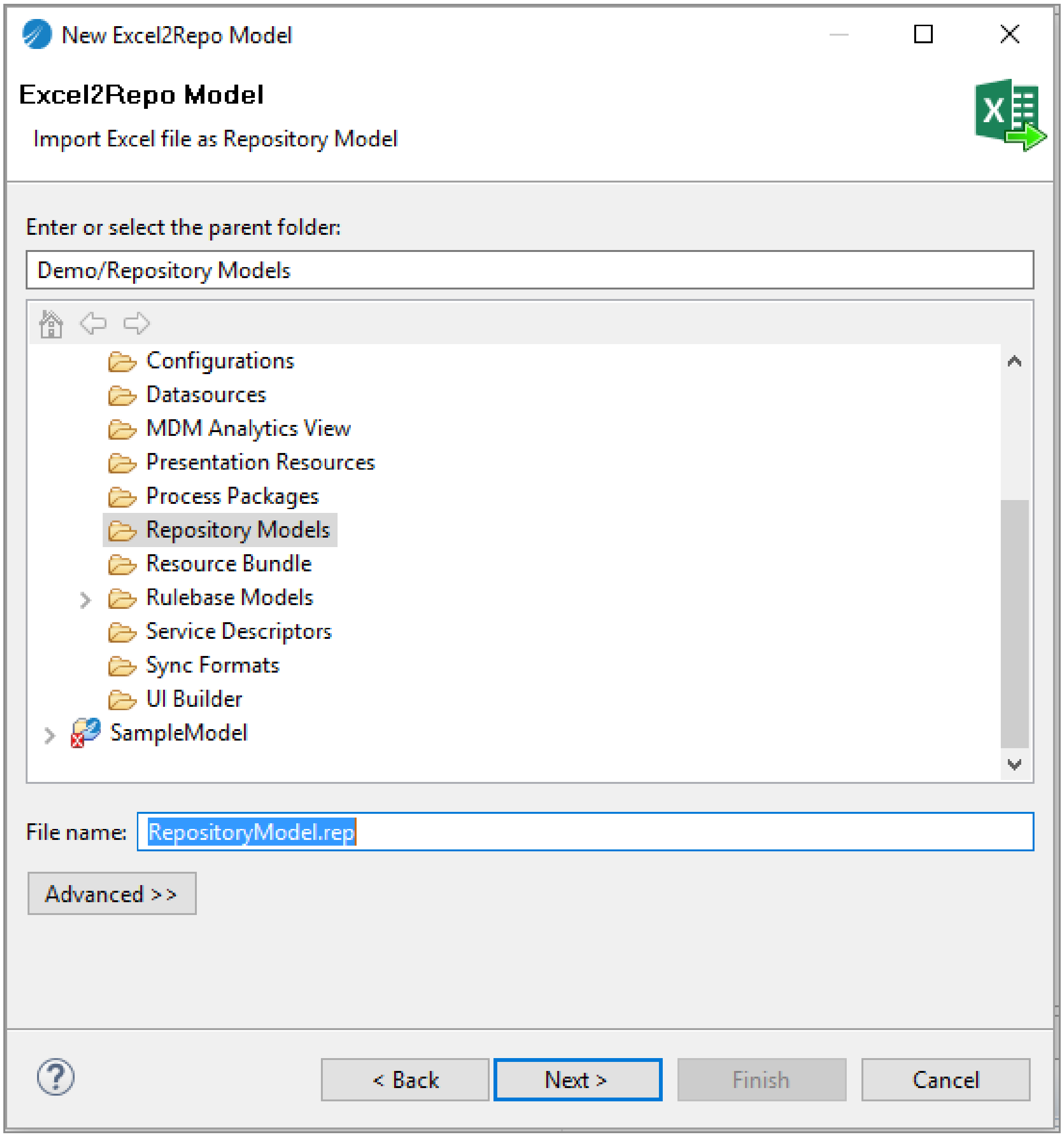
Procedure

1. Right click on the **Repository Models** and click **Import**.
2. The **Import** wizard is displayed.



Expand the **Excel2RepoModel** folder, select **Excel2RepoModel** and click **Next**.

3. The New Excel2RepoModel wizard is displayed.



New Excel2Repo Model

Excel2Repo Model

Import Excel file as Repository Model

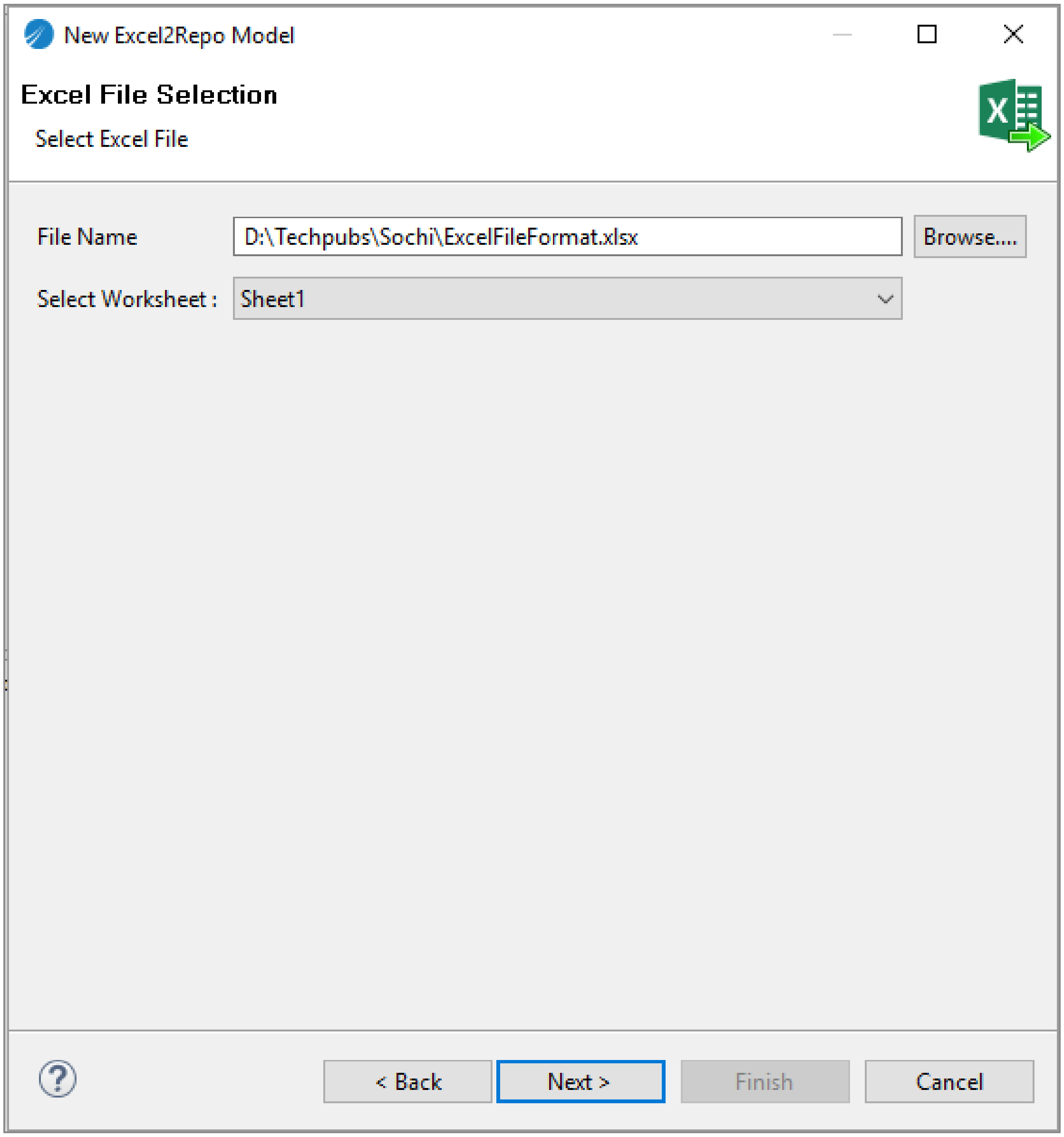
Enter or select the parent folder:

Demo/Repository Models

File name: RepositoryModel.rep

By default, the `RepositoryModel.rep` filename is displayed, you can modify the filename. Ensure that the filename extension is .rep. Click **Next**.

4. The New Excel2RepoModel wizard displays the excel file selection

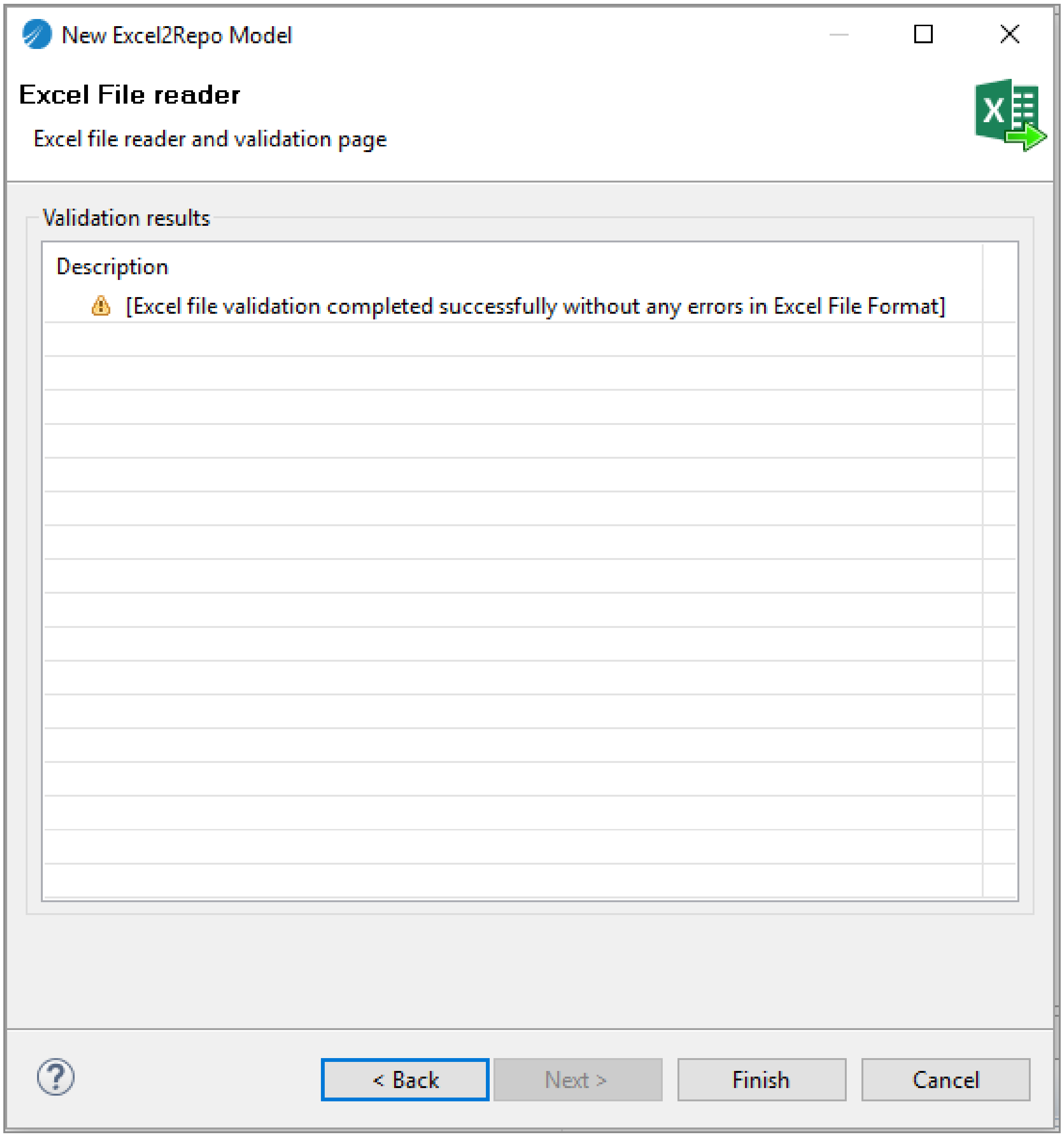


File Name D:\Techpubs\Sochi\ExcelFileFormat.xlsx

Select Worksheet : Sheet1

Click **Browse** and navigate to the path where the excel file is located. Select the sheet where the information is saved from the **Select Worksheet** drop-down list. Click **Next**

5. The New Excel2RepoModel wizard displays validations results.


New Excel2Repo Model

Excel File reader

Excel file reader and validation page

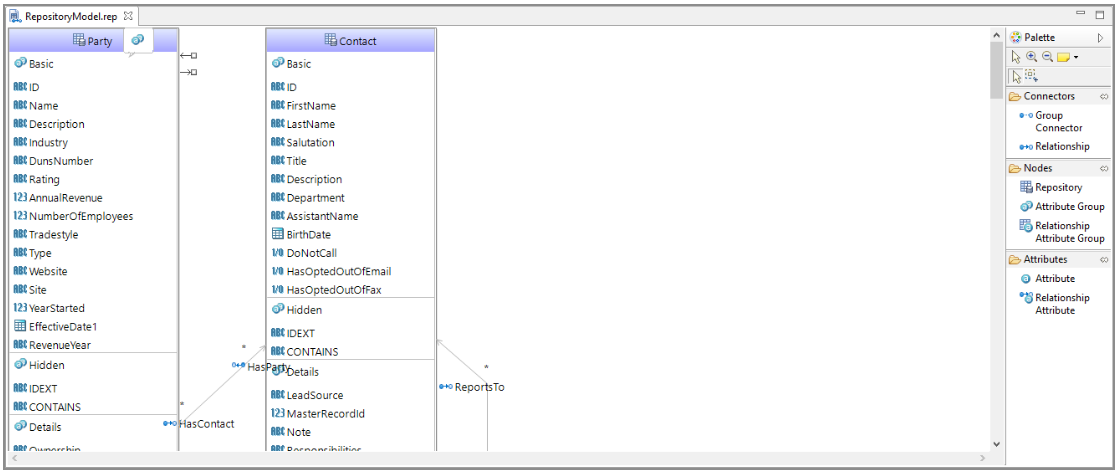
Validation results

Description

⚠ [Excel file validation completed successfully without any errors in Excel File Format]

If there are any validation errors, the description of the errors is displayed. Click **Finish**.

6. The Repository model is generated based on the information in the excel sheet.



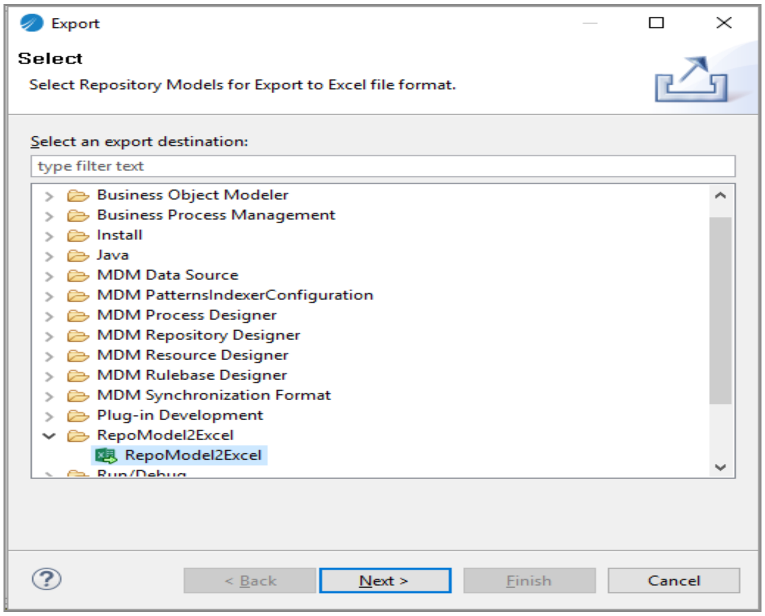
Exporting Repository Model in Excel File Format

You can export Repository Model in Excel file format.

Procedure

1. Right click on the **Repository Models**, click **Export** and click **Export**.

2. The **Export** wizard is displayed.



Export

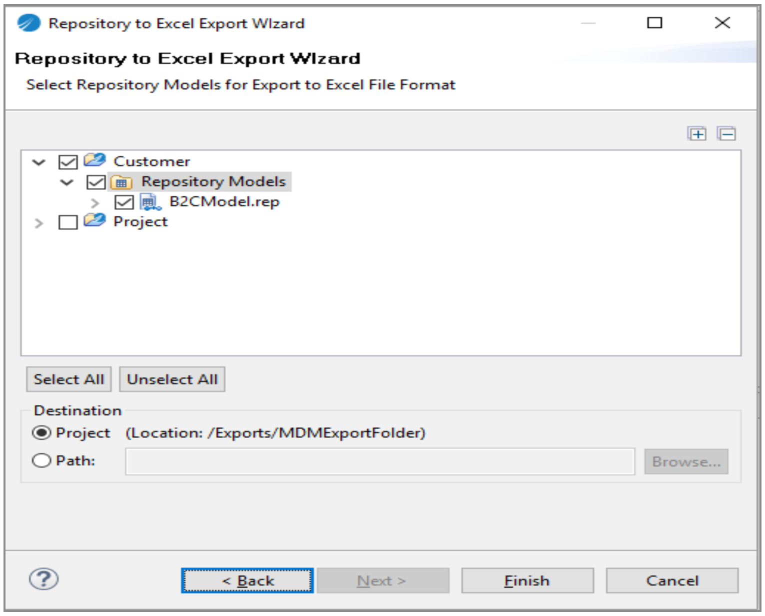
Select

Select Repository Models for Export to Excel file format.

Select an export destination:

Expand the **RepoModel2Excel** folder, select **RepoModel2Excel** and click **Next**.

3. The **Repository to Excel Export Wizard** is displayed.

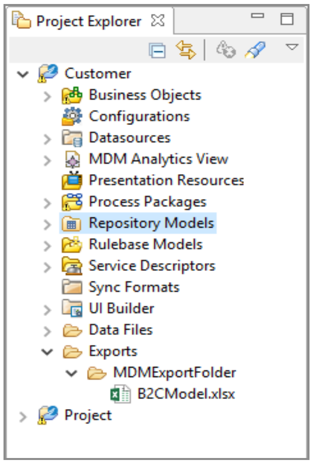


Repository to Excel Export Wizard
Repository to Excel Export Wizard
Select Repository Models for Export to Excel File Format
☑ Customer
☑ Repository Models
☑ B2CModel.rep
□ Project
Select All
Unselect All
Destination
● Project (Location: /Exports/MDMExportFolder)
○ Path:
Browse...
< Back
Next >
Finish
Cancel

Select the repository model which you want to export to excel format. By default, the project location is `/Exports/MDMExportFolder`, select **Path**, click **Browse** and navigate to the new location. Click **Finish**.

> **Note:** If you select multiple repository model to export, separate excel files are created.

4. The Project Explorer displays the Excel format in MDMExportFolder.



Deploy Repository Model

You can select the appropriate repository model, to deploy from the select module page or you can select the enterprise to deploy. Selecting the **standard** enterprise option deploys the repositories at application level.

For more information on how to deploy repository model, see Deployment

Deployment

This chapter explains direct deployment of repository and other models created in TIBCO MDM Studio.

Deployment Overview

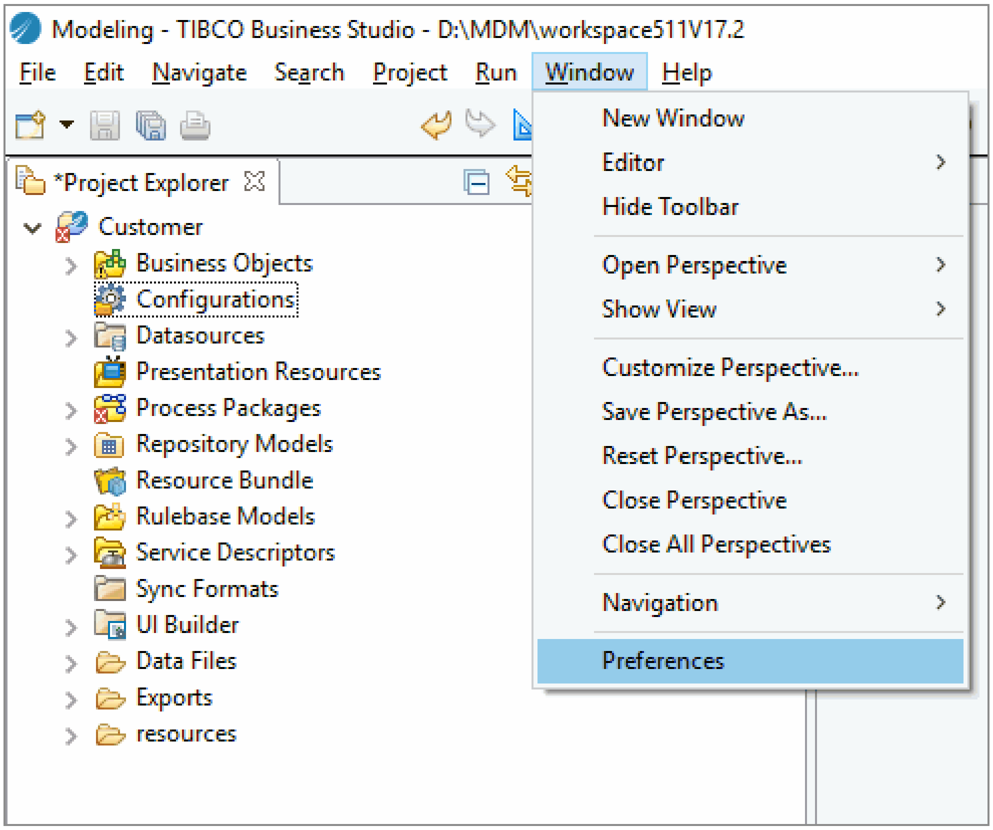
This direct deployment of models created in TIBCO MDM Studio provides a quick way to deploy the graphically defined components - rather than using the export wizard and importing or moving the exported file to the TIBCO MDM Studio manually.

Setting Server Validation Preferences

The following steps are involved in setting server validation preferences:

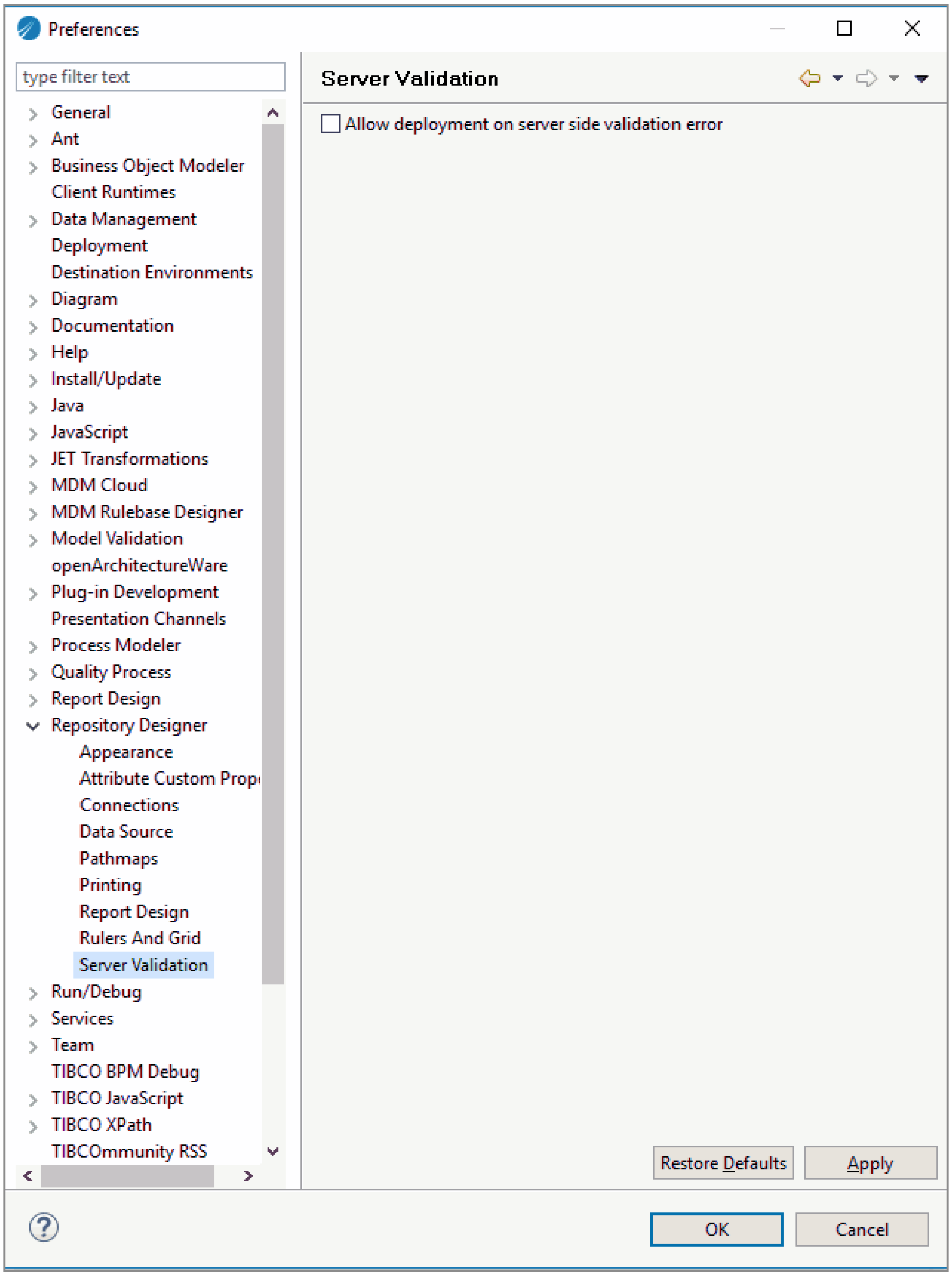
Procedure

1. On the menu bar, click **Windows** and select **Preferences**. The Preferences dialog box opens.



2. In the left navigation pane of the Preferences dialog box, click **Repository Designer > Server Validations**. The Server Validations Page opens in the right pane.

3. On the Server Validations page, select the **Allow Deployment on server side validation error** check box and click **Apply**.



□ Allow deployment on server side validation error

4. To save all changes, click **OK**.

Command Line Interface

This chapter introduces the Command Line Interface (CLI) added to TIBCO MDM Studio for automated use of the application.

Overview

With TIBCO MDM Studio evolving, all the design time activities are done using TIBCO MDM Studio. It is necessary to expose these design time activities and similar interface scripts which support in automatic deployment and testing.

To achieve this, the Command Line Interface (CLI) is introduced. The command line interface services are used for external actions such as export and deployment. In this chapter we will go through each of these external actions interfaces and the various command line services used to execute each of the actions.

Export Interfaces

Using TIBCO MDM export wizards we can export the Repository Models, Rulebases, and Workflows however to use this from an automated program is hard. It is easier to use this from the command line.

Using the command line the design artifacts (.rep, .rul, .xpdl) are transformed from the eclipse based format into the format executable by the TIBCO MDM Server. The transformed artifacts are imported into the TIBCO MDM server.

Note: On Linux, manually create an export folder in your workspace. Specify the path of the export folder while defining the output file.

Note: While running the sample from the documentation, if you encounter error, copy the syntax to a text editor and retype the hyphen.

Data Source Export Transformation

To transform a data source file to into a specific TIBCO MDM server format, use the following syntax in the command line. The file name and the workspace information is mandatory in the syntax.

For Windows

<install dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.mdm.datasourcemodel.export.TransformDataSource -i <input file> -o <output file>

-i : <full path to the input data source file>

-o : <full path to the output metadata file containing the data source information>

For example,

```
<install dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -
application com.tibco.mdm.datasourcemodel.export.TransformDataSource -i
c:\Users\myname\workspace\MyProject\Datasources\Names.ds -o
c:\Users\myname\workspace\MyProject\Datasources\export\Names.xml
```

For Linux

./TIBCOBusinessStudio -application com.tibco.mdm.datasourcemodel.export.TransformDataSource -i <input file> -o <output file>

-i : <full path to the input data source file>

-o : <full path to the output metadata file containing the data source information>

For example,

```
./TIBCOBusinessStudio -application
com.tibco.mdm.datasourcemodel.export.TransformDataSource -i
/home/apps/workspace/MyProject/Datasources/Names.ds -o
/home/apps/workspace/MyProject/Datasources/export/Names.xml
```

Synchronization Format Export Transformation

To transform a synchronization file to into a specific TIBCO MDM server format, use the following syntax in the command line. The file name and the workspace information is mandatory in the syntax.

For Windows

```
<Install Dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.mdm.syncformat.editor.TransformSyncFormat -data <full path to the Workspace
Directory> -i <input file> -o <output file>
```

-i : <full path to the input data source file>

-o : <full path to the output metadata file containing the data source information>

For example,

```
<Install Dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.mdm.syncformat.editor.TransformSyncFormat -data
"c:\Users\myname\workspace" -i "c:\Users\myname\workspace\MyProject\Sync
Formats\SyncFormat.sf" -o "c:\Users\myname\workspace\MyProject\Sync
Formats\Export\SyncFormat.xml"
```

For Linux

```
./TIBCOBusinessStudio -application
com.tibco.mdm.syncformat.editor.TransformSyncFormat -data <full path to the Workspace
Directory> -i <input file> -o <output file>
```

-i : <full path to the input data source file>

-o : <full path to the output metadata file containing the data source information>

For example,

```
./TIBCOBusinessStudio -application
com.tibco.mdm.syncformat.editor.TransformSyncFormat -data "/home/apps/workspace" -i
"/home/apps/workspace/MyProject/Sync Formats/SyncFormat.sf" -o
"/home/apps/workspace/MyProject/Sync Formats/Export/SyncFormat.xml"
```

Repository Model Export Transformation

To transform a repository file to into a specific TIBCO MDM server format, use the following syntax in the command line. The file name and the workspace information is mandatory in the syntax.

For Windows

```
<Install Dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.mdm.repositorymodel.export.TransformRepository -data <full path to the
Workspace Directory> -i <input file> -o <output file>
```

-i: <full path to the input repository file>

-o: <full path to the output metadata file containing the repository information>

For example,

```
<Install Dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.mdm.repositorymodel.export.TransformRepository -data
"c:\Users\myname\workspace" -i "c:\Users\myname\workspace\MyProject\Repository
Models\Person.rep" -o "c:\Users\myname\workspace\MyProject\Repository
Models\Export\Person.xml"
```

For Linux

```
./TIBCOBusinessStudio -application
com.tibco.mdm.repositorymodel.export.TransformRepository -data <full path to the
Workspace Directory> -i <input file> -o <output file>
```

-i: <full path to the input repository file>

-o: <full path to the output metadata file containing the repository information>

For example,

```
./TIBCOBusinessStudio -application
com.tibco.mdm.repositorymodel.export.TransformRepository -data
"/home/apps/workspace" -i "/home/apps/workspace/MyProject/Repository
Models/Person.rep" -o "/home/apps/workspace/MyProject/Repository
Models/Export/Person.xml"
```

Rulebase Export Transformation

To transform a single rulebase design file into a specific TIBCO MDM server format, use the following syntax in the command line. The file name and the workspace information is mandatory in the syntax.

For Windows

```
<Install Dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.mdm.rulebasemodel.transformation.TransformRulebase -data <Workspace Dir> -i
<input file> -o <output file>
```

-i: <full path to the input rulebase file>

-o: <full path to the output metadata file containing the rulebase information>

For example,

```
<Install Dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.mdm.rulebasemodel.transformation.TransformRulebase -data
"c:\Users\myname\workspace" -i "c:\Users\myname\workspace\MyProject\Rulebase
Models\Basic.rul" -o "c:\Users\myname\workspace\MyProject\Rulebase
Models\Export\Basic.xml"
```

For Linux

```
./TIBCOBusinessStudio -application
com.tibco.mdm.rulebasemodel.transformation.TransformRulebase -data <Workspace Dir> -i
<input file> -o <output file>
```

-i: <full path to the input rulebase file>

-o: <full path to the output metadata file containing the rulebase information>

For example,

```
./TIBCOBusinessStudio -application
com.tibco.mdm.rulebasemodel.transformation.TransformRulebase -data
"/home/apps/workspace" -i "/home/apps/workspace/MyProject/Rulebase Models/Basic.rul"
-o "/home/apps/workspace/MyProject/Rulebase Models/Export/Basic.xml"
```

Workflow Export Transformation

Note: For Linux, copy the latest CIMServices.wsdl from <Workspace Dir>/<project name>/Service Descriptors to <Install Dir>/studio-mdm/<version>/eclipse for process transformation.

To transform a single workflow design file (XPDL) into a specific TIBCO MDM server format, use the following syntax in the command line. The file name and the workspace information is mandatory in the syntax.

For Windows

```
<Install Dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.xpd.cim.export.TransformProcessFlow -data <Workspace Dir> -i <input file> -o <output file>
```

-i: <full path to the input process package file>

-o: <full path to the output metadata file containing the process package information>

For example,

```
<Install Dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.xpd.cim.export.TransformProcessFlow -data "c:\Users\myname\workspace" -i "c:\Users\myname\workspace\MyProject\Process Packages\ProcessPackage.xpdl" -o "c:\Users\myname\workspace\MyProject\Process Packages\Export\ProcessPackage.xml"
```

For Linux

```
./TIBCOBusinessStudio -application com.tibco.xpd.cim.export.TransformProcessFlow -data <Workspace Dir> -i <input file> -o <output file>
```

-i: <full path to the input process package file>

-o: <full path to the output metadata file containing the process package information>

For example,

```
./TIBCOBusinessStudio -application com.tibco.xpd.cim.export.TransformProcessFlow -data "/home/apps/workspace" -i "/home/apps/workspace/MyProject/Process Packages/ProcessPackage.xpdl" -o "/home/apps/workspace/MyProject/Process Packages/Export/ProcessPackage.xml"
```

Deploy Interfaces

The following section depicts how to deploy the Data Sources, Synchronization Formats, Repository Models, Rulebase Models and Workflow Models.

The parameters passed in the deployment syntax are as follows:

-data : fully qualified path to the workspace

-u : username of the login user. The user must have administrator privileges.

-pwd : password of the administrator login user.

-mode: Mode of deployment. It must be 'project' for project deployment. The flag also supports the values of 'rep','rul,'xpdl','project','ds','sf'.

-baseURL : The BaseURL to deploy the TIBCO MDM Project.

-project_name : TIBCO MDM Studio project name. It is also the name of the folder which contains the project file.

-e : User enterprise name.

-d : Enterprise where the project is deployed.

-nodatafile : this switch is optional. If actual data file is associated with datasource it is deployed by default. However, if -nodatafile switch is set then it is not deployed.

-datafilepath : this switch is optional. If actual data file is associated with datasource it is deployed by default. However, if -nodatafile switch is set then it is not deployed. If file is placed at relative path then mention -datafilepath <path of file>, only one flag is supported at a time in command -datafilepath or -nodatafile.

-datafilepath : this switch is optional. If actual data file is associated with datasource it is deployed by default. However, if -nodatafile switch is set then it is not deployed. If file is placed at relative path then mention -datafilepath <path of file>, only one flag is supported at a time in command -datafilepath or -nodatafile.

Note: While running the sample from the documentation, if you encounter error, copy the syntax to a text editor and retype the hyphen.

Data Source Deployment

To deploy data source, file uploaded to the data source is automatically deployed if -nodatafile parameter is not provided. To deploy a data source from eclipse format to a specific TIBCO MDM server, use the following syntax in the command line.

For Windows

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.xpd.deploy.server.cim.DeployModel -data <Workspace Directory> -u <username>
-pwd <password> -mode ds -baseURL <BaseURL> -i <input file> -e <User Enterprise> -d
<Deployment Enterprise> -datafilepath <location of datafile>
```

For example,

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.xpd.deploy.server.cim.DeployModel -data "c:\Users\mdm\workspace" -u jsmith -
pwd jsmith -mode ds -baseURL http://localhost:8080 -i
"c:\Users\myname\workspace\MyProject\Datasources\Contact.ds" -e techpubs -d techpubs
-datafilepath "C:\Users\DataFile.txt"
```

For Linux

```
./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel -data
<Workspace Directory> -u <username> -pwd <password> -mode ds -baseURL <BaseURL> -i
<input file> -e <User Enterprise> -d <Deployment Enterprise> -datafilepath <location of
datafile>
```

For example,

```
TIBCOBusinessStudio.exe -application com.tibco.xpd.deploy.server.cim.DeployModel -data
"c:\Users\mdm\workspace" -u jsmith -pwd jsmith -mode ds -baseURL http://localhost:8080
-i "c:\Users\myname\workspace\MyProject\Datasources\Contact.ds" -e techpubs -d
techpubs -datafilepath "C:\Users\DataFile.txt"
```

Synchronization Format Deployment

To deploy a synchronization format from eclipse format to a specific TIBCO MDM server, use the following syntax in the command line.

For Windows

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.xpd.deploy.server.cim.DeployModel -data <Workspace Directory> -u <username> -pwd <password> -mode sf -baseURL <BaseURL> -i <input file> -e <User Enterprise> -d <Deployment Enterprise>
```

For example,

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.xpd.deploy.server.cim.DeployModel -data "c:\Users\mdm\workspace" -u jsmith -pwd jsmith -mode sf -baseURL http://localhost:8080 -i "c:\Users\myname\workspace\MyProject\Sync Formats\SalesForce.sf" -e techpubs -d techpubs
```

For Linux

```
./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel -data <Workspace Directory> -u <username> -pwd <password> -mode sf -baseURL <BaseURL> -i <input file> -e <User Enterprise> -d <Deployment Enterprise>
```

For example,

```
./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel -data "/home/apps/workspace" -u jsmith -pwd jsmith -mode sf -baseURL http://localhost:8080 -i "/home/apps/workspace/MyProject/Sync Formats/SalesForce.sf" -e techpubs -d techpubs
```

Repository Model Deployment

To deploy a single repository model file from eclipse format to a specific TIBCO MDM server, use the following syntax in the command line. The repository file translation should be performed before deployment.

For Windows

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.xpd.deploy.server.cim.DeployModel -data <Workspace Directory> -u <username> -pwd <password> -mode rep -baseURL <BaseURL> -i <input file> -e <User Enterprise> -d <Deployment Enterprise>
```

For example,

<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.xpd.deploy.server.cim.DeployModel -data "c:\Users\mdm\workspace" -u jsmith -pwd jsmith -mode rep -baseURL http://localhost:8080 -i "c:\Users\myname\workspace\MyProject\Repository Models\Person.rep" -e techpubs -d techpubs

For Linux

./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel -data <Workspace Directory> -u <username> -pwd <password> -mode rep -baseURL <BaseURL> -i <input file> -e <User Enterprise> -d <Deployment Enterprise>

For example,

./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel -data "/home/apps/workspace" -u jsmith -pwd jsmith -mode rep -baseURL http://localhost:8080 -i "/home/apps/workspace/MyProject/Repository Models/Person.rep" -e techpubs -d techpubs

Rulebase Model Deployment

To deploy a single rulebase model file from eclipse format to a specific TIBCO MDM server, use the following syntax in the command line. As a result of the deployment the master catalog folder is added or overwritten in the deployment folder in the TIBCO MDM server.

For Windows

<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.xpd.deploy.server.cim.DeployModel -data <Workspace Dir> -u <username> -pwd <password> -mode rul -baseURL <BaseURL> -i <input file> -e <User Enterprise> -d <Deployment Enterprise>

For example,

<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.xpd.deploy.server.cim.DeployModel -data "c:\Users\myname\workspace" -u jsmith -pwd jsmith -mode rul -baseURL http://localhost:8080 -i "c:\Users\myname\workspace\MyProject\Rulebase Models\Basic.rul" -e techpubs -d techpubs

For Linux

```
./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel -data
<Workspace Dir> -u <username> -pwd <password> -mode rul -baseURL <BaseURL> -i <input
file> -e <User Enterprise> -d <Deployment Enterprise>
```

For example,

```
./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel -data
"/home/apps/workspace" -u jsmith -pwd jsmith -mode rul -baseURL http://localhost:8080 -i
"/home/apps/workspace/MyProject/Rulebase Models/Basic.rul" -e techpubs -d techpubs
```

Workflow Model Deployment

To deploy a single workflow model file from eclipse format to a specific TIBCO MDM server. This will result in one file added/overwritten in the workflow deployment folder in the TIBCO MDM server (common workflow folder in most cases).

For Windows

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.xpd.deploy.server.cim.DeployModel -data <"full path of the Workspace"> -u
<username having admin login credentials> -pwd <password having admin login
credentials> -mode xpdl -baseURL <the BaseURL to deploy the MDM Design service> -i
<"full path to the .xpdl file"> -e <User Enterprise> -d <Deployment Enterprise>
```

For example,

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.xpd.deploy.server.cim.DeployModel –data "c:\Users\myname\workspace" -u
jsmith -pwd jsmith -mode xpdl -baseURL http://localhost:8080 -i
"c:\Users\myname\workspace\MyProject\Process Packages\ProcessPackage.xpdl" -e
techpubs -d techpubs
```

For Linux

```
./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel -data
<"full path of the Workspace"> -u <username having admin login credentials> -pwd
<password having admin login credentials> -mode xpdl -baseURL <the BaseURL to deploy
the MDM Design service> -i <"full path to the .xpdl file"> -e <User Enterprise> -d
<Deployment Enterprise>
```

For example,

```
./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel –data
"/home/apps/workspace" -u jsmith -pwd jsmith -mode xpdl -baseURL http://localhost:8080
-i "/home/apps/workspace/MyProject/Process Packages/ProcessPackage.xpdl" -e techpubs
-d techpubs
```

Deploy Projects

The deployment of an entire TIBCO MDM Studio project is also supported, including all .rep, .rul and .xpdl model files, from their eclipse format to a specific TIBCO MDM server. Since the deployment includes lot of artifacts a check is done. The deployment process is terminated as soon as an error occurs. The error details are captured in error log.

The interface takes the workspace location and the project name (same as folder) and then iterates through all the folders and adds all Data Source, Rulebase, Repository Model and Process Designer files definition of a project. In addition, you can also deploy limited set of artifacts by adding a file list as an argument to the project deployment.

To deploy the entire TIBCO MDM Studio project, use the following syntax in the command line.

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application
com.tibco.xpd.deploy.server.cim.DeployModel -data <Workspace Dir> -u <username> -pwd
<password> -mode project -baseURL <BaseURL> -project_name <Project Name> -e <User
Enterprise> -d <Deployment Enterprise> (-abort_on_error) -fileFilter <deployment
file1>,<deployment file2>,
```

-data : fully qualified path to the workspace

-u : username of the login user. The user must have administrator privileges.

-pwd : password of the administrator login user.

-mode: Mode of deployment. It must be 'project' for project deployment. The flag also supports the values of 'rep','rul,'xpdl','project',’ds’,’sf’.

-baseURL : The Base URL to deploy the TIBCO MDM Project.

-project_name : TIBCO MDM Studio project name. It is also the name of the folder which contains the project file.

-e : User enterprise name.

-d : Enterprise where the project is deployed.

-abort_on_error : aborts on encountering an error.

-fileFilter : this switch is optional. If specified, only files from the list are deployed. If not specified, all the files from the project are deployed.

The sample deploys the project "Demo" to a localhost server in a techpubs enterprise.

> **Note:** While running the sample from the documentation, if you encounter error, copy the syntax to a text editor and retype the hyphen.

For Windows

<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -application com.tibco.xpd.deploy.server.cim.DeployModel -data "c:\Users\myname\workspace" -project_name Demo -u jsmith -pwd jsmith -mode project -baseURL http://localhost:8080 -e techpubs -d techpubs -abort_on_error -fileFilter "Repository Models\RepoAllTypeAttributes.rep,Rulebase Models\RepoWithAllTypes.rul,Sync Formats\SyncFormat.sf,Process Packages\wfin26productaddinternaleditv1.xpdl,Datasources\LanguageCodes.ds"

For Linux

./TIBCOBusinessStudio -application com.tibco.xpd.deploy.server.cim.DeployModel -data "/home/apps/workspace" -project_name Demo -u jsmith -pwd jsmith -mode project -baseURL http://localhost:8080 -e techpubs -d techpubs -fileFilter "Repository Models/RepoAllTypeAttributes.rep,Rulebase Models/RepoWithAllTypes.rul,Sync Formats/SyncFormat.sf,Process Packages/wfin26productaddinternaleditv1.xpdl,Datasources/Country.ds"

Incremental Deployment of Repositories

The incremental deployment is currently supported only for .rep.

To incremental deploy a single project, use the following syntax in the command line.

```
./TIBCOBusinessStudio -application
com.tibco.xpd.deploy.server.cim.DeployModel -data <Workspace Directory>
-u <username> -pwd <password> -mode rep -baseURL <BaseURL> -i
```

```
<repository model input file> -e <User Enterprise> -d <Deployment
Enterprise> -repositoryFilter "PERSON,ADDRESS,LOCATION" -
relationshipFilter "PERSON\PersonToAddress,ADDRESS\AddressOfPerson" -
inputmapFilter "PERSON\InputMap1,ADDRESS\InputMap2" -outputmapFilter
"PERSON\OutputMap1" -classificationFilter "PERSON\Classification1"
```

`-data` : fully qualified path to the workspace

`-u` : username of the login user. The user must have administrator privileges.

`-pwd` : password of the administrator login user.

`-mode`: Mode of deployment. It must be 'project' for project deployment. The flag supports the values only of 'rep'.

`-baseURL` : The Base URL to deploy the TIBCO MDM Project.

`-i` : full path to the repository model.

`-e` : User enterprise name.

`-d` : Enterprise where the project is deployed.

`-repositoryFilter` : repository names which you want to filter.

`-relationshipFilter` : relationship names which you want to filter.

`-inputmapFilter` : inputmap names which you want to filter.

`-outputmapFilter` : output names which you want to filter.

`-classificationFilter` : classification names which you want to filter.

The sample deploys the project "Demo" to a localhost server in a techpubs enterprise.

For Windows

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe
-application com.tibco.xpd.deploy.server.cim.DeployModel -data
"c:\Users\myname\workspace" -u jsmith -pwd jsmith -mode rep -baseURL
http://localhost:8080 -i
"c:\Users\myname\workspace\Demo\RepositoryModels\Demo.rep" -e techpubs -
d techpubs -repositoryFilter "PERSON,ADDRESS,LOCATION" -
relationshipFilter "PERSON\PersonToAddress,ADDRESS\AddressOfPerson" -
inputmapFilter "PERSON\InputMap1,ADDRESS\InputMap2" -outputmapFilter
```

```
"PERSON\OutputMap1" -classificationFilter "PERSON\Classification1"
```

For Linux

```
/TIBCOBusinessStudio -application
com.tibco.xpd.deploy.server.cim.DeployModel -data "/home/apps/workspace"
-u jsmith -pwd jsmith -mode rep -baseURL http://localhost:8080 -i
"/home/apps/workspace/Demo/RespositoryModels/Demo.rep" -e techpubs -d
techpubs -repositoryFilter "PERSON,ADDRESS,LOCATION" -relationshipFilter
"PERSON\PersonToAddress,ADDRESS\AddressOfPerson" -inputmapFilter
"PERSON\InputMap1,ADDRESS\InputMap2" -outputmapFilter
"PERSON\OutputMap1" -classificationFilter "PERSON\Classification1"
```

Custom Form Deployment

Custom forms can be deployed in both Windows and Linux environments. From the CLI, you can deploy a single custom form or multiple custom forms to the TIBCO MDM server. To deploy the custom forms use the following properties and syntax in the Command Line Interface:

Deploying a Single Custom Form

You can deploy a single custom form to the TIBCO MDM server in a Windows or a Linux environment.
For Windows

To deploy a single custom form, enter the following command at the command prompt:

```
<Install Dir>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe -
application com.tibco.xpd.deploy.server.cim.DeployModel -data "D:/cim_
Studio/Studio_Projects_Ws_1" -i "D:\cim_Studio\Studio_Projects_Ws_
1\UiPageDeploy\UI Builder\UiCustomPage5.mub" -u cim90 -pwd cim90 -mode
mub -baseURL http://localhost:8080/  -e cimdev90 -d cimdev90  -
uiPageConfiguration ["menuAction":"Add to Menu","rootItem":"Custom
Pages","menuItem":"UiCustomPage5","metadataOperation":"[Add Record, View
Record, Modify
Record]","directoryStructure":"Default","componentName":"UiCustomPage5"]
```

For Linux

To deploy a single custom form, enter the following command at the command prompt:

```
./ TIBCOBusinessStudio.exe -application
com.tibco.xpd.deploy.server.cim.DeployModel -data
“/home/apps/workspace/UiPageDeploy/UI Builder/UiCustomPage5.mub” -i
“/home/apps/workspace” u cim90 -pwdcim90 -mode mub -baseURL
http://localhost:8080/ -e cimdev90 -d cimdev90 -uiPageConfiguration
["menuAction":"Add to
Menu","rootItem":"CustomPages","menuItem":"UiCustomPage5","metadataOpera
tion":"[Add Record, View Record, Modify
Record]","directoryStructure":"Default","componentName":"UiCustomPage5"]
```

To deploy a single custom form, use the following properties:

Properties	Description
`-mode`	The mode in which you want to deploy the custom form. For example, `-mode mub`
`-i`	A full path of the input UI builder file
`-uiPageConfiguration`	Page configuration in square brackets followed by the –`uiPageConfiguration` flag. For example, `-uiPageConfiguration ["menuAction":"Add to Menu","rootItem":"Custom Pages","menuItem":"UiCustomPage5","metadataOperation":"[Add Record, View Record, Modify Record]","directoryStructure":"Default","componentName": "UiCustomPage5"]`
`-menuAction`	Add to Menu or Link to Metadata Operation • If you add the value of `-menuAction` as Add to Menu, then -`metadataOperation` is optional . • If you add the value of `-menuAction` as Link to Metadata Operation, then `-rootItem` and `-menuItem` are optional.

`-directoryStructure`	The directory structure such as Default or Standalone
`-metadataOperation`	The operations such as Add Record, View Record, and Modify Record
`-rootItem`	Provide any of the following root items: • Custom Pages • Inbox • Administration • System Operations • Master Data • Browse and Search • Business Process • Event Log • Tools

Deploying Multiple Custom Forms at a Time

For Windows

To deploy multiple custom forms at a time, enter the following command on the command line:

```
<Install Directory>\studio-mdm\<version>\eclipse>TIBCOBusinessStudio.exe
-application com.tibco.xpd.deploy.server.cim.DeployModel -data " D:\cim_
Studio\Studio_Projects_Ws_1" -project_name UiPageDeploy -u cim90 -pwd
cim90 -mode project -baseURL http://localhost:8080 -e cimdev90 -d
cimdev90 -abort_on_error -fileFilter "UI Builder\UiCustomPage1.mub,UI
Builder\UiCustomPage2.mub,UI Builder\UiCustomPage3.mub" -
uiPageConfiguration "C:\Users\amshinde\Desktop\uiconfig.json"
```

For Linux

To deploy multiple custom forms at a time, enter the following command on the command line:

```
./TIBCOBusinessStudio -application
com.tibco.xpd.deploy.server.cim.DeployModel -data "/home/apps/workspace"
-project_name UiPageDeploy -u cim90 -pwd cim90 -mode project -baseURL
http://localhost:8080 -e cimdev90 -d cimdev90 -fileFilter
"UIBuilder\UiCustomPage1.mub,UI Builder\UiCustomPage2.mub,UI
Builder\UiCustomPage3.mub” -uiPageConfiguration
"/home/apps/workspace/uiconfig.json"
```

To deploy multiple custom forms at a time use the following properties:

Properties	**Description**
`-mode`	A mode in which you want to deploy the custom form. For example, `-mode project`
`-fileFilter`	A filter to specify files that you want to deploy. This is optional, all the files in a project are deployed by default. Use this filter only to deploy specific files. For example, `"UI Builder\UiCustomPage1.mub,UI Builder\UiCustomPage2.mub"`
`-uiPageConfiguration` (file path)	UI Page configuration `-uiPageConfiguration` file path. For example, `-uiPageConfiguration "C:\Users\amshinde\Desktop\uiconfig.json"`
`-uiPageConfiguration` (file)	The format of `-uiPageConfiguration` written in the `.json` file: `{"UiCustomPage1":{"menuAction":"Add to Menu","rootItem":"Custom Pages","menuItem":"UiCustomPage1","metadataOperation":"","directoryStructure":"Default","componentName":"UiCustomPage1"},"UiCustomPage2":{"menuAction":"Add to Menu","rootItem":"Custom Pages","menuItem":"UiCustomPage2","metadataOperation":"","directoryStructure":"Default","componentName":"UiCustomPage2"}}`

Custom Import

Custom Import Sample Project

A sample plug-in has been provided for reference on how to import repository metadata from custom XML format.

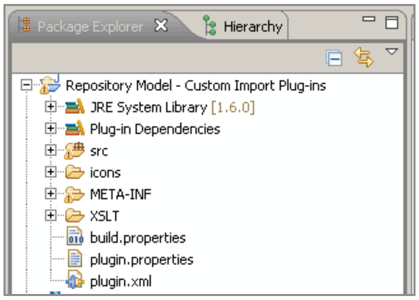
This sample provides an example that facilitates user to import metadata from TIBCO MDM version 9.1.1 format.

Download the Sample

To download the custom import sample, refer Accessing Samples

View the Sample Project

A project named **Repository Model - Custom Import Plug-ins** will be available in the workspace and can be modified. The perspective automatically changes to Java.

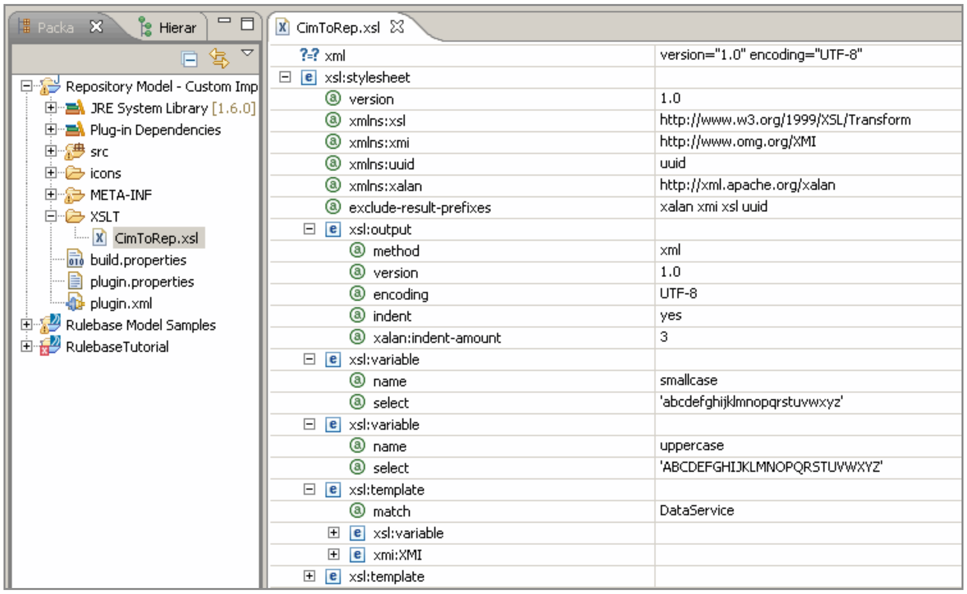


Edit the Sample Project

The different import formats can be implemented by changing the xsl file (**XSLT/CimToRep.xsl**).

Note: The name of the xsl file is fixed (CimToRep.xsl) but can be changed in the RepoImportWizard.java file.

The xslt file can be modified through the Eclipse editor:



?=? xml	version="1.0" encoding="UTF-8"
xsl:stylesheet	
version	1.0
xmlns:xsl	http://www.w3.org/1999/XSL/Transform
xmlns:xmi	http://www.omg.org/XMI
xmlns:uuid	uuid
xmlns:xalan	http://xml.apache.org/xalan
exclude-result-prefixes	xalan xmi xsl uuid
xsl:output	
method	xml
version	1.0
encoding	UTF-8
indent	yes
xalan:indent-amount	3
xsl:variable	
name	smallcase
select	'abcdefghijklmnopqrstuvwxyz'
xsl:variable	
name	uppercase
select	'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
xsl:template	
match	DataService
xsl:variable	
xmi:XMI	
xsl:template	

The supplied xslt transforms the TIBCO MDM Server 7.2 export meta format into the Repository Designer format.

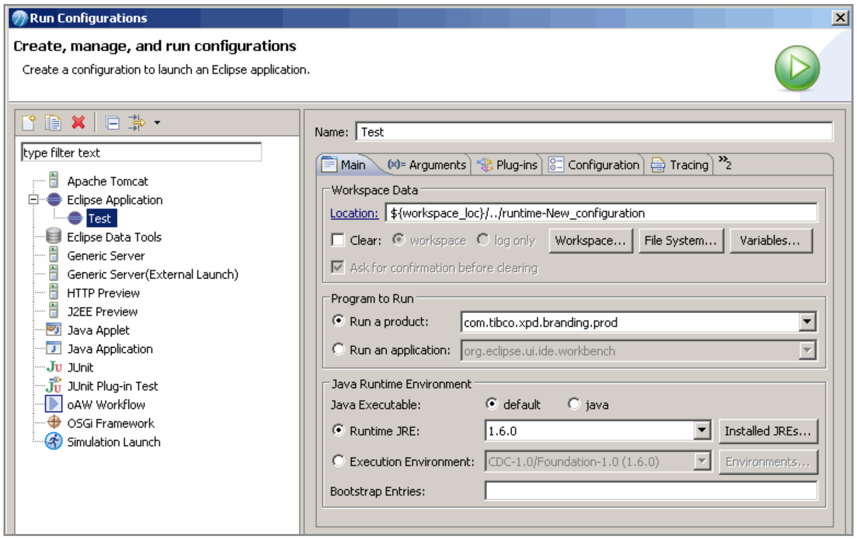
Note: The <DataDomain> part of the structure has to be generated by the xslt transformation and contains the structural model. The <Diagram> section is automatically generated after xslt transformation.

Running and Debugging the Project

The project can be run (and debugged) using the Run (or Debug) dialog.

Procedure

1. Goto **Run > Run Configurations**
2. Right click **Eclipse** and select **New** to create a new Eclipse Application (for example, **Test**).



Name: Test

Workspace Data

Location: ${workspace_loc}/../runtime-New_configuration

□ Clear: ● workspace ○ log only

☑ Ask for confirmation before clearing

Program to Run

● Run a product: com.tibco.xpd.branding.prod

○ Run an application: org.eclipse.ui.ide.workbench

Java Runtime Environment

Java Executable: ● default ○ java

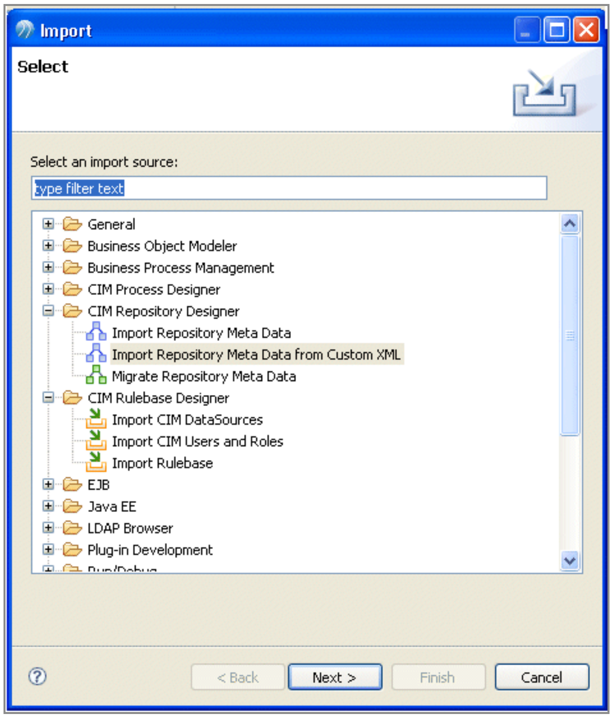
● Runtime JRE: 1.6.0

○ Execution Environment: CDC-1.0/Foundation-1.0 (1.6.0)

Bootstrap Entries:

3. Adjust the memory parameter by going to the **Arguments** tab and adding the following into the **VM arguments** section: (-XX:MaxPermSize=256m -Xmx384m)

4. Click **Apply** and then click **Run**. A new application will come up and the new wizard can be tested and debugged. The **File > Import** menu will show the new wizard.



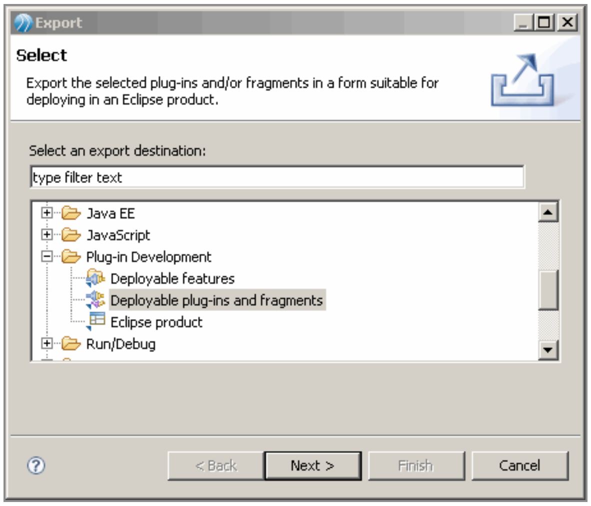
Exporting and Deploying

The plugin can be exported as a deployable plugin and added to the distribution plugin folder for deployment.

Procedure

1. Right click the Project and select **Export.**

2. Select **Plug-in Development > Deployable plug-ins and fragments.**



Export

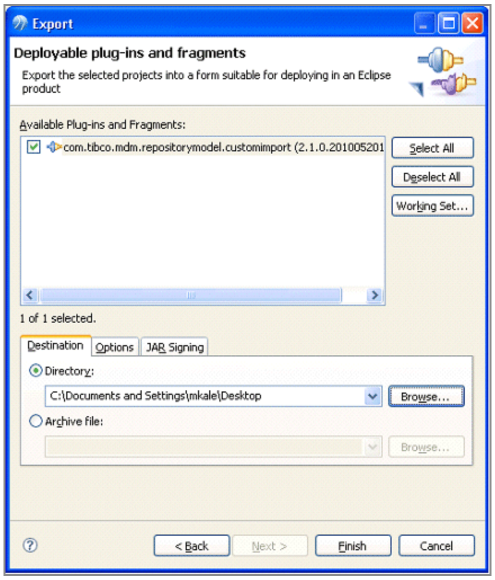
Select

Export the selected plug-ins and/or fragments in a form suitable for deploying in an Eclipse product.

Select an export destination:

type filter text

3. Select the plug-in to export and destination directory. Click **Next**.



Export

Deployable plug-ins and fragments

Export the selected projects into a form suitable for deploying in an Eclipse product

Available Plug-ins and Fragments:

☑ com.tibco.mdm.repositorymodel.customimport (2.1.0.201005201

1 of 1 selected.

● Directory:

C:\Documents and Settings\mkale\Desktop

○ Archive file:

4. Click **Finish**.

5. Copy the exported plug-in jar into <BS_INSTALL_HOME>\studio\<version>\studio-addins\eclipse\plugins.
6. Restart TIBCO MDM Studio and verify that the custom import plug-in works properly.

Data Source Explorer

This chapter describes the process for creating a new database connection profile.

Data Source Explorer

The Data Source Explorer contains connection profile instances. The Data Source Explorer provides a list of configured connection profiles. If categories are enabled, you can see the list grouped into categories, for example, Databases and ODA Data Sources. Use the Data Source Explorer to connect to, navigate, and interact with resources associated with the selected connection profile.

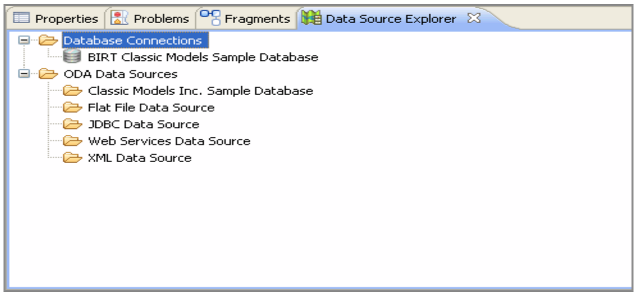
Creating a New Database Connection profile

A connection profile contains the connection property information needed to connect to a database runtime instance.

> **Warning:** Ensure that the database server is running before attempting to test the connection or to connect to the database server.

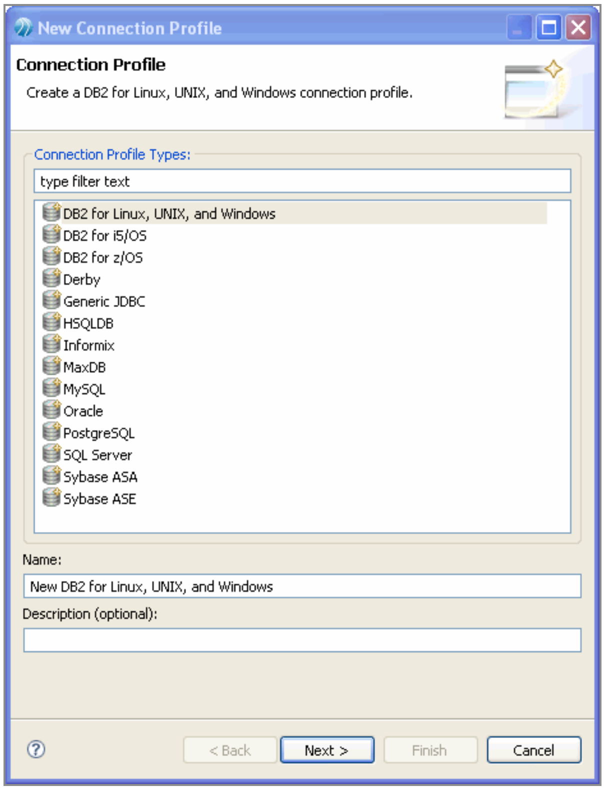
Procedure

1. In **Data Source Explorer**, Right-click on **Database Connections** and select **New**.



- Database Connections
  - BIRT Classic Models Sample Database
- ODA Data Sources
  - Classic Models Inc. Sample Database
  - Flat File Data Source
  - JDBC Data Source
  - Web Services Data Source
  - XML Data Source

2. The new connection profile dialog box is displayed.



New Connection Profile

Connection Profile

Create a DB2 for Linux, UNIX, and Windows connection profile.

Connection Profile Types:

type filter text

- DB2 for Linux, UNIX, and Windows
- DB2 for i5/OS
- DB2 for z/OS
- Derby
- Generic JDBC
- HSQLDB
- Informix
- MaxDB
- MySQL
- Oracle
- PostgreSQL
- SQL Server
- Sybase ASA
- Sybase ASE

Name: New DB2 for Linux, UNIX, and Windows

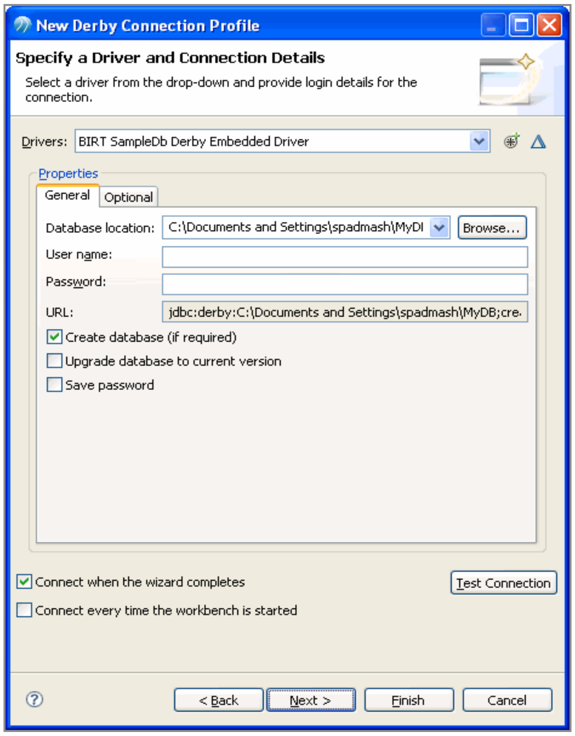
Description (optional):

3. Select the database from the filter type list for which you want to create a connection profile. For example, to list all databases that begin with the letter D, enter a d in the text box. You can also enter a text string such as JDBC or embedded to filter the list

4. Change the Name from the system-provided default, and enter a Description. You must change the name if a connection profile already exists with the same name.

5. Click **Next**.

6. The driver details dialog box for the selected database if displayed.



Specify a Driver and Connection Details

Select a driver from the drop-down and provide login details for the connection.

Drivers: BIRT SampleDb Derby Embedded Driver

Database location: C:\Documents and Settings\spadmash\MyDI

User name:

Password:

URL: jdbc:derby:C:\Documents and Settings\spadmash\MyDB;cre

☑ Create database (if required)

□ Upgrade database to current version

□ Save password

☑ Connect when the wizard completes

□ Connect every time the workbench is started

7. Select the appropriate driver from the drop-down list. If a driver definition does not exist for the database type selected, you can create one. Click ⊕ to create a new driver definition. Select △ to edit an existing driver definition. .

8. The Properties tab displays the default template information for the selected driver. Add or modify the Properties information in the **General** or **Optional** Properties tabs.

   - Select the **Save Password** checkbox to save the password for the database server login.

   - Select **Connect When the Wizard Completes** checkbox to connect to the database server once you finish defining this connection profile.

   - Select **Connect Every Time the Workbench Is Started** checkbox to automatically connect to the database server using this connection profile when you launch Eclipse.

   - Click **Test Connection** to ping the server and to verify that the connection profile is working.

9. If you want to create database select the **Create database** checkbox and if you want to upgrade your database then select the **Upgrade database to current version** checkbox.

10. Click **Next** to view Summary information.
11. Click **Finish** to create the connection profile.

Troubleshooting

This appendix lists some of the common errors and ways to troubleshooting them.

General Troubleshooting

Failing Deployment

Issue: Repository Model deployment on WebSphere Application Server failed with error "java.sql.SQLException: ORA-01430: column being added already exists in table " on TIBCO MDM server.

Solution: On Websphere Application Server, the datasource property "Statement cache size" property value is **10**. Set the "Statement cache size" property value to **0**. An incorrect setting may result in a TransactionRolledback exception. This setting is Windows-specific.

To change statement cache size:

Login to the Administrative console to configure the properties.

In the left panel, expand **Resources** > **JDBC** Providers. The Data Sources panel is displayed on the right.

Under Preferences, click *data source name*. For example, **eCMDataSource**. The **Configuration** tab is displayed.

Under Additional Properties, click the **WebSphere Application Server data source properties** link.

In the Statement cache size field, enter 0.

Click the **OK** button. A message is displayed with the Save and Review options.

Click the **Save** link to save changes to the master configuration.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact Support, and join Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO® MDM Studio Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal.