Informatica PowerCenter (Version 9.1.0 Hotfix 3)

Metadata Manager Custom Metadata Integration Guide
This product includes software licensed under the Academic Free License (http://www.opensource.org/licenses/afl-3.0.php), the Common Development and Distribution License (http://www.opensource.org/licenses/cddl1.php ) the Common Public License (http://www.opensource.org/licenses/cpl1.0.php), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (http://www.opensource.org/licenses/bsd-license.php) the MIT License (http://www.opensource.org/licenses/mit-license.php ) and the Artistic License (http://www.opensource.org/licenses/artistic-license-1.0).

This product includes software copyright © 2003-2006 Joe Waines, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at http://xstream.codehaus.org/license.html. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit http://www.extreme.indiana.edu/.

This Software is protected by U.S. Patent Numbers 5,794,246; 6,014,670; 6,016,501; 6,029,178; 6,032,158; 6,040,871; 6,044,374; 6,092,086; 6,208,990; 6,339,775; 6,640,226; 6,789,096; 6,820,077; 6,823,373; 6,850,947; 6,895,471; 7,117,215; 7,162,843; 7,256,590; 7,281,001; 7,406,748; 7,496,588; 7,523,121; 7,564,422; 7,720,842; 7,721,270; and 7,774,791 , international Patents and other Patents Pending.

DISCLAIMER: Informatica Corporation provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica Corporation does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.

2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

Part Number: MM-CIG-910-HF3-0001
Table of Contents

Preface  ................................................................. iv
Informatica Resources. .......................................................... iv
  Informatica Customer Portal. ............................................ iv
  Informatica Documentation. .............................................. iv
  Informatica Web Site ................................................... iv
  Informatica How-To Library ........................................... iv
  Informatica Knowledge Base ........................................... v
  Informatica Multimedia Knowledge Base .......................... v
  Informatica Global Customer Support ............................... v

Chapter 1: Understanding Custom Metadata Integration ..................... 1
Understanding Custom Metadata Integration Overview ........................ 1
Custom Metadata Integration Process ....................................... 2
  Adding and Loading Metadata for a Custom Metadata Source .......... 2
  Adding Custom Properties ......................................... 3
  Adding Custom Relationships ...................................... 3
Metadata Manager Concepts .................................................. 4
  Models .......................................................... 4
  Classes ...................................................... 4
  Properties .................................................. 4
  Groups ..................................................... 5
  Relationships ............................................... 5
  Business Name .............................................. 5
Using the Model Tab .......................................................... 6
  Browsing Models ............................................... 6
  Editing Models ............................................... 6
AccessDB Example ............................................................. 7
  AccessSchema .................................................. 7
  AccessTable and AccessView ....................................... 7
  AccessTableColumn and AccessViewColumn .......................... 7

Chapter 2: Creating and Configuring Custom Models ............................ 9
Creating and Editing a Custom Model Overview ................................ 9
  Establish the Model Structure .................................... 10
Configuring a Model ............................................................ 10
  Creating a Model .................................................. 10
  Editing a Model .................................................. 10
  Deleting a Model ................................................. 11
Chapter 5: Migrating Custom Metadata ............................................ 33
Migrating Custom Metadata Overview ............................................ 33
Step 1. Migrate the Models .......................................................... 34
  Exporting a Model .................................................................. 34
  Importing a Model .................................................................. 34
Step 2. Migrate Custom Metadata ................................................. 35
  Exporting Metadata ................................................................ 35
  Importing Metadata ................................................................ 35
  Rules and Guidelines for Importing Custom Metadata ............... 35
Step 3. Migrate Custom Resource Templates .................................... 36
  Exporting a Custom Resource Template ................................... 36
  Importing a Custom Resource Template ................................... 36
  Rules and Guidelines for Migrating Custom Resource Templates .... 37
Step 4. Configure and Load the Custom Resources ............................ 37
Index ................................................................................. 38
Preface

The *Metadata Manager Custom Metadata Integration Guide* provides methodology and procedures for integrating custom metadata into the Metadata Manager warehouse. This book is written for system administrators who want to load metadata from a repository type for which Metadata Manager does not package a model. This book assumes that system administrators have knowledge of relational database concepts, models, and PowerCenter.

Informatica Resources

Informatica Customer Portal

As an Informatica customer, you can access the Informatica Customer Portal site at [http://mysupport.informatica.com](http://mysupport.informatica.com). The site contains product information, user group information, newsletters, access to the Informatica customer support case management system (ATLAS), the Informatica How-To Library, the Informatica Knowledge Base, the Informatica Multimedia Knowledge Base, Informatica Product Documentation, and access to the Informatica user community.

Informatica Documentation

The Informatica Documentation team takes every effort to create accurate, usable documentation. If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at info_documentation@informatica.com. We will use your feedback to improve our documentation. Let us know if we can contact you regarding your comments.

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to Product Documentation from [http://mysupport.informatica.com](http://mysupport.informatica.com).

Informatica Web Site

You can access the Informatica corporate web site at [http://www.informatica.com](http://www.informatica.com). The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

Informatica How-To Library

As an Informatica customer, you can access the Informatica How-To Library at [http://mysupport.informatica.com](http://mysupport.informatica.com). The How-To Library is a collection of resources to help you learn more about Informatica products and features. It includes articles and interactive demonstrations that provide solutions to common problems, compare features and behaviors, and guide you through performing specific real-world tasks.
Informatica Knowledge Base

As an Informatica customer, you can access the Informatica Knowledge Base at http://mysupport.informatica.com. Use the Knowledge Base to search for documented solutions to known technical issues about Informatica products. You can also find answers to frequently asked questions, technical white papers, and technical tips. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team through email at KB_Feedback@informatica.com.

Informatica Multimedia Knowledge Base

As an Informatica customer, you can access the Informatica Multimedia Knowledge Base at http://mysupport.informatica.com. The Multimedia Knowledge Base is a collection of instructional multimedia files that help you learn about common concepts and guide you through performing specific tasks. If you have questions, comments, or ideas about the Multimedia Knowledge Base, contact the Informatica Knowledge Base team through email at KB_Feedback@informatica.com.

Informatica Global Customer Support

You can contact a Customer Support Center by telephone or through the Online Support. Online Support requires a user name and password. You can request a user name and password at http://mysupport.informatica.com.

Use the following telephone numbers to contact Informatica Global Customer Support:

<table>
<thead>
<tr>
<th>North America / South America</th>
<th>Europe / Middle East / Africa</th>
<th>Asia / Australia</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Toll Free</strong></td>
<td><strong>Toll Free</strong></td>
<td><strong>Toll Free</strong></td>
</tr>
<tr>
<td>Brazil: 0800 891 0202</td>
<td>France: 0805 804632</td>
<td>Australia: 1 800 151 830</td>
</tr>
<tr>
<td>Mexico: 001 888 209 8853</td>
<td>Germany: 0800 5891281</td>
<td>New Zealand: 09 9 128 901</td>
</tr>
<tr>
<td>North America: +1 877 463 2435</td>
<td>Italy: 800 915 985</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Netherlands: 0800 2300001</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Portugal: 800 208 360</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spain: 900 813 166</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Switzerland: 0800 463 200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>United Kingdom: 0800 023 4632</td>
<td></td>
</tr>
<tr>
<td><strong>Standard Rate</strong></td>
<td><strong>Standard Rate</strong></td>
<td></td>
</tr>
<tr>
<td>Belgium: +31 30 6022 797</td>
<td>Belgium: +31 30 6022 797</td>
<td></td>
</tr>
<tr>
<td>France: +33 1 4138 9226</td>
<td>Germany: +49 1805 702 702</td>
<td></td>
</tr>
<tr>
<td>Germany: +49 1805 702 702</td>
<td>Netherlands: +31 306 022 797</td>
<td></td>
</tr>
<tr>
<td>Netherlands: +31 306 022 797</td>
<td>United Kingdom: +44 1628 511445</td>
<td></td>
</tr>
<tr>
<td>United Kingdom: +44 1628 511445</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 1

Understanding Custom Metadata Integration

This chapter includes the following topics:

- Understanding Custom Metadata Integration Overview, 1
- Custom Metadata Integration Process, 2
- Metadata Manager Concepts, 4
- Using the Model Tab, 6
- AccessDB Example, 7

Understanding Custom Metadata Integration Overview

You can use Metadata Manager to load metadata from metadata sources for which Metadata Manager packages a resource type. You can also add custom metadata into the Metadata Manager warehouse.

Metadata Manager uses models to define the metadata it extracts from metadata sources. To add custom metadata to the Metadata Manager warehouse, you must first create or edit the model that defines the metadata. Then, you add or load the resource for the custom metadata to the Metadata Manager warehouse.

You can add the following types of custom metadata into the Metadata Manager warehouse:

- **Metadata for a custom metadata source.** Load or add metadata from a source for which Metadata Manager does not package a resource type. For example, you want to load metadata from a Microsoft Access database. Metadata Manager does not package a resource type for Microsoft Access. Create a custom model for the source metadata and then load or add the metadata into the Metadata Manager warehouse.

- **Properties.** Add custom properties to the metadata in the Metadata Manager warehouse. For example, you want to add a property to a report for Cognos ReportNet. Edit the Cognos ReportNet model in Metadata Manager to add the property and then add the metadata for the property on the Browse tab.

- **Relationships.** Add relationships from custom metadata classes to model classes for which Metadata Manager packages a resource type. For example, a column in a custom metadata source is also used in an Oracle table. Create a class-level relationship between the custom source column and the Oracle table column, and create the object-level relationship on the Browse tab. Create a relationship to run data lineage analysis on the custom metadata.

You create or edit models for Metadata Manager on the Model tab in Metadata Manager. After you create or edit the model for the custom metadata, you add the metadata to the Metadata Manager warehouse. You can add the metadata using the metadata catalog. You can also create a custom resource, create a custom resource template
and generate PowerCenter workflows using the Custom Metadata Configurator, and load the metadata on the Load tab in Metadata Manager.

After you add the custom metadata into the Metadata Manager warehouse, use Metadata Manager or the packaged Metadata Manager reports to analyze the metadata. You can create reports to analyze information. You can also export and import the models, or export and import the metadata that you added to the metadata catalog.

Custom Metadata Integration Process

You can add metadata for a custom metadata source or modify metadata by adding properties or relationships. The tasks you complete depend on the type of metadata you want to add to the Metadata Manager warehouse. You can add metadata for a custom metadata source, add properties for a packaged model, or configure relationships from a custom model to a packaged model.

Adding and Loading Metadata for a Custom Metadata Source

When you add metadata for a custom metadata source, you define a model for the source metadata to define the type of metadata that Metadata Manager extracts. You create the model and add classes, properties, and relationships. After you define the model, add the metadata to the metadata catalog using the Browse tab, or use the Custom Metadata Configurator to create a custom resource template. Then load the metadata into the Metadata Manager warehouse from metadata source files.

Use the Custom Metadata Configurator to create the template and the PowerCenter objects, including mappings, sessions, and workflows, that Metadata Manager uses to extract metadata from metadata source files. Export the metadata from the metadata source to a metadata source file, create a custom resource in Metadata Manager, and load the metadata from the metadata source files.

Adding Custom Metadata

To add metadata for a custom metadata source, complete the following steps:

1. **Create the model.** Create the model to represent the metadata in the metadata source using the Model tab in Metadata Manager.
2. **Add classes, properties, and relationships.** Add custom classes, properties, and relationships to the model using the Model tab.
3. **Add the metadata to the Metadata Manager warehouse.** Create a resource in the metadata catalog that represents the source metadata using the Browse tab. Add custom metadata objects based on the classes that you create.

**RELATED TOPICS:**
- “Creating and Configuring Custom Models” on page 9
- “Creating Custom Resources and Metadata Objects” on page 28

Loading Custom Metadata

To load metadata from metadata source files for a custom metadata source, complete the following steps:

1. **Create the model.** Create the model to represent the metadata in the metadata source using the Model tab in Metadata Manager.
2. **Add classes, properties, and relationships.** Add custom classes, properties, and relationships to the model using the Model tab.

3. **Create the custom resource.** Create a resource for the model on the Load tab.

4. **Create the template and generate PowerCenter objects.** Create a custom resource template and generate the custom PowerCenter objects used by Metadata Manager to load the custom metadata into the Metadata Manager warehouse.

5. **Configure and load the resource.** Configure the custom resource template and metadata source files for the resource and then load the metadata for the custom resource.

This guide uses a Microsoft Access example to demonstrate the custom metadata integration process.

**RELATED TOPICS:**

- “Creating and Configuring Custom Models” on page 9
- “Using the Custom Metadata Configurator” on page 17
- “AccessDB Example” on page 7

### Adding Custom Properties

To add custom properties to metadata in the Metadata Manager warehouse, complete the following steps:

1. **Add the property to the model class.** Add the property to a packaged model class.

2. **Add the metadata for the property.** Edit the metadata object in the metadata catalog to add the metadata for the property.

**RELATED TOPICS:**

- “Configuring Properties” on page 13
- “Editing Properties” on page 30

### Adding Custom Relationships

To add relationships between custom metadata and other metadata objects in the Metadata Manager warehouse, complete the following steps:

1. **Add the relationship to the class for the metadata object.** Add a class-level relationship to the class of the metadata object that you want to create the relationship for.

2. **Create the relationship between the metadata objects.** Edit the metadata object to create an object-level relationship. You create object-level relationships on the Browse tab.

When you add a relationship from one class to another class on the Model tab, you add a class-level relationship. When you create the relationship between metadata objects in the metadata catalog, you create an object-level relationship.
**Metadata Manager Concepts**

Metadata Manager uses the following concepts to define custom metadata in the Metadata Manager warehouse:

- **Models**
- **Classes**
- **Properties**
- **Groups**
- **Relationships**
- **Business Name**

Metadata Manager stores the models, including the associated classes, properties, and relationships in the Metadata Manager repository. You can run Metadata Manager reports in the JasperReports application to get more information about the models, classes, properties, and relationships in the Metadata Manager repository.

**Models**

A model is a group of classes, properties, and relationships designed for a particular type of metadata source. Metadata Manager uses models to classify metadata stored in the Metadata Manager warehouse. When you load metadata into the Metadata Manager warehouse, Metadata Manager extracts the metadata defined in the model.

You can create or edit the following model types in Metadata Manager:

- **Packaged models.** Models that define the metadata that Metadata Manager can extract from specific application, business intelligence, data integration, data modeling, and relational metadata sources. Edit packaged models to add properties.
- **Custom models.** Models you create to define metadata from metadata sources for which Metadata Manager does not package a model.

View the packaged and custom models on the Model tab.

**Classes**

Classes define the types of objects that a metadata source contains. For example, the Source Definition class defined in the PowerCenter model contains PowerCenter source definitions.

View the classes for packaged models and add custom classes for custom models on the Model tab. For more information about packaged classes for specific models in the Metadata Manager repository, run and review the Classes Exploration model report in the JasperReports application.

**Properties**

Properties are characteristics of metadata objects. For example, a metadata object can have a property called Usage that contains a description of how to use the object or how it is used within a metadata source. You can create properties that apply to all objects of a class for a packaged model, or you can add properties that apply to a custom class for a custom model.
Groups

Groups are metadata objects of the same object class type. Metadata Manager can include metadata objects in logical groups in the metadata catalog when it extracts metadata. You can specify a group when you create a class for a metadata object. When you view the metadata for the class in the metadata catalog, Metadata Manager groups objects from the same class type in a logical group.

Objects of different classes can also belong to the same group. Metadata Manager does not store groups as metadata in the Metadata Manager repository.

Relationships

Relationships are associations between two classes or two metadata objects. Create or view the following types of relationships in Metadata Manager:

- **Class-level relationship.** Association between two specific classes. When Metadata Manager extracts metadata from a metadata source, it uses the class-level relationships to create object-level relationships between metadata objects. View relationships between classes and create class-level relationships for custom classes on the Model tab.

- **Object-level relationship.** Association between two metadata objects. View and create object-level relationships between metadata objects on the Browse tab.

Some class relationships establish a parent-child relationship. For example, Metadata Manager includes a class relationship between database tables and columns, where a table is the parent object and a table column is the child object. The metadata catalog uses the parent-child relationships to display objects in a hierarchical structure, where all child objects appear under their corresponding parent object.

You can also establish parent-child relationships when you create classes on the Model tab. Create a class and then create a subclass to establish a parent-child relationship between the parent class and the child class, or subclass.

You can only create relationships from custom classes and custom metadata objects to other classes or metadata objects. You cannot create relationships from classes or metadata objects for which Metadata Manager packages a model. For example, you can create a relationship from a custom metadata object to an Oracle table, but you cannot create a relationship from an Oracle table to any other metadata object type.

View object-level relationships you create in data lineage analysis and in the Details panel on the Browse tab.

For more information about class relationships, run and review the Associations by Package model report in the JasperReports application.

Business Name

A business name is a property used to identify metadata objects according to their business usage instead of the metadata object name in the metadata catalog. For example, a table named CUST_ADDR identifies a customer shipping address. Configure the business name property as Customer Shipping Address to indicate the table contains the customer shipping address.

All classes for packaged and custom models include a business name property.

Edit business name properties for metadata objects in custom or packaged resource types in the metadata catalog. Edit business name properties for metadata objects in a packaged resource type by exporting the properties to an Excel file, editing the property values, and importing the properties from the Excel file into the metadata catalog.
View the business name property for metadata objects in the Details panel on the Browse tab.

Using the Model Tab

The Model tab displays the packaged application, business intelligence, data integration, data modeling, and relational models. It also displays the custom models that you create. Use the Model tab to browse the models in the Metadata Manager repository, create and edit custom models, properties, and relationships, and create properties for packaged models.

The Model tab includes the following components:

- **Model navigator.** Displays all models in the Metadata Manager repository in a hierarchical structure.
- **Content panel.** Displays child classes, folders, and groups of models and classes that you select in the Models navigator.
- **Properties panel.** Displays the class details, properties, and relationships for classes that you open to edit.

You can view details for multiple models and classes simultaneously on the Model tab. Each time you open a class, Metadata Manager opens a tab for the class.

Browsing Models

Use the Model tab to complete the following tasks:

- **Browse the model hierarchy.** Browse all models and view the model hierarchy in the Models navigator.
- **View model and class properties.** View the hierarchy for a specific class and view the model and child classes and properties in the Content and Properties panels. Model properties include name and description. Class properties include the name, description, group name, and if the class shows in a lineage diagram. The packaged models may include folders in the model hierarchy. You can only create classes for custom models.

Editing Models

You can edit model and class properties for custom models and classes. Use the Model tab to perform the following tasks:

- **Create models and edit model properties.** Create or edit a model from the Models navigator.
- **Create and edit classes and class properties and relationships.** Create and edit custom classes, properties, and relationships, view properties and relationships for packaged classes, and add custom properties for packaged models. You can also organize the way in which you want to display the class properties.

To edit or view properties and relationships for a class, you must select the class and choose Actions > Open or select Open from the toolbar, or right-click a class and choose Open.

**Note:** Class properties are called attributes in the Model tab.
AccessDB Example

This book uses an example of a Microsoft Access database to show the concepts for custom metadata integration. The Access database contains tables and views. It includes the model, AccessDB, and the following classes:

- AccessSchema
- AccessTable
- AccessTableColumn
- AccessView
- AccessViewColumn

AccessSchema

Schema class for the tables and views in the Access database. The class is not configured to display in a data lineage diagram. AccessSchema is the root class for the AccessDB model.

The following table describes the properties for this class:

<table>
<thead>
<tr>
<th>Property</th>
<th>Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
</tr>
</tbody>
</table>

AccessTable and AccessView

Table and view classes for the Access database. The classes are configured to display in data lineage. AccessTable and AccessView are child classes of the AccessSchema class. Group names are Tables and Views.

The following table describes the properties for the class:

<table>
<thead>
<tr>
<th>Property</th>
<th>Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>String</td>
</tr>
<tr>
<td>Business Usage</td>
<td>String</td>
</tr>
<tr>
<td>Date Created</td>
<td>Date</td>
</tr>
</tbody>
</table>

AccessTableColumn and AccessViewColumn

Column class for the Access database. AccessTableColumn and AccessViewColumn are configured to display in data lineage and are child classes of AccessTable and AccessView.
The following table describes the properties for the class:

<table>
<thead>
<tr>
<th>Property</th>
<th>Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>String</td>
</tr>
<tr>
<td>Business Usage</td>
<td>String</td>
</tr>
<tr>
<td>Datatype</td>
<td>String</td>
</tr>
<tr>
<td>Length</td>
<td>Integer</td>
</tr>
<tr>
<td>Date Created</td>
<td>Date</td>
</tr>
</tbody>
</table>
Creating and Configuring Custom Models

This chapter includes the following topics:

- Creating and Editing a Custom Model Overview, 9
- Configuring a Model, 10
- Configuring Classes, 11
- Configuring Properties, 13
- Configuring Relationships, 15

Creating and Editing a Custom Model Overview

Create models to define the custom metadata you want to add to the Metadata Manager warehouse. You can create and edit custom models. You can create and edit custom properties for packaged application, business intelligence, data integration, data modeling, and relational models.

To create a model and configure a model, complete the following steps:

1. Create the model. Create the model that will contain the classes for the model in the Model navigator. You can edit model properties in the Properties panel.
2. Configure classes. On the Model navigator, create the classes and subclasses for the metadata objects that you want to add. You can edit class properties in the Properties panel.
3. Configure the class properties. In the Properties panel, configure the properties for each class. You can also configure properties for a packaged class.
4. Configure the class relationships. Open the class to configure the relationships for each class.

You can also configure custom properties for the packaged models.

Note: You cannot create a custom model using an existing business glossary model as template.

After you create and configure a model, add or load the metadata using one of the following methods:

- Load a custom resource. Use the Custom Metadata Configurator to create a custom resource template to load the metadata.
- Add a resource on the Browse tab. Add the metadata objects using the metadata catalog on the Browse tab in Metadata Manager.
Establish the Model Structure

Before you begin to create the model, you must establish the required classes and subclasses, and the properties and relationships between the classes. Identify the following components of the model:

- **Model name and description.** The Model serves as the container for the classes in the model, and appears as a resource in the metadata catalog when you add or load metadata. For example, AccessDB is the model name for the Access database example.

- **Parent classes and subclasses.** The classes represent the metadata objects that you want to add or load into the Metadata Manager warehouse. For example, AccessSchema is the parent class for the Access database example. AccessTable and AccessView are the child classes for the AccessSchema class.

- **Properties and relationships.** Determine the properties for each class and the relationships between the custom metadata classes and classes in the packaged models. For example, in the Access database example, the columns in the AccessTable schema have a relationship to source columns in an Oracle database.

Configuring a Model

You can create a model or edit a model. When you create or edit a model, you configure the model name and description. You can use a custom model as a template for the model.

Creating a Model

Create a model and configure the name, description, and template to base the model on.

To create a model:

1. On the Model tab, click Actions > New > Model.
   
   The New Model window appears.

2. Enter the name and description for the model.

3. Optionally, select a model to use as a template.

4. Click OK.
   
   Metadata Manager creates the model. The model appears in the Model navigator.

Editing a Model

You can edit a model to change the name and description for the model. If you change a model name, Metadata Manager updates the model name in the Resource wizard on the Load tab.

**Note:** You cannot edit model names for packaged models.

To edit a model:

1. On the Model tab, select the model that you want to edit.

2. In the Properties panel, click the Edit icon.

   The Edit Model window appears.

3. Update the name or description.

4. Click OK.
Deleting a Model

You can delete any custom model you create. When you delete a model, Metadata Manager removes any classes, properties, and associations for the model.

You cannot delete a model if you added metadata based on the model. If you loaded metadata using the model, or if you added metadata to the metadata catalog based on the model, purge the metadata from the Metadata Manager warehouse or delete the resource in the metadata catalog before you delete the model.

To delete a model:
1. Select the model that you want to delete in the Model navigator.
2. Click Actions > Delete.
3. Click OK.

Metadata Manager deletes the model and all classes for the model from the Model navigator and the Metadata Manager repository.

Configuring Classes

You can create, edit, or delete classes for custom models. When you create a class, you select the level in the model hierarchy where you want to create the class and configure the class properties. You can edit and delete classes and change the location of the class in the model hierarchy.

The following table describes the class properties and options you can configure:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Name</td>
<td>Name of the class. Do not use the INPUT or OUTPUT reserved words as the class name. The PowerCenter workflow fails to generate.</td>
</tr>
<tr>
<td>Description</td>
<td>Description for the class.</td>
</tr>
<tr>
<td>Group Name</td>
<td>Name of the group under which metadata objects of this class appear in the metadata catalog. Select a group name when you create a class. New group name cannot be edited.</td>
</tr>
<tr>
<td>Icon File Name</td>
<td>Name of the graphic file you want to use as the icon for the class. The icon appears in the Model navigator for the class and in the metadata catalog for any objects based on the class. You must place the graphic file in the following location: &lt;Inforntica installation directory&gt;\services\MetadataManagerService\mappm\images</td>
</tr>
<tr>
<td>Show in Lineage</td>
<td>Displays the class when you perform data lineage analysis on the metadata for the model. You must enable this property to view data lineage for metadata objects of this class.</td>
</tr>
</tbody>
</table>

Creating a Class

Create a class at any hierarchy level for a custom model. Optionally, you can add a previously created class.

For example, in the Access database example, you want to add an AccessTableColumn class as a subclass to the AccessTable class. You must select the AccessTable class to create a subclass for AccessTable.
To create a class:

1. On the Model tab, select the model or class under which you want to create the class.
2. If you select the model or class in the Model navigator, click Actions > New > Class.
   The New Class window appears.
   If you selected a model in the Model navigator, a different New Class window appears.
3. Enter the class properties.
4. Click OK.

Metadata Manager creates the class at the selected level of the hierarchy. After you create the class, configure the class properties and relationships.

Editing a Class

You can edit a class to change the name, description, icon file name, or lineage properties. If you edit a class to show the metadata objects for the class in data lineage, you must load the resource again to view the metadata objects for the class in the data lineage diagram.

To edit a class:

1. On the Model tab, select the class in the Model navigator or Content panel.
2. In the Properties panel, click the Edit icon.
   The Edit Class window appears.
3. Edit the class properties.
4. Click OK.

Metadata Manager updates the class with the changed properties.

Deleting a Class

You can delete a class from a model. If you delete a class and the metadata catalog contains a metadata object for the class, you cannot add more classes of the same type to the catalog. The metadata objects for the class will remain in the catalog.

Note: You cannot delete a class if it has a relationship to another class.

To delete a class:

1. On the Model tab, select the class in the Model navigator or the Content panel.
2. Click Actions > Delete.

Moving and Copying Classes

To move or copy a class from one level of the model hierarchy to another in the Model navigator, drag the class to another location.

Use the following rules and guidelines when you move or copy classes:

- If you drag a class from one level of the hierarchy to another for the same model, Metadata Manager moves the class.
- If you drag a class from one custom model to another custom model, Metadata Manager copies the class.
- If you move or copy a class, Metadata Manager moves or copies all subclasses for the class.
- You cannot drag classes into packaged models.
To move a class:
1. In the Model navigator, select the class you want to move or copy.
2. Drag the class to another location.
   If the location to which you want to move or copy the class is collapsed, move the pointer over the parent class to expand the child classes in the navigator.

## Configuring Properties

You can configure properties for custom classes and classes for packaged models. You can create, edit, and delete properties that apply to all metadata objects for a custom class or a packaged class. Open the class on the Model tab from the Model navigator or the Content panel to configure properties.

The following table describes the properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the property. This name appears for all metadata objects for the class in the metadata catalog when you add or load metadata.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the property.</td>
</tr>
</tbody>
</table>
| Type     | Datatype for the property. You can configure the following datatypes:  
- Integer  
- String  
- Long  
- Date |

After you configure the properties, the method you use to configure the values for the properties depends on the method you use to add or load the metadata:

- **Using the Browse tab.** Use the Browse tab to edit values for the custom properties.
- **Using a custom resource.** If you create a custom resource to load the metadata, you configure the values that Metadata Manager uses for the properties in the Attributes Map tab of the Custom Metadata Configurator.

**Note:** You can also change the order in which properties appear for objects on the Browse tab. To change the order of a property, click the property and drag it to a different location.

**Related Topics:**

- “Using the Custom Metadata Configurator” on page 17
- “Editing Properties” on page 30

## Creating Properties

When you create properties, you configure the name, description, and datatype for the property. By default, Metadata Manager includes Name, Label, Description, and Business Name properties for each class.

1. On the Model tab, select the class for which you want to configure properties.
2. In the Model navigator or the Content panel, click Actions > Open.
   Metadata Manager opens a tab for the class and lists the default properties for the class.
3. Click Actions > Add Attribute and configure the name, description, and datatype of the property.

   The datatype can be Integer, Long, String, Date, or LongString.

4. Click Save.

### Editing Properties

You can edit properties for custom and packaged models. If you change the name of a property for which metadata objects exist in the metadata catalog, Metadata Manager updates the objects to use the property name. If you change the type of a property for which metadata objects exist in the metadata catalog, Metadata Manager does not update the metadata to use the datatype. You must delete the value for type and enter a value that is valid for the datatype.

1. On the Model tab, select the class for which you want to configure properties.
2. In the Model navigator or the Content panel, click Actions > Open.

   Metadata Manager opens a tab for the class and lists the properties.
3. To edit the properties, click the Name, Description, or Type box.

   The datatype can be Integer, Long, String, or Date. For custom properties, the datatype can also be LongString.
4. To organize properties, drag them to change their order or to ensure that they appear in either the Basic or Advanced section of the class properties, in all Metadata Manager perspectives. You can also use Action > Move Up or Action > Move Down to change the order of the class properties.
5. Click Save.

### Organizing Properties

You can organize the way in which you want to display the class properties.

When you edit the properties, you can drag them to change their order or to ensure that they appear in either the Basic or Advanced section of the class properties, in all Metadata Manager perspectives.

If you move the properties below the properties divider, users need to click More in the Properties pane to view these properties in the Advance section. Users need to click Less if they want to hide the properties listed in the Advanced section.

For a class, Source Creation Date, Source Update Date, MM Creation Date, and MM Update Date properties are referred as the synthetic date properties. You can set the Show_Synthetic_Dates_In_Basic_Section property in the imm.properties file to specify if these properties should be located in the Basic or Advanced section.

Synthetic date properties are always grouped together and you cannot change their order. You cannot move any property below the synthetic date properties.

The Class and Location properties are grouped together and you cannot move them below the properties divider.

The Name and Description properties are grouped together and you cannot move any other property above these properties.

### Deleting Properties

You can delete properties for custom classes and custom properties you added for packaged model classes. You cannot delete the default Name and Description properties for any class.

If you added metadata for the property to any metadata object in the metadata catalog, Metadata Manager also removes the metadata for the property. For example, you added a property to an OracleTable class and then
added the metadata for the property on the Browse tab. If you delete the property, Metadata Manager removes the property from all metadata objects of type OracleTable in the metadata catalog.

To delete a property:

1. On the Model tab, select the class for which you want to delete a property.
2. In the Model navigator or the Content panel, click Actions > Open.
   Metadata Manager opens a tab for the class and lists the properties for the class.
3. Select the property and click Actions > Delete Attribute.

Configuring Relationships

Use relationships to create associations between metadata objects. To create an association between two metadata objects, use the Model tab to create a class-level relationship between the classes of the metadata objects that you want to relate. You can only create relationships between custom classes or from custom classes to classes for which Metadata Manager packages a model.

Use the Model tab to create a class-level relationship. When you create a relationship, you configure the relationship properties and the model and class to which you are creating the relationship.

The following table describes the relationship properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the relationship.</td>
</tr>
<tr>
<td>Description</td>
<td>Description for the relationship.</td>
</tr>
<tr>
<td>Model</td>
<td>Model of the class that you want to create the relationship to. Use the Any Model option for this property for the most flexibility in creating object-level relationships to objects of this class on the Browse tab.</td>
</tr>
<tr>
<td>Class</td>
<td>Class you want to create the relationship to.</td>
</tr>
<tr>
<td>Direction</td>
<td>Establish a parent-child relationship between the two classes. You can select one of the following options:</td>
</tr>
<tr>
<td></td>
<td>- &gt;&gt;. The class to which you create the relationship is the parent class.</td>
</tr>
<tr>
<td></td>
<td>- &lt;&lt;. The class to which you create the relationship is the child class.</td>
</tr>
</tbody>
</table>

After you create the class-level relationship on the Model tab, you create the relationship between the metadata objects for the two classes. You can create object-level relationships using the following methods:

- **Using the Custom Metadata Configurator.** If you use Metadata Manager to load the custom metadata, establish the object-level relationships in the Associations tab of the Custom Metadata Configurator. When you load the custom metadata, Metadata Manager creates the object-level relationships. Use this method to create object-level relationships between metadata objects in a custom metadata source.

- **Using the Browse tab.** Use the Related Catalog Objects section on the Browse tab to create object-level relationships between custom metadata objects and other metadata objects. Use this method to create object-level relationships between custom metadata objects and metadata objects for which Metadata Manager packages a model.

For example, a column in the Access database comes from an Oracle table that exists in the metadata catalog. You want to create a relationship between the metadata objects for the two columns. First, create a relationship between the AccessTableColumn class and the OracleColumn class on the Model tab. After you load the
metadata for the Access database, create an object-level relationship on the Browse tab between the two metadata objects.

Creating Class-Level Relationships

When you create a class-level relationship, you configure the model and class to which you want to create the relationship.

To create a relationship:
1. On the Model tab, select a class.
2. In the Model navigator or the Content panel, click Actions > Open.
   Metadata Manager opens a tab for the class.
4. Click Actions > Add Relationship and configure the relationship properties.
5. Click the Save icon.

Editing and Removing Class-Level Relationships

Edit a relationship to change any of the properties or delete a relationship. If you edit a relationship and change the model, class, or parent-child relationships, Metadata Manager does not remove object-level relationships. The object-level relationships that you create use the updated properties.

For example, you edit the relationship for the AccessTableColumn class and remove the OracleColumn class from the relationship. Existing object-level relationships in the metadata catalog do not change, but you cannot create additional object-level relationships from AccessTableColumn objects to the OracleColumn metadata objects.

If you remove a relationship, Metadata Manager does not remove object-level relationships.

To edit a relationship:
1. On the Model tab, select a class.
2. In the Model navigator or the Content panel, click Actions > Open.
   Metadata Manager opens a tab for the class.
4. Edit the relationship properties.
5. Click the Save icon.
6. Optionally, select the relationship and click Actions > Delete Relationship.
Using the Custom Metadata Configurator Overview

Use the Custom Metadata Configurator to load metadata for a custom model from metadata source files. Use the Custom Metadata Configurator to configure the custom resource template that contains the format of the metadata source files and generate the custom PowerCenter objects in the PowerCenter repository. The PowerCenter objects include the sessions, mappings, and workflows that Metadata Manager uses to load the metadata from the metadata source files.

Complete the following tasks to create a template and the custom PowerCenter objects:

1. **Create the metadata files.** Create the element and association metadata files.
2. **Log in to the Custom Metadata Configurator.** Use the Metadata Manager repository connection information to log in to the Custom Metadata Configurator.
3. **Configure the template.** Configure the custom resource template name and template properties.
4. **Configure delimiters.** Configure the delimiters for the metadata files.
5. **Map attributes.** Map attributes in the element metadata file to the attributes in the custom model.
6. **Map relationships.** Map relationships in the association metadata file to the relationships for the classes in the custom model.
7. **Add class rules.** Add the class rules to select particular records in an element or association metadata.
8. **Generate the PowerCenter objects.** Generate the PowerCenter objects required to load metadata from the metadata source files into the Metadata Manager warehouse.

**Note:** The Custom Metadata Configurator refers to relationships as associations.

After you create the custom resource template and generate the PowerCenter objects, you configure the custom resource you create on the Load tab and configure the template name and metadata source files.

**Step 1. Create Metadata Source Files**

Before you log in to the Custom Metadata Configurator, create the element metadata file and the association metadata files that contain the metadata source information. You use the files to map attributes and associations to the classes you created for the model on the Load tab.

Create the following metadata source files:

- **Element metadata file.** Contains metadata object names and attributes. The metadata objects are instances of the classes you defined in the model. You map the object attributes in the element metadata file to the class attributes defined in the corresponding model you created on the Model tab.

- **Association metadata file.** Contains relationships between metadata objects in an association metadata file. The relationships between metadata objects are instances of the class associations you defined in the model. You use the Custom Metadata Configurator, which exposes the information in the association metadata file, to map the associations between the objects.

If you want to preview the data in the element and association metadata files in the Custom Metadata Configurator, you must map the PATH environment variable to the location of the Microsoft Excel excel.exe executable.

**Metadata Source File Rules and Guidelines**

Use the following rules and guidelines to create the element and association metadata source files:

- You can store the metadata object attributes and associations together in the same file or in separate files.

- The association metadata file must identify the From and To objects in each association. It should store one or more object attributes to enable you to uniquely identify each From and To object in the file and store object attributes in separate columns.

- The custom resource fails to load some metadata objects, attributes, and relationships if the metadata sources file names contain any spaces or the following characters:
  
  ~ " ! \ ^ & * ( ) _ + = { } [ ] ; " ' < > , . ? /

  If the column names contain spaces or any of these characters, the Custom Metadata Configurator converts them to underscores when you generate the PowerCenter objects. After converting the characters and spaces to underscores, columns may have the same modified name. The Custom Metadata Configurator ignores one of the columns when you generate the workflows.

- The association and element metadata file names cannot start with a number. If the file name starts with a number, PowerCenter object generation fails.
Step 2. Log In to the Custom Metadata Configurator

To log in to the Custom Metadata Configurator, specify the Metadata Manager warehouse database that contains the custom model for the metadata source files you want to load.

The following table describes the properties you must configure to log in to the Custom Metadata Configurator:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Connections</td>
<td>Contains a list of connect strings used to connect to multiple Metadata Manager warehouses. Select a Metadata Manager warehouse from the list, or configure another warehouse.</td>
</tr>
<tr>
<td>User ID</td>
<td>User account for the Metadata Manager warehouse database.</td>
</tr>
<tr>
<td>Password</td>
<td>Password for the Metadata Manager warehouse database user account.</td>
</tr>
<tr>
<td>Database Type</td>
<td>Type of database for the Metadata Manager warehouse database.</td>
</tr>
<tr>
<td>Connection String</td>
<td>JDBC connection string for the Metadata Manager warehouse database. The connection string depends on the database type you select. For example, if you use an Oracle database, use the following connection string: jdbc:informatica:oracle://[host]:1521;SID=[sid] For Oracle databases, you can enter the SID or edit the string to use the full service name. For example: jdbc:informatica:oracle://[host]:1521;ServiceName=[service name] To authenticate the database user credentials using the Windows protocol and establish a trusted connection to a Microsoft SQL Server repository, append the following text to the connection string: AuthenticationMethod=ntlm. For example: jdbc:informatica:sqlserver://[host]:[port];DatabaseName=[DB name];AuthenticationMethod=ntlm</td>
</tr>
</tbody>
</table>

To log in to the Custom Metadata Configurator:

1. Click Start > Programs > Informatica <version> > Client > PowerCenter Client > Custom Metadata Configurator.
   The Custom Metadata Configurator login window appears.
2. Enter the connection properties.
3. Click OK.

You can override the default database code page for the Metadata Manager warehouse database when you log in. Override the code page if the custom resource templates contain characters that the database code page does not support. For example, the Custom Metadata Configurator does not retrieve saved templates correctly. To override the code page, add the CODEPAGEOVERRIDE parameter to the connection string and specify a compatible code page.

For example, use the following JDBC URL to override the default code page with MS932:

```
jdbc:informatica:sqlserver://myhost:1433;DatabaseName=mm861;CODEPAGEOVERRIDE=MS932;
``` 

Step 3. Configure the Custom Resource Template

A custom resource template stores information about how to map the metadata object attributes to the class attributes and can also store the class relationships between metadata objects. Create a template using attribute and association files.
Map the object attributes in an element metadata file to the class attributes configured in the model. Map class relationships defined in the model between objects in the custom source repository. Save the associations in a template.

Creating a Custom Resource Template

Create a template on the Configure Maps tab. Enter the configuration information for the template, select the classes to map, and select the metadata element and association files.

The following table describes the template properties you configure:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Name</td>
<td>Name of the template. The name must contain alphanumeric characters and cannot contain spaces. Maximum length is 255 characters. If you create multiple templates, each template name must be unique.</td>
</tr>
<tr>
<td>Repository Name</td>
<td>Custom resource you create on the Load page in Metadata Manager. If you load metadata objects from multiple custom metadata sources, select one of the custom resources.</td>
</tr>
<tr>
<td>Classes</td>
<td>Classes created in the model for the selected custom source repository. Use this list of classes to map the class attributes to the element metadata file.</td>
</tr>
<tr>
<td>Element Metadata File</td>
<td>Contains the attributes for each metadata object. Click Configure to configure the delimiters for the element metadata file. Click Preview to preview the element metadata file in Microsoft Excel.</td>
</tr>
<tr>
<td>Association Metadata File</td>
<td>Contains the information required to establish the associations between metadata objects in the metadata file. Click Configure to configure delimiters for the associations file. Click Preview to preview the metadata file in Microsoft Excel.</td>
</tr>
</tbody>
</table>

**Note:** You must create another template if the format of the metadata source files change after you create the template.

To create a custom resource template:

1. In the Custom Metadata Configurator, click the Template Summary tab.
2. Click Configure New Template.
   - The Configure Maps tab appears.
3. Enter the template properties.
4. Click Select to select the classes from the element metadata file to include in the template.
5. Optionally, click Preview to preview the metadata source file data in Microsoft Excel.
6. Optionally, click Clear All to delete all values set in the template.
   - Use Clear All to delete all information in the template, including the attribute and association maps. This option is available if you have not saved the template.
7. Click Save.
Editing and Deleting a Custom Resource Template

You edit a template after you save it to perform the following tasks:

- Add, edit, or delete class attribute maps to metadata objects in the element metadata file.
- Add, edit, or delete association maps between metadata objects.
- Delete classes from a template.
- Configure delimiters for the element and association metadata files.
- Change a class rule.

If you edit classes for a template, the changes occur in the Metadata Manager warehouse when you load the resource. Because of this, the metadata in the Metadata Manager warehouse matches the metadata in the element and association metadata files.

You can delete a template if it becomes obsolete. If you delete a template, you can purge the metadata loaded by the custom resource from the Metadata Manager warehouse.

To edit or delete a custom resource template:

1. Log in to the Custom Metadata Configurator and select the Metadata Manager warehouse you used to create the template.
2. Click the Template Summary tab.
3. Click View/Edit for the template you want to edit.
   The Configure Maps tab appears.
4. Edit the template.
   You can edit the template properties, and you can edit the attribute and relationship maps.
5. Click Save.
6. Optionally, click Delete to delete the template.

Viewing a Custom Resource Template Summary

After you create and save a template, you can view the template under the corresponding custom resource on the Template Summary tab of the Custom Metadata Configurator.

To view a custom resource template:

1. Log in to the Custom Metadata Configurator.
   The Template Summary tab appears. It displays one of the following statuses for each template:
   - **Saved.** Indicates you saved the template, but did not generate the PowerCenter objects.
   - **Last generated date.** The date you last generated the PowerCenter objects.
2. Select a template, and then click View/Edit.
   The template appears on the Configure Maps tab.
Step 4. Configure Delimiters for Files

To make sure that the Custom Metadata Configurator reads the element and association metadata files, configure the delimiters in the files. Use the Configure Delimiters window to configure the metadata source file delimiters.

The following table describes the properties you configure:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start in Row</td>
<td>First row in the file that contains column headers, metadata, or relationships. Exclude preliminary rows that contain header information or no information. You must enter an integer greater than 0. Default is 1.</td>
</tr>
<tr>
<td>Header Row</td>
<td>File has a header row. Header rows of element metadata files are mapped to class attributes. Header rows of association metadata files are mapped to relationship attributes. Default is enabled.</td>
</tr>
<tr>
<td>Delimiter</td>
<td>Character that separates entries in the file. Default is a comma (,).</td>
</tr>
<tr>
<td>Text Qualifier</td>
<td>Character used to enclose text that should be treated as one entry. Use a text qualifier to disregard the delimiter character within text. Default is quotes (&quot;).</td>
</tr>
</tbody>
</table>

Some element or association metadata files can have extra records that the Custom Metadata Configurator ignores. For example, the first five records in an element metadata file provide a description about the type of information in the file. Record six is the first record that contains object attributes. You can direct the Custom Metadata Configurator to start in row six.

Specify whether the element and association metadata files contain column headers. If the file does not have column headers, the Custom Metadata Configurator displays generic headers for each column, such as Column 1, Column 2, and Column 3.

Since you provide the object attribute and association information in flat files, you must specify the delimiter between records.

You can also specify the text qualifier to escape the delimiter character if it is used within text. For example, you specify a comma (,) as the delimiter in an element metadata file.

To configure delimiters:

1. On the Configure Maps tab, click Configure for the element or association metadata file. The Configure Delimiters window appears.
2. Configure the delimiter properties.
3. Click Set Default to set the current settings as the default for all element and association metadata files that have no settings.
4. Click OK.
5. Click Save.

Step 5. Map Class Attributes

The element metadata file stores the attributes for each metadata object. When you map the object attributes to the class attributes, you map the column of the element metadata file to a class attribute. To map an attribute of
an object to a class attribute, identify the class of the object. Then, map the attribute of the object to the attribute defined for the selected class.

Each column in the element metadata file contains an object attribute. To map the object attribute to the class attribute, map the header of each column in the element metadata file to a class attribute.

If the file does not contain column headers, the Custom Metadata Configurator displays generic header names, such as Column 1, Column 2, and Column 3.

Each model class has a Name attribute. Map an object attribute to the Name attribute for every class. You do not need to map object attributes to other class attributes.

You must also establish the primary key of the element metadata file. The Custom Metadata Configurator uses the primary key to identify each record in the element metadata file for a given class. The key can be a composite value if you need to specify multiple columns in the element metadata file to uniquely identify each record.

One element metadata file can contain object attributes for multiple classes of objects. You can map a column in the element metadata file to a class attribute that is common to multiple classes. For example, the element metadata file contains the following metadata objects:

<table>
<thead>
<tr>
<th>Object</th>
<th>Object_Description</th>
<th>Object Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>Customer information.</td>
<td>Database Table</td>
</tr>
<tr>
<td>Address</td>
<td>Customer address.</td>
<td>Database Column</td>
</tr>
<tr>
<td>Product</td>
<td>Product information.</td>
<td>Database Table</td>
</tr>
</tbody>
</table>

The Object_Description column in the element metadata file describes all column and table metadata objects. In the Custom Metadata Configurator, you map the Object_Description column to the Description class attribute for the AccessTableColumn class. Next, you click Apply to All. The Custom Metadata Configurator also maps the Object_Description column to the Description class attribute for the AccessTable class.

To map class attributes:

1. On the Configure Maps tab of the Custom Metadata Configurator, click Map Information.
   The Attributes Map tab appears.
2. For each class, map the columns in the element metadata file to the class attributes.
   To map an object attribute to the class attribute, select the class attribute from the Class Attributes column for each applicable column in the File Columns column.
   Click the field to display the list of possible values for the Class Attributes columns.
   **Note:** You must map the Name attribute for all classes.
3. Select the Key option for all element metadata file columns that are used to identify each record in the element metadata file.
4. Click Apply to All to apply the common class attribute settings to all other classes.
5. For each class, click Add Rule to select the records in the element metadata file that apply to the attribute map for a particular class.
   You must create a rule for each class if you map multiple classes.
Step 6. Map Class Relationships

The association metadata file stores relationships between objects. You can establish relationships between metadata objects by specifying an association between them. In the Custom Metadata Configurator, you can choose any relationship that belongs to the classes of the two objects involved in the relationship. When you set the class relationship between two metadata objects, you specify the following information:

- The metadata source and class of each object in the relationship
- The From and To objects that participate in the relationship

Select all columns required to uniquely identify each object in the association metadata file.

You can create a relationship between two objects from the same metadata source. You cannot create relationships between a class in a custom metadata source and a class in a resource for which Metadata Manager packages a resource type. Use the Model page in Metadata Manager to create a class-level relationship between a custom class for which Metadata Manager packages a resource type, and then create the object-level relationship on the Browse page.

You configure the relationship maps on the Associations Map tab.

The following table describes the relationship properties you configure:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Repository</td>
<td>Metadata source containing the From element.</td>
</tr>
<tr>
<td>From Class</td>
<td>Class of the From element.</td>
</tr>
<tr>
<td>From Element</td>
<td>Metadata object. Select the columns in the associations file that uniquely identify the From element in the association.</td>
</tr>
<tr>
<td>Association Type</td>
<td>Class association between the From class and To class.</td>
</tr>
<tr>
<td>Struct</td>
<td>Displays a link between the two associated objects in the lineage diagram. You can select either Struct or Field. Use if both objects are similar to data structures. The associated objects contain child objects.</td>
</tr>
<tr>
<td>Field</td>
<td>Displays a link between the two associated objects in the lineage diagram. You can select either Struct or Field. Use if both objects are fields. The associated objects do not contain child objects.</td>
</tr>
<tr>
<td>To Repository</td>
<td>Metadata source containing the To element. Either the From repository or the To repository must refer to the metadata source containing the custom metadata.</td>
</tr>
<tr>
<td>To Class</td>
<td>Class of the To element.</td>
</tr>
</tbody>
</table>
To map class associations:

1. In the Configure Maps tab of the Custom Metadata Configurator, click Map Information.
2. Click the Associations Map tab.
   The grid appears for the association map.
3. Click Add to add an association.
4. Map the associations between each related From and To class.
5. To display values for a cell in the grid, click the cell.
6. To delete a relationship, highlight the relationship and click Delete.
7. Click Clear All to remove all relationships from the grid.
   If you click Clear All, you cannot undo it.

**Related Topics:**
- “Step 7. Add Class Rules to Files” on page 25

## Step 7. Add Class Rules to Files

Use class rules to select specific records in an element or association metadata file for an attribute or association map. Configure class rules in the Rules Setup window.

The following table describes the class rule properties you can configure:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Column</td>
<td>Column in the element or association metadata file that you use to filter the records in the attribute or association map.</td>
</tr>
<tr>
<td>Operator</td>
<td>Operator you want to apply between the selected file column and the entered value.</td>
</tr>
<tr>
<td>Value</td>
<td>Value used to evaluate the operation.</td>
</tr>
</tbody>
</table>

If a file has different classes or an associations file has more than one association, use at least one column to identify each record in the file. Use a class rule to filter records in the file that do not apply to the class attribute or association map.

For example, an element metadata file contains objects for the following Microsoft Access database classes:

- AccessTableColumns
- AccessTables
- AccessSchema
The following table shows some of the records in the element metadata file:

<table>
<thead>
<tr>
<th>Repository Name</th>
<th>Type</th>
<th>Object Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS_DB_SOURCE</td>
<td>Column</td>
<td>SRC_PRODUCT_ID_FK</td>
<td>Foreign key to the SRC_PRODUCTS table.</td>
</tr>
<tr>
<td>ACCESS_DB_SOURCE</td>
<td>Column</td>
<td>SRC_ORDER_QTY</td>
<td>Number of items sold for a given sales order.</td>
</tr>
<tr>
<td>ACCESS_DB_SOURCE</td>
<td>Table</td>
<td>SRC_CUSTOMERS</td>
<td>Contains information about customers from website registry.</td>
</tr>
<tr>
<td>ACCESS_DB_SOURCE</td>
<td>Table</td>
<td>SRC_PRODUCTS</td>
<td>Contains information about products.</td>
</tr>
<tr>
<td>ACCESS_DB_SOURCE</td>
<td>Table</td>
<td>SRC_ORDERS</td>
<td>Contains information about orders made by customers using the website.</td>
</tr>
<tr>
<td>ACCESS_DB_SOURCE</td>
<td>Repository Name</td>
<td>ACCESS_DB_SOURCE</td>
<td>Contains operational data store records for customers, sales, and products.</td>
</tr>
</tbody>
</table>

Create a rule using the Type column in the element metadata file to select the records that apply to each class. For example, the AccessTables class should only contain Microsoft Access database table objects. You can create a rule for the AccessTables class that selects the records based on the following condition:

Type = 'Table'

Create complex filter statements that involve more than one filter condition. To create a complex filter statement, use AND and OR to join two different conditions and use parenthesis to group conditions. For example, you can create the following complex filter statement:

(FileColumnA = 'MappingInstances' OR FileColumnA = 'MappingShortcuts')
AND (FileColumnB = 'PowerCenterDemoRepository')

If you map multiple classes for a custom resource template, specify a rule for each class. If you do not specify a class rule when mapping multiple classes for a template, the following error displays when you save the template:

Class Rule is not defined.

If you map only one class for a template, you do not have to provide a class rule.

You can create one rule for each class or association in a template.

To add a rule to a file:

1. Click Add Rule.
   The Rules Setup window appears.
2. Configure the rule properties.
3. Click Add to Filter to add the selected filter.
4. Use the following options to create complex filter statements:
   - **AND.** Use to create an intersection statement.
   - **OR.** Use to create a union statement.
   - () . Use to group statements.
5. Click Clear to clear the filter.
Step 8. Generate the PowerCenter Objects

After you finalize the custom resource template, use the Custom Metadata Configurator to create the PowerCenter objects. The PowerCenter objects include the workflows that extract the metadata from the element and association metadata files and load it into the Metadata Manager warehouse.

When you generate the PowerCenter workflows, the Custom Metadata Configurator creates the following PowerCenter objects:

- **PowerCenter mappings.** One mapping for the class attribute map and another for the association map for each template.
- **PowerCenter sessions.** One session for each PowerCenter mapping. It includes all sessions in a PowerCenter workflow.
- **PowerCenter workflows.** One PowerCenter workflow for each template.

The Custom Metadata Configurator stores the PowerCenter objects in an XML file, and then imports the XML file into the PowerCenter repository.

The following table lists the naming convention for the generated PowerCenter objects:

<table>
<thead>
<tr>
<th>Component</th>
<th>Naming Convention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow</td>
<td>WF_&lt;repository_name&gt;_template_name_Custom_MetaData</td>
</tr>
<tr>
<td>Mapping</td>
<td>M_&lt;repository_name&gt;_template_name_Elmnt_Attr</td>
</tr>
<tr>
<td></td>
<td>M_&lt;repository_name&gt;_template_name_Elmnt_Assoc</td>
</tr>
<tr>
<td>Session</td>
<td>S_&lt;mapping_name&gt;</td>
</tr>
</tbody>
</table>

To generate PowerCenter workflows:

1. In the Custom Metadata Configurator, click the Template Summary tab.
2. Click View/Edit for the template you want to view.
   The template appears on the Configure Maps tab.
3. Click Generate Workflow to generate the PowerCenter workflows used to extract and load the custom metadata.
CHAPTER 4

Working with Custom Metadata

This chapter includes the following topics:
- Creating Custom Metadata Overview, 28
- Creating Custom Resources and Metadata Objects, 28
- Editing Properties, 30
- Exporting and Importing Custom Properties, 31

Creating Custom Metadata Overview

You can create custom metadata in the Metadata Manager warehouse. After you create a model, create a
resource that represents the model and create metadata objects for the resource using the metadata catalog.

You can also edit the values for custom properties. You can edit metadata object properties you created for
custom classes and properties you created for the packaged application, business intelligence, data integration,
data modeling, and relational classes. For example, you created a custom attribute in a report class for Cognos
ReportNet. You can edit the value for the report property.

You can create and edit metadata in the Metadata Manager warehouse using the following methods:
- Create a custom resource and custom metadata objects. Use the metadata catalog to add a resource
  based on a custom model, and add metadata objects defined by classes in the model.
  After you create the resource and metadata objects, you can edit the properties and relationships for the
  metadata objects.
- Edit custom properties. Edit the values of custom properties for metadata objects.

You can also create object-level relationships for metadata objects after you create class-level relationships for
custom classes on the Model tab.

Note: You can also use the Custom Metadata Configurator and the Metadata Manager Load tab to create a
custom resource, create a template for a custom model and generate the PowerCenter objects required to load
the metadata, and load the metadata for the resource.

Creating Custom Resources and Metadata Objects

Create a model with classes, properties, and relationships on the Model tab. After you create the model, you can
create a resource based on the model to the Metadata Manager warehouse and add metadata objects based on
the classes you created. Create resources and metadata objects using the metadata catalog on the Browse tab in Metadata Manager.

To create a custom resource based on a custom model, complete the following steps:

- **Create the custom resource.** Create the resource based on the model for the resource.
- **Create custom metadata objects.** Create the metadata objects based on the model classes. You can create classes and subclasses.

You can edit or delete resources and metadata objects after you create them.

### Creating a Resource

Create a resource based on a custom model. All the child classes of the resource will appear under the resource name in the metadata catalog. Create the resource using the Create Custom Metadata window.

The following table describes the Create Custom Metadata properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the custom resource.</td>
</tr>
<tr>
<td>Description</td>
<td>Description for the custom resource.</td>
</tr>
<tr>
<td>Preview of Model Hierarchy</td>
<td>Lists the child classes for the selected model. You can create the metadata objects based on these classes after you create the resource.</td>
</tr>
</tbody>
</table>

To create a custom resource:

1. In the Catalog view on the Browse tab, click Actions > New > Custom Metadata. The Create Custom Metadata window appears.
2. On the left pane, select the model you want to base the custom resource on.
3. Configure the resource properties.
4. Optionally, view the model hierarchy and the classes included in the model you base the custom resource on.
5. Click OK.

Metadata Manager creates the resource and the resource appears in the metadata catalog.

### Create Custom Metadata Objects

After you create a custom resource, you can add custom metadata objects to the resource. Add custom metadata objects based on the classes for the model. When you create the custom metadata object, select the class you want to create the metadata object for. Add metadata objects based on the model hierarchy you created on the Model tab.

For example, when you add a metadata object to the AccessDB resource, you add metadata objects of type AccessSchema, because the AccessSchema class is the root class of the model.

Create custom metadata objects in the Create Custom Objects window. You can create multiple metadata objects using this window.

By default, Metadata Manager displays Untitled as the name for a metadata object before you configure it. The properties you configure depend on the properties you created for the class. For example, the AccessSchema class includes the Name, Label, and Description properties.
To create a custom metadata object:

1. In the Catalog view, select the metadata object or resource for which you want to create child objects.
2. Click Actions > New and select the name of the class for which you want to create the metadata object.
   The Create Custom Objects window appears.
3. Configure the metadata object properties.
4. Click Add.
   Metadata Manager adds the metadata object to the Create Custom Objects window.
5. Repeat steps 3 to 4 for each metadata object you want to add.
6. Click the arrow icons to navigate between metadata objects you created to configure the properties.
7. Optionally, select a metadata object you created and click Delete to delete the object.
8. Click OK.
   Metadata Manager adds the metadata objects to the metadata catalog.

Deleting Resources and Metadata Objects

You can delete any custom resource or custom metadata object from the metadata catalog. You can delete resources and metadata objects based on the permissions for the resource and metadata objects.

You can delete resources or metadata objects for which you have write permission on the resource or metadata object and all the child objects. If you do not have write permission on all the child objects, Metadata Manager deletes any child objects on which you have write permission that do not have any child objects.

For example, you created a custom resource, AccessDB, with a schema object ACCESS_DB_SOURCE. ACCESS_DB_SOURCE contains a child table object named CUSTOMERS. CUSTOMERS has no child objects. If you have write permissions on the resource and all objects and delete the resource, Metadata Manager deletes the resource and all child objects. However, if you have read permission on ACCESS_DB_SOURCE and write permission on CUSTOMERS, and delete the resource, Metadata Manager only deletes CUSTOMERS.

You can use the Actions menu or right-click menu to delete a resource or object from the following Catalog view on the Browse tab.

To delete a custom resource or custom metadata object:

1. Select the metadata object or resource you want to delete.
2. Click Actions > Delete.

Editing Properties

You can edit custom metadata object properties or metadata object properties you added to packaged models on the Model tab. Edit custom metadata object properties and business name properties in the Edit Metadata window. The Edit Properties window displays different properties and options depending on the object types you edit. You can edit properties for metadata objects, categories, and business terms.

You can also edit custom metadata object properties you added to packaged model classes and business name properties for packaged resource types by exporting the properties to an Excel file. You can then edit the properties in the Excel file and import the properties into the metadata catalog.
You can use the Actions menu or right-click menu to edit properties for a single object from the following areas on the Browse tab in Metadata Manager:

- Shortcuts view
- Catalog view
- Glossary view
- Details panel
- Data lineage analysis

To edit properties for an object:

1. Select the object whose properties you want to edit.
2. Click Actions > Edit Properties.
   The Edit Properties window appears.
3. Edit the applicable properties.
4. Click OK.

Exporting and Importing Custom Properties

You can edit the values for custom properties you add to packaged model classes and business name properties using Microsoft Excel. Export custom properties and business name properties from the metadata catalog to an Excel file. Use the Excel file to edit values for the properties and import the properties from the Excel file into the metadata catalog.

The Excel file contains a worksheet for each object type you export. Each worksheet contains properties for all metadata objects for a specific object type.

The following table describes the content of each worksheet:

<table>
<thead>
<tr>
<th>Row or Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export Root Path</td>
<td>Metadata catalog root path for the metadata objects in the worksheet. Do not edit the Export Root Path.</td>
</tr>
<tr>
<td>Class Identifier</td>
<td>Name and path of the class in the Metadata Manager repository for the metadata objects in the worksheet. Do not edit the Class Identifier.</td>
</tr>
<tr>
<td>Element ID</td>
<td>Resource and object name for a metadata object. Metadata Manager displays the Element ID in the following format: &lt;resource name&gt;.&lt;object name&gt;.</td>
</tr>
<tr>
<td>Element Path</td>
<td>Hierarchical path for a metadata object in the metadata catalog.</td>
</tr>
<tr>
<td>Business Name</td>
<td>Business name property for a metadata object.</td>
</tr>
<tr>
<td>Custom Attribute Name</td>
<td>Custom property for a metadata object.</td>
</tr>
</tbody>
</table>

Exporting Custom Properties

When you export custom and business name properties, Metadata Manager exports the property values for the selected metadata objects and any child objects to the Excel file.
To limit the custom and business name properties that Metadata Manager exports, configure catalog preferences. Metadata Manager only exports the resources and object types you configure in catalog preferences.

To export custom and business name properties:

1. On the Browse tab, configure preferences to limit the object types you want to export.
2. In the Catalog view, select the resource, logical group, or metadata object for which you want to export properties.
3. Click Actions > Export Metadata > Excel.
4. Click Yes in the Note window to include the business name property.
5. Open or save the Excel file.
   The options to save or download in the Excel file depend on your browser.

**Editing Custom Properties**

Open the Excel file to edit custom and business name properties.

To edit custom and business name properties:

1. In Microsoft Excel, open the Excel file that contains the exported properties.
2. Select the worksheet that contains the properties for the class type of the objects you want to edit.
3. In the row that contains the object for which you want to edit properties, enter the property value in the appropriate column.
4. Repeat steps 2 to 3 for all object types and properties that you want to edit.
5. Save the Excel file.

**Importing Custom Properties**

Metadata Manager updates the properties for the objects in the metadata catalog with the custom or business name properties in the Excel file.

Metadata Manager does not import properties for objects that no longer exist in the catalog. If you delete an object from the catalog, Metadata Manager ignores the properties for the object when you import properties.

To import custom and business name properties:

1. In the Catalog view, click Actions > Import Metadata > Excel.
   The Import Catalog Metadata window appears.
2. Click Browse and select the Excel file.
3. Click Import.
   Metadata Manager imports the properties and displays the number of metadata objects that were updated, not changed, not found, or not valid.

**Rules and Guidelines for Exporting and Importing Custom Properties**

Use the following rules and guidelines when you work with Excel files:

- If you export a large number of objects for which there are custom or business name properties, you cannot perform any operation in Metadata Manager until all the properties are exported.
- If the number of metadata objects in a class is greater than the number of rows in an Excel spreadsheet, Metadata Manager does not export all objects for the object type. Metadata Manager can export a maximum of 65,536 objects for each worksheet.
Migrating Custom Metadata

This chapter includes the following topics:
- Migrating Custom Metadata Overview, 33
- Step 1. Migrate the Models, 34
- Step 2. Migrate Custom Metadata, 35
- Step 3. Migrate Custom Resource Templates, 36
- Step 4. Configure and Load the Custom Resources, 37

Migrating Custom Metadata Overview

You can migrate metadata between Metadata Manager instances. Use Metadata Manager to export models, custom metadata objects, and custom resource templates from one Metadata Manager repository to another, and then load the metadata. Migrate custom metadata to avoid re-creating the metadata in a Metadata Manager instance.

For example, when you move from a development to a production environment, migrate any customization you added to the development environment to the production environment and then load all resources in the production environment.

To migrate custom metadata between source and target Metadata Manager instances, complete the following steps:

1. **Create the new resource.** The resource represents the metadata source that you will import.
2. **Migrate models.** Export models from the source Metadata Manager repository and import them into the target repository. Migrate custom models and custom attributes added to packaged models.
3. **Migrate custom metadata objects and properties.** Export any custom metadata objects and properties you added to the source Metadata Manager repository using the metadata catalog and import it into the target repository. You can migrate metadata objects for custom resources and custom metadata you added to packaged resources.
4. **Migrate custom resource templates.** Export custom resource templates from the source Metadata Manager repository and import them into the target repository.
5. **Configure and load custom resources.** Generate PowerCenter objects and then configure and load custom resources.

You can perform these steps in Metadata Manager or use the Metadata Manager command line to create a script that automatically performs the migration.

**Note:** Metadata Manager does not have commands to migrate custom resource templates. Use Metadata Manager to migrate the templates.
Step 1. Migrate the Models

To migrate custom models or modified packaged models, you export the models from the source Metadata Manager repository to an XML file. You then import the XML file into the target Metadata Manager repository.

When you export a model, the XML file contains all the classes, attributes, and relationships for the model. When you import a model, Metadata Manager analyzes the contents of the XML file and compares it to the existing models in Metadata Manager. Metadata Manager imports the models, classes, and relationships in the XML file that do not exist in the Metadata Manager repository.

For example, you export a custom model from a development environment and then import it into a production environment. If you add a class to the custom model in the development environment and export and import it again, Metadata Manager imports the class you added.

Exporting a Model

Export a custom model or custom attributes for a packaged model to an XML file. On the Model tab, click Actions > Export Models. Select the models to export in the Export Model window, and click Export. The options to save the XML file depend on the browser.

You can export a packaged model to an XML file. You can export a custom model or multiple custom models to an XML file. However, you cannot export a combination of packaged and custom model to a single XML file.

Importing a Model

Use the Import Model wizard to import models into a Metadata Manager repository. When you import a model, you select the XML file that contains models and select the models you want to import.

The Import Model wizard analyzes and validates the XML file you select. If the model does not exist in the Metadata Manager repository, Metadata Manager imports the entire model. If the model exists, Metadata Manager imports the new and changed classes and relationships. If the model in the XML file matches the model in the Metadata Manager repository, Metadata Manager does not import the model.

In this example, the AccessIndex class does not exist in the Metadata Manager repository and the attributes for the AccessSchema class have changed. Metadata Manager creates the AccessIndex class and updates the AccessSchema class. Metadata Manager also lists the classes and relationships in the model that are unaffected by the model import process.

To import a model:
1. On the Model tab, click Actions > Import Models. The Import Model window appears.
2. Click Browse and select the XML file that contains the model you want to import.
3. Click Next.
4. Select the models in the XML file that you want to import.
5. Click Next.
6. The wizard analyzes the XML file and validates it against the models in the Metadata Manager repository. The wizard lists the classes and relationships it will create and update, and the classes and relationships that are not affected by the import process.
7. Click Import to create or update classes and relationships.
8. Optionally, if there are no classes or relationships to create or update, click Back to select another model or click Cancel to stop the Import Model wizard.
Step 2. Migrate Custom Metadata

To migrate custom metadata, you export the custom metadata from the source Metadata Manager repository to an XML file. You then import the XML file into the target Metadata Manager repository.

Export and import custom metadata between Metadata Manager repositories with the same version. You cannot import custom metadata from a different version.

You can export and import an XML file that contains resources and metadata objects. When you export metadata objects for a custom resource, the XML file contains all metadata objects for the resource. When you export a packaged resource type, the XML file contains the custom attributes you added for the resource.

When you import metadata for a custom resource, Metadata Manager imports all metadata objects for the resource in the XML file. When you import metadata for a packaged resource, Metadata Manager imports only the custom attributes from the XML file.

Exporting Metadata

Export metadata to an XML file in the Catalog view on the Browse tab.

To export metadata:
1. On the Browse tab, select the resource for which you want to export metadata.
2. Click Actions > Export Metadata > XML.

   The options to save or open the XML file depend on the browser.

Importing Metadata

Import metadata from an XML file in the Catalog view on the Browse tab.

Before you import metadata for a resource, complete the following tasks:

- **Verify that the model exists.** Verify that the model for the resource exists in the Metadata Manager repository. If the model does not exist, import it into the Metadata Manager repository.
- **Verify that the resource exists.** Verify that the resource exists in the metadata catalog. If the custom resource does not exist in the metadata catalog, create the resource. If the packaged resource does not exist in the metadata catalog, create the resource and load the metadata for the resource.

To import metadata:
1. On the Browse tab, select the resource for which you want to import metadata.
2. Click Actions > Import Metadata > XML.

   The Import Catalog Metadata window appears.
3. Click Browse and select the XML file that contains the metadata you want to import.
4. Click Import.

Rules and Guidelines for Importing Custom Metadata

Review the following rules and guidelines before you import custom metadata:

- **If you export comments and links, verify that the users who added the comments and links are registered in the target Metadata Manager instance.** Metadata Manager does not import a comment or link if the user is not registered in the target instance.
If you export values for custom attributes from a source resource model, verify that the custom attributes are defined in the target resource model. Metadata Manager does not import the values for custom attributes if the custom attributes are not defined in the target resource model.

Metadata Manager writes an entry in the log when it does not import a custom object or custom attribute value.

Step 3. Migrate Custom Resource Templates

Create a custom resource template in the Custom Metadata Configurator before you load custom metadata into the metadata catalog. When you migrate a custom resource template, you export the template from the source Metadata Manager repository and import it into the target Metadata Manager repository. Use the Custom Metadata Configurator to export and import custom resource templates.

When you export a custom resource template, the Custom Metadata Configurator exports it to a binary custom template file with a .ctf extension. Use the exported custom template file to import the custom resource template into the target repository. You can export or import one template at a time.

To troubleshoot errors and get more information about the export and import process, view customwizard.log in the following location:

```
<PowerCenter Client installation directory>\client\custom-configurator
```

Exporting a Custom Resource Template

To export a custom resource template:

1. In the Custom Metadata Configurator, log in to the Metadata Manager repository that contains the custom resource template you want to export.
2. Click Export.
   The Export Custom Template dialog box appears.
3. Select the custom resource for the template you want to export.
4. Enter the path and name of the custom template file that you want to export to or click Browse to select the file.
5. Click Export.
   The Custom Metadata Configurator exports the custom resource template to the file. Import the custom resource template into the target repository from the file.

Importing a Custom Resource Template

When you import a custom resource template, Metadata Manager creates a template in the Metadata Manager repository for a custom resource.

Before you import the custom resource template into the repository, you must import the model for the custom resource into the Metadata Manager instance. After you import the template, you must generate the PowerCenter objects that Metadata Manager requires to load the custom metadata.

If the template contains relationships to other resources, you must also configure the resources for the relationships before you import the template. If the resources do not exist in Metadata Manager, you must load the resources before you import the template.
To import a custom resource template:

1. Log in to the Metadata Manager instance where you want to import the custom resource template.
2. Create a custom resource on the Load tab.
3. In the Custom Metadata Configurator, log in to the Metadata Manager repository where you created the custom resource.
4. Click Import.
   
The Import Custom Template dialog box appears.
5. Select the name of the custom resource for which you want to import the template.
6. Enter the path and name of the custom template file you want to import or click the Browse button to select the custom template file.
7. If the custom resource template contains relationships to other resources, configure a target resource for the relationship for each source resource in the custom template file.
8. Click Import.
   
The Custom Metadata Configurator imports the template from the custom template file.

**Rules and Guidelines for Migrating Custom Resource Templates**

Use the following rules and guidelines when you migrate custom resource templates:

- You can migrate custom resource templates between instances of the same version of Metadata Manager only.
- You must create a custom resource before you can import a custom resource template.
- When you import a custom resource template and configure the source and target resources for relationships, the names of the source and target resources do not need to match.
- You can import custom resource templates for custom resources in Metadata Manager only if the custom resource does not have a template configured.
- If the template you want to import exists in the Metadata Manager repository, the import process fails.
- The models for the source and target resources and the model names must be the same.

**Step 4. Configure and Load the Custom Resources**

If you load metadata for a custom resource using a metadata source file, you must generate the PowerCenter objects, configure the metadata source file, and then load the custom resource.

To configure and load a custom resource, complete the following tasks:

1. **Generate the PowerCenter objects.** In the Custom Metadata Configurator, generate the PowerCenter objects, including mappings, sessions, and workflows, that extract the metadata from the metadata source files and load it into the Metadata Manager warehouse.
2. **Configure the metadata source files for the custom resource.** On the Load tab in Metadata Manager, configure the custom resource.
3. **Load the custom resource.** On the Load tab in Metadata Manager, load the custom resource.

**Related Topics:**

- “Step 8. Generate the PowerCenter Objects” on page 27
# INDEX

**A**
- AccessDB
  - AccessSchema class 7
  - AccessTable class 7
  - AccessTableColumn class 7
  - AccessView class 7
  - AccessViewColumn class 7
  - model example 7
- AccessSchema
  - class 7
- AccessTable
  - class 7
- AccessTableColumn
  - class 7
- AccessView
  - class 7
- AccessViewColumn
  - class 7
- association metadata files
  - adding class rules 25
  - creating 18
  - delimiters 22
  - guidelines 18
- Associations by Package report
  - running 5
- Associations Map tab
  - configuring relationships 24
- Attributes Map tab
  - configuring properties 22

**B**
- Browse tab
  - creating custom metadata objects 29
  - deleting metadata objects 30
  - deleting resources 30
  - editing properties 30
  - exporting and importing metadata 35
- business name
  - description 5
  - editing 30

**C**
- class rules
  - adding to association metadata files 25
  - adding to element metadata files 25
- classes
  - changing the lineage property 12
  - Class Exploration report 4
  - configuring 11
  - configuring groups 11

  - copying 12
  - creating 11
  - creating properties 13
  - creating relationships 16
  - deleting 12
  - deleting properties 14
  - description 4
  - editing 12
  - editing properties 14
  - editing relationships 16
  - moving 12
  - organizing properties 14
  - removing relationships 16

- custom metadata
  - adding 2
  - adding metadata 2
  - creating custom metadata objects 29
  - deleting custom metadata objects 30
  - deleting objects 30
  - editing properties 30
  - integration process overview 2
  - loading 2, 37
  - loading metadata 2
  - migrating 35
  - overview 1, 28

  - Custom Metadata Configurator
    - configuring class rules 25
    - configuring file delimiters 22
    - creating custom resource templates 19
    - creating rules 25
    - custom resource templates 19
    - deleting custom resource templates 21
    - editing custom resource templates 21
    - logging in 19
    - mapping properties 22
    - mapping relationships 24
    - overview 17
    - viewing custom resource templates 21

- custom metadata objects
  - creating 29
  - deleting 30
  - editing properties 30

- custom metadata source
  - adding and loading metadata 2

- custom model
  - overview 9

- custom properties
  - editing 32
  - export and import overview 31
  - exporting 31
  - importing 32

- custom resource templates
  - creating 19
  - Custom Metadata Configurator 19
  - deleting 21
  - editing 21
creating 16
creating parent-child 15
description 5
inge 16
mapping 24
object-level 5, 15
parent-child 5
removing 16
resources
  creating in metadata catalog 29
deleting custom resources 30
exporting 35

importing 35
loading 37
rules
  creating in Custom Metadata Configurator 25

Template Summary tab
  generating workflows 27