Informatica® PowerCenter
10.2 HotFix 1

Mapping Analyst for Excel Guide

Informatica PowerCenter Mapping Analyst for Excel Guide
10.2 HotFix 1
August 2018

Publication Date: 2018-09-25

Table of Contents

Preface

The *PowerCenter® Mapping Analyst for Excel Guide* is written for business analysts who want to use Microsoft Excel to develop PowerCenter mappings. It is also written for PowerCenter developers who want to use the Microsoft Excel files to begin developing PowerCenter mappings. This book assumes you have a working knowledge of Microsoft Excel.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit https://network.informatica.com.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit https://kb.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access Informatica Velocity resources at http://velocity.informatica.com.

If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through Online Support on Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
http://www.informatica.com/us/services-and-training/support-services/global-support-centers.

If you are an Informatica Network member, you can use Online Support at http://network.informatica.com.

CHAPTER 1

Understanding Mapping Analyst for Excel

This chapter includes the following topics:

Understanding Mapping Analyst for Excel Overview

Use Mapping Analyst for Excel to configure PowerCenter mappings in Microsoft Office Excel, and to export PowerCenter mappings to Microsoft Office Excel. The following types of users can collaborate when creating PowerCenter mappings:

- Business analyst. Creates a mapping specification in Microsoft Excel to define a mapping that can include sources, targets, and transformations. A business analyst is familiar with project requirements and source and target data, but is not a PowerCenter user.
- PowerCenter developer. Imports the mapping specification with the Repository Manager to create the corresponding PowerCenter objects. A PowerCenter developer can edit objects, implement additional functionality, and run a workflow generated from the mapping.

For example, a business analyst wants to merge employee data from multiple sources into a single data warehouse. However, the business analyst is not familiar enough with the PowerCenter Client to create the required mapping. The business analyst uses the Standard mapping specification template included with the PowerCenter Client to create a mapping specification that describes the required sources, transformations, and targets. A PowerCenter developer imports the mapping specification into a PowerCenter repository and develops the mapping in the PowerCenter Client.

Mapping Specifications

A mapping specification is a Microsoft Excel file that includes metadata to import into the PowerCenter repository. Use a mapping specification to define source or target definitions or to define a mapping by defining the source and target definitions, and the transformations between sources and targets.

When you create a mapping specification, you do not have to define all possible objects. The PowerCenter Repository Service imports the object definitions that you define in the mapping specification. For example, if you do not define transformations, the PowerCenter Repository Service imports source and target definitions only.

Mapping Specification Template

Mapping Analyst for Excel includes the Standard mapping specification template that you can use to develop a mapping specification in Microsoft Excel. The Standard mapping specification template is a Microsoft Excel file installed with the PowerCenter Client that defines the structure for a mapping specification.

The Standard mapping specification template contains the following Excel worksheets:

- Models. Use to configure all source and target definitions.
- Packages. Use to configure the hierarchy of packages in each model.
- Domains. Use to configure reference data within a mapping specification.
- Enumerations. Use to configure a list of reference values for each domain.
- Mappings. Use to configure a mapping name, source and target port connections, and aggregate and non-aggregate expressions. You can configure multiple mappings on one Mappings worksheet.
- Joins. Use to join source data from two related heterogeneous sources residing in different locations or file systems.
- Lookups. Use to configure a lookup to find data outside the mapping pipeline.
- Filters. Use to configure a filter to remove source data from the mapping pipeline.
- Rules. Use to configure reusable rules that you can use as expressions on the Mappings worksheet.

Note: The Standard mapping specification template also includes a Relationships worksheet that PowerCenter does not use. As a result, the Relationships worksheet is not supported.

The Standard mapping specification template, Standard-Blank.xlsx, is in the following directory:

```
<PowerCenterClientInstallationDir>\client\bin\mimb\conf\MIRModelBridgeTemplate
\MIRMicrosoftExcel
```

Mapping Analyst for Excel Process

To work with Mapping Analyst for Excel, use the following process:

1. A business analyst creates a mapping specification based on the Standard mapping specification template.
2. The business analyst enters the source and target metadata on the Models worksheet.
3. The business analyst uses Microsoft Excel to connect the source and target ports and develop expressions, joins, lookups, filters, and rules.
4. A PowerCenter developer uses the Repository Manager to import the mapping specification. The PowerCenter Repository Service creates the PowerCenter objects.
5. The PowerCenter developer completes the PowerCenter mapping in the PowerCenter Designer.

A PowerCenter developer can also use the Repository Manager to export PowerCenter metadata to a mapping specification.

You can use Mapping Analyst for Excel to import or export metadata. However, Mapping Analyst for Excel does not support a combination of exporting, editing, and importing in a single development process. For example, you export a PowerCenter mapping to Microsoft Excel, edit the metadata in Microsoft Excel, and then import the mapping specification back to PowerCenter. The imported mapping might contain inconsistent metadata.

CHAPTER 2

Standard Mapping Specification Template

This chapter includes the following topics:

Standard Mapping Specification Template Overview

Use the Standard mapping specification template to create a mapping specification for a single mapping or multiple mappings. The Standard mapping specification template contains mappings configured on multiple Excel worksheets. Mapping specifications based on this template can contain sources, targets, and Joiner, Filter, Expression, Lookup, Aggregator, and Java transformations. Use the Repository Manager to import the mapping specification and create the corresponding PowerCenter objects.

To create a mapping specification based on the Standard mapping specification template, complete the following steps:

1. Copy and rename the template.
2. Configure sources and targets on the Models worksheet.
3. Optionally, configure packages.
4. Optionally, configure domains and enumerations.
5. Optionally, configure rules.
6. Configure mappings.
7. Configure transformations.
8. Validate the mapping specification.

Before you create a mapping specification, install the Excel add-in included with Mapping Analyst for Excel. For more information about the Excel Add-in, see Excel Add-In on page 13.

Excel Add-in

Mapping Analyst for Excel includes an Excel add-in that adds a Metadata menu to Microsoft Excel. Use the Metadata menu to configure mapping specifications. You can install the add-in for Microsoft Excel 2016.

Installing the Add-in for Excel 2016

Install the Excel add-in to use the Metadata menu in Microsoft Excel 2016.

1. In Microsoft Excel, click **File** > **Options**.

   The **Excel Options** dialog box appears.
2. Click **Add-Ins**.
3. Select **Excel Add-Ins** from the **Manage** list, and click **Go**.

   The **Add-Ins** dialog box appears.
4. Click **Browse**.
5. Navigate to the Standard-Addin.xlam file located in the following directory:

   ```
   <PowerCenterClientInstallationDir>\client\bin\mimb\conf\MIRModelBridgeTemplate
   \MIRMicrosoftExcel
   ```
6. Click **OK**.
7. In the **Add-Ins** dialog box, verify that Meta Integration Standard Add-in is selected, and click **OK**.

   Microsoft Excel displays the Metadata menu that you can use to configure a mapping specification.

Copying and Renaming the Standard Mapping Specification Template

To create a mapping specification based on the Standard mapping specification template, copy and rename the template. You can find the Standard mapping specification template, Standard-Blank.xlsx, in the following directory:

```
<PowerCenterClientInstallationDir>\client\bin\mimb\conf\MIRModelBridgeTemplate
\MIRMicrosoftExcel
```

Viewing Columns on a Worksheet

When configuring a mapping specification worksheet, you can view the required columns, all used columns, all columns, or all extra columns where you can enter user-defined properties.

1. Click **Metadata** > **Show and Hide**, and then select one of the following options:
   - Min. Displays the required columns on the worksheet.
   - Used. Displays all used columns on the worksheet.
   - Max. Displays all columns on the worksheet.

- Extra. Displays all columns on the worksheet in addition to extra columns where you can enter user-defined properties.

2. To change the view, select another option from the Metadata menu.

User-defined Properties

User-defined properties enable you to add metadata that the Standard mapping specification template does not define. When you import a mapping specification that contains user-defined properties, the PowerCenter Repository Service creates the property as a metadata extension.

You can enter user-defined properties on the following worksheets:

- Models
- Packages
- Domains
- Enumerations
- Mappings

Configuring User-defined Properties

You can define user-defined properties for components configured in a mapping specification.

1. Click **Metadata** > **Show and Hide** > **Extra** to display columns for user-defined properties.
2. Enter a name for the user-defined property in the column header.
3. Optionally, add the datatype for the property inside parentheses after the name. Valid values are Text, Num, Date, or Bool. Use the following format:

   ```
   <property_name> (<data_type>)
   ```

   For example:

   ```
   Date created (Num)
   ```

   If you do not enter a value, Mapping Analyst for Excel uses Text as the default datatype.
4. Enter a value for the user-defined property in the appropriate row.

   When you import the mapping specification, the PowerCenter Repository Service adds the property as a metadata extension for all components that have a value for the property.

CHAPTER 3

Models Worksheet

This chapter includes the following topics:

Models Worksheet Overview

Use the Models worksheet to configure source and target definitions. You can configure multiple source and target definitions on a single Models worksheet.

Configuring the Models Worksheet

To configure the Models worksheet, enter the source and target metadata in one of the following ways:

- A business analyst types the source and target metadata into the Models worksheet.
- If the source and target metadata exists in the PowerCenter repository, a PowerCenter developer adds the source and target definitions to a mapping, but does not connect the ports. The developer then exports the mapping to a mapping specification. The business analyst opens the mapping specification in Microsoft Excel and views the Models worksheet populated with the source and target metadata. The business analyst does not edit the source and target metadata on the Models worksheet.

To type source and target metadata into the Models worksheet:

1. Click **Metadata** > **Show and Hide** > **Min** to view the required properties.
2. Enter the name of the model containing the source or target metadata.
3. Select the database type for the model.
4. Enter the name of the source or target schema.
5. Enter the name of the source or target table or file.
6. Enter the name of each column in the table or file.
7. Select the datatype for each column from the list of datatypes valid for the database type.

To enter a user-defined domain configured on the Domains worksheet, enter the name of the domain in the Datatype Name column.

8. Enter the length and scale of each column.

   If you selected a user-defined domain for the column datatype, the PowerCenter Repository Service assigns the datatype, length, and scale defined on the Domains worksheet to the column.

9. To enter optional properties, click **Metadata** > **Show and Hide** > **Max**.

Related Topics:

- "Models Worksheet Properties" on page 17
- "Domains and Enumerations Worksheets Overview" on page 23

Creating Multiple Models Worksheets

You can create multiple Models worksheets. For example, you might want to create one Models worksheet named "Sources" to define all source definitions and another Models worksheet named "Targets" to define all target definitions.

1. Click **Metadata** > **Insert Worksheet** > **Models**.

   The mapping specification adds another Models worksheet.

2. Rename the worksheet.

   Note: The PowerCenter Repository Service does not import the name of the worksheet.

Validating the Models Worksheet

The Standard mapping specification template includes macros to perform validation of the Models worksheet.

The Models worksheet performs the following validation:

- Names are defined for all tables and columns.
- Model type is one of the types supported by Mapping Analyst for Excel.
- Datatypes are valid for the specified model type.

1. Save the mapping specification.
2. On the Models worksheet, click **Metadata** > **Update and Check** > **Validate**.

   The validation displays a dialog box listing the number of errors.

3. Click **OK**.

   The mapping specification displays the first worksheet that contains errors. Cells that contain an error are colored red.

4. Move the pointer over a red cell to display the error message for the cell.
5. Correct errors and validate again.

Models Worksheet Properties

The Models worksheet includes the following sections:

- Model/Catalog. Source or target model name and database or file system type.
- Package. Hierarchy of packages within the model.
- Schema. Source or target schema name.
- Entity/Table/Record/View. Source or target table and file information.
- Attribute/Column/Field. Source or target column and field details.
- Datatype. Datatype for each column or field.

Model/Catalog Section

The Model/Catalog section defines the source or target model name and database or file system type.

The following table describes the properties that you can configure in the Model/Catalog section:

Property	Required/ Optional	Description
Name	Required	Name of the source or target model. Source and target models must have different names. The PowerCenter Repository Service does not import this value.
Type	Required	Source or target database or file system type. The PowerCenter Repository Service assigns this database type to the source or target definition. Informatica does not support Access or MySQL database types.

Package Section

The Package section defines the hierarchy of packages within the model.

Enter a name for each level in the package hierarchy. The Package section is optional. The PowerCenter Repository Service does not import values in the Package section.

Schema Section

The Schema section defines the source or target schema name.

The following table describes the property that you can configure in the Schema section:

Property	Required/ Optional	Description
Name	Required	Name of the source or target schema. During import, the PowerCenter Repository Service assigns the schema name to the database name for the source definition. The PowerCenter Repository Service does not import this value for target definitions.

Entity/Table/Record/View Section

The Entity/Table/Record/View section defines source or target table and file information.

The following table describes the properties that you can configure for each source and target:

Column Name	Required/ Optional	Description
Name	Required	Table or file name. The PowerCenter Repository Service creates a PowerCenter source or target definition with this name.
Logical Only	Optional	Indicates whether the table or file is logical only. Select true to set to logical only. A blank value is the same as selecting false. The PowerCenter Repository Service does not import this value. Default is blank.
View (SQL)	Optional	SQL statement definition for a view. The PowerCenter Repository Service does not import this value.
Business Name	Optional	Business name for the table or file. The PowerCenter Repository Service assigns this value to the business name of the source or target definition. If you do not enter a value, the PowerCenter Repository Service uses the name for the business name.
Description	Optional	Logical description of the table or file. If the Comment column contains a value, the PowerCenter Repository Service assigns the Description value to a metadata extension named Description for the source or target definition. If the Comment column is blank, the PowerCenter Repository Service assigns the Description value to the description of the source or target definition.
Comment	Optional	Physical description of the table or file. The PowerCenter Repository Service assigns this value to the description of the source or target definition.
Dimensional Role	Optional	Identifies a table as a dimension, a fact, or an outrigger. The PowerCenter Repository Service assigns this value to a metadata extension named DimensionalRoleType for the source or target definition.
Dimensional Type	Optional	Identifies a dimension or an outrigger table as fixed or evolving based on whether the number of rows remain constant or change. The PowerCenter Repository Service assigns this value to a metadata extension named DimensionalType for the source or target definition.

Attribute/Column/Field Section

The Attribute/Column/Field section defines source or target column and field details.

The following table describes the properties that you can configure for each source and target column:

Column Name	Required/ Optional	Description
Name	Required	Column name. The PowerCenter Repository Service creates a port in the source or target definition with this name.
Business Name	Optional	Business name for the column. The PowerCenter Repository Service assigns this value to the port business name. If you do not enter a value, the PowerCenter Repository Service uses the name for the business name.
Description	Optional	Logical description for the column. If the Comment column is blank, the PowerCenter Repository Service assigns the Description value to the port description.
Comment	Optional	Physical description of the column. The PowerCenter Repository Service assigns this value to the port description.
Position	Optional	Integer indicating the order of the field or column in the table or file. For example, for the first column of a table, enter 1. If you do not enter a value, the PowerCenter Repository Service uses the order of the rows in the worksheet.
Primary Key	Optional	Indicates whether the column is a primary key of the table. If you enter text, the PowerCenter Repository Service assigns this column as the primary key. Default is blank.
Alternate Key	Optional	Indicates whether the column is an alternate key of the table. The PowerCenter Repository Service does not import this value.
Inversion Key	Optional	Indicates whether the column is an inversion key of the table. The PowerCenter Repository Service does not import this value.

Index	Optional	Indicates whether this column is part of an index for the target table. Use the following format:
		`<IndexName> <n> <Asc/Dsc>`
		Use the n parameter to indicate the order of the column in the index. The Asc/Dsc parameter is optional. Enter Asc for ascending sort order or Dsc for descending sort order. For example:
		`IDX 1 Asc`
		The PowerCenter Repository Service imports the index information for target definitions.

Is Optional	Optional	Indicates whether you want to allow null data in the column or field. Enter No if the column cannot be null. If set to No, the PowerCenter Repository Service does not allow this column to contain null data. Default is Yes.

Column Name	Required/ Optional	Description
Default Value	Optional	Default value of the column or field. The PowerCenter Repository Service does not import this value.
Logical Only	Optional	Indicates whether the column is logical only. Select true to set to logical only. A blank value is the same as selecting false. The PowerCenter Repository Service does not import this value. Default is blank.

Datatype Section

The Datatype section defines the datatype for each column or field.

The following table describes the properties that you can configure for each source and target column:

Column Name	Required/ Optional	Description
Name	Optional	Name of a domain defined on the Domains worksheet. If you enter a name, the PowerCenter Repository Service assigns the datatype, length, and scale defined on the Domains worksheet to the column.
Business Name	Optional	Business name of the datatype. The PowerCenter Repository Service does not import this value.
Datatype	Required	Datatype of the column. The datatype must be valid for the database type. The PowerCenter Repository Service assigns this value to the datatype of the port.
Native Name	Optional	Native system datatype name for the column. For example, char (10). The PowerCenter Repository Service does not import this value.
Native Business Name	Optional	Logical native system datatype name for the column. For example, string. The PowerCenter Repository Service does not import this value.
Length	Required	Length of the column. The PowerCenter Repository Service assigns this value to the precision of the port.
Scale	Required	Scale of the column. The PowerCenter Repository Service assigns this value to the scale of the port.

Related Topics:

- "Configuring the Domains Worksheet" on page 24

CHAPTER 4

Packages Worksheet

This chapter includes the following topics:

Packages Worksheet Overview

Use the Packages worksheet to configure the hierarchy of packages within each model.

The Packages worksheet is optional. The PowerCenter Repository Service does not import information from the Packages worksheet. If you do not configure packages, you can delete the worksheet type from the mapping specification.

When you export a mapping to Microsoft Excel, the Packages worksheet includes the PowerCenter repository name and repository folder name as package levels.

Configuring the Packages Worksheet

Use the Packages worksheet to configure the hierarchy of packages within each model. Before you configure the Packages worksheet, configure packages in the Packages section on the Models worksheet.

1. Click **Metadata** > **Show and Hide** > **Max** to view all properties.
2. Select the name of a model defined on the Models worksheet.

   After selecting the model name, the mapping specification adds lists to the remaining columns in the Package section with valid package values from the Models worksheet.
3. In the Package section, select a name for each level of the hierarchy.
4. Select the schema name from the list of schemas defined for the selected model.
5. Optionally, enter a business name, description, and comment for the package.

Related Topics:

- "Package Section" on page 17
- "Packages Worksheet Properties" on page 22

Validating the Packages Worksheet

The Standard mapping specification template includes macros to perform validation of the Packages worksheet.

The Packages worksheet performs the following validation:

- Model, package, and schema names are defined on the Models worksheet.

1. Save the mapping specification.
2. On the Packages worksheet, click **Metadata** > **Update and Check** > **Validate**.

   The validation displays a dialog box listing the number of errors.
3. Click **OK**.

   The mapping specification displays the first worksheet that contains errors. Cells that contain an error are colored red.
4. Move the pointer over a red cell to display the error message for the cell.
5. Correct errors and validate again.

Packages Worksheet Properties

The Packages worksheet defines the hierarchy of packages within each model.

The following table describes the properties that you can configure on the Packages worksheet:

Property	Required/ Optional	Description
Model Name	Optional	Name of the model containing the package. Select from the list of models defined on the Models worksheet.
Package Level n Name	Optional	Name of the package level. You can configure a maximum of eight levels. Select from the list of package levels defined for the model on the Models worksheet.
Schema Name	Optional	Name of the schema containing the package. Select from the list of schemas defined on the Models worksheet.
Package Information Business Name	Optional	Business name of the package level.
Package Information Description	Optional	Description for the package level.
Package Information Comment	Optional	Comment for the package level.

CHAPTER 5

Domains and Enumerations Worksheets

This chapter includes the following topics:

Domains and Enumerations Worksheets Overview

You can optionally configure domains and enumerations to define reference data within a mapping specification. A domain is a reference table. An enumeration includes the reference table values for a domain. For example, you can create a domain named MovieType and define the possible enumerations as Action, Comedy, or Mystery.

When you configure the datatype for a column on the Models worksheet, you can enter a domain name. The PowerCenter Repository Service assigns the datatype, length, and scale defined for the domain on the Domains worksheet to the column.

Or, you can use domains and enumerations when you configure rules on the Rules worksheet. For example, to create a rule that checks for valid values of a medical procedure code, create a domain named ProcedureCode. Define all possible values for ProcedureCode on the Enumerations worksheet. When you create the rule and reference the ProcedureCode domain, the mapping specification includes a list of all possible values defined on the Enumerations worksheet.

The Domains and Enumerations worksheets are optional. You use domains and enumerations on other worksheets in the mapping specification. The PowerCenter Repository Service does not import information from the Domains and Enumerations worksheets. If you do not configure domains and enumerations, you can delete the worksheet types from the mapping specification.

Related Topics:

- "Rule Example Using Domains and Enumerations" on page 42

Configuring the Domains Worksheet

Use the Domains worksheet to configure a reference table within a mapping specification.

1. Click **Metadata** > **Show and Hide** > **Min** to view the required properties.
2. Select the name of the model that the domain belongs to.
3. Enter the name of the domain.
4. Select the datatype for the domain from the list of valid datatypes for the selected model.
5. Enter the length and scale of the domain.
6. To enter optional properties, click **Metadata** > **Show and Hide** > **Max**.

Related Topics:

- "Domains Worksheet Properties" on page 25

Configuring the Enumerations Worksheet

After you configure a domain, configure the reference values for the domain on the Enumerations worksheet.

1. Click **Metadata** > **Show and Hide** > **Max** to view all properties.
2. Select the name of the model that the domain belongs to.
3. Select the name of a domain configured for the selected model.
4. For the Enumeration Name column, enter a value or code.

   For example, for an ApprovalCode domain, enter the values `a` and `r`.
5. For the Enumeration Business Name, enter a name for the enumeration value.

   For example, for an ApprovalCode domain, enter the business names `Approved` and `Rejected`.
6. Optionally, enter a description for the enumeration value.

Related Topics:

- "Enumerations Worksheet Properties" on page 26

Creating Multiple Domains or Enumerations Worksheets

You can create multiple Domains or Enumerations worksheets. For example, you might define a ProcedureCode domain that has 200 enumerations. You can create one Enumerations worksheet named

ProcedureCode that includes the enumerations for the ProcedureCode domain. You can use another Enumerations worksheet to include the enumerations for all other domains.

1. Click **Metadata** > **Insert Worksheet** > **Domains or Enumerations**.

   The mapping specification adds another Domains or Enumerations worksheet.

2. Rename the worksheet.

   Note: The PowerCenter Repository Service does not import the name of the worksheet.

Validating the Domains and Enumerations Worksheets

The Standard mapping specification template includes macros to perform validation of the Domains and Enumerations worksheets.

The Domains worksheet performs the following validation:

- Model names are defined on the Models worksheet.

The Enumerations worksheet performs the following validation:

- Datatypes are defined on the Domains worksheet.
- Enumeration values are the correct datatype for the domain.

1. Save the mapping specification.
2. On the Models worksheet, click **Metadata** > **Update and Check** > **Validate**.

   The validation displays a dialog box listing the number of errors.

3. Click **OK**.

   The mapping specification displays the first worksheet that contains errors. Cells that contain an error are colored red.

4. Move the pointer over a red cell to display the error message for the cell.
5. Correct errors and validate again.

Domains Worksheet Properties

The Domains worksheet defines reference tables within a mapping specification.

The following table describes the properties that you can configure on the Domains worksheet:

Property	Required/ Optional	Description
Model Name	Required	Name of the source or target model that the domain belongs to. Select from the list of models defined on the Models worksheet.
Domain Name	Required	Name of the domain.

Property	Required/ Optional	Description
Parent Name	Optional	Name of the parent domain.
Business Name	Optional	Business name of the domain.
Description	Optional	Logical description of the domain.
Comment	Optional	Physical description of the domain.
Datatype	Required	Datatype of the domain. The datatype must be valid for the model type.
Native Datatype	Optional	Native system datatype name for the domain. For example, char (10).
Length	Optional	Length of the domain.
Scale	Optional	Scale of the domain.
Initial Value	Optional	Initial value of the domain. The format of the value must be valid for the domain datatype.
Min Value	Optional	Minimum value for the valid range of values for the domain. The format of the value must be valid for the domain datatype.
Max Value	Optional	Maximum value for the valid range of values for the domain. The format of the value must be valid for the domain datatype.
Virtual	Optional	Determines whether the domain is reusable. If set to true, the domain is not reusable. You can associate the domain with one column on the Models worksheet. If set to false, the domain is reusable. You can associate the domain with multiple columns on the Models worksheet.

Enumerations Worksheet Properties

The Enumerations worksheet defines reference values for each domain.

The following table describes the properties that you can configure on the Enumerations worksheet:

Property	Required/ Optional	Description
Model Name	Required	Name of the source or target model that the domain belongs to. Select from the list of models defined on the Models worksheet.
Domain Name	Required	Name of the domain for which you want to provide enumeration values. Select from the list of domains defined for the selected model.
Enumeration Name	Required	Enumeration value or code. For example, for an ApprovalCode domain, enter `a` and `r`.

Property	Required/ Optional	Description
Enumeration Business Name	Required	Business name associated with an enumeration value or code. For example, for an ApprovalCode domain, enter `Approved` and `Rejected`. If you configure a rule on the Rules worksheet that references enumerations, the mapping specification displays the enumeration business name values.
Enumeration Description	Optional	Description of the enumeration value or code.

CHAPTER 6

Mappings Worksheet

This chapter includes the following topics:

Mappings Worksheet Overview

Use the Mappings worksheet to configure the mapping name, connect source and target ports, configure aggregate and non-aggregate expressions, and use rules defined on the Rules worksheet.

You can configure multiple mappings on one Mappings worksheet.

Configuring the Mappings Worksheet

Use the Mappings worksheet to connect source and target ports and to configure expressions.

Connect a single source port to a single target port by selecting the source and target columns in the same row.

Connect multiple source ports to a single target port by selecting the source columns in consecutive rows, and then by selecting the target column in the top row. You can use Microsoft Excel to merge empty cells. You can connect the columns, street, city, state, and zip in the address source to the address column in the Emp_tbl target.

1. Click **Metadata** > **Show and Hide** > **Min** to view the required properties.
2. Enter the mapping name in the Mapping Name column.

   The PowerCenter Repository Service assigns this name to the mapping.
3. Enter the source model name. The name must match the model name defined on the Models worksheet.

   After entering the model name, the mapping specification adds lists to the remaining columns in the Source section with valid values from the Models worksheet.
4. Select the Source Schema, Table, and Column names for each source port that you want to configure.

5. Click **Metadata** > **Update and Check** > **Annotate**.

   The mapping specification adds the business name, description, and datatypes for each column as defined on the Models worksheet.

6. Repeat steps 3 through 5 to configure each connecting target port.

   The PowerCenter Repository Service connects source and target columns that are on the same row in the worksheet.

7. Click **Metadata** > **Show and Hide** > **Max** to display the Mapping Specification Expression section.

8. Optionally, enter an aggregate expression, non-aggregate expression, or rule in the Mapping Specification Expressions column to transform the data.

Related Topics:

- "Mapping Specification Section" on page 30
- "Mappings Worksheet Properties" on page 30

Creating Multiple Mappings Worksheets

You can configure multiple mappings on one Mappings worksheet. Or, you can create additional Mappings worksheets and configure each mapping on a separate worksheet.

1. Click **Metadata** > **Insert Worksheet** > **Mappings**.

   The mapping specification adds another Mappings worksheet.

2. Rename the worksheet.

   Note: The PowerCenter Repository Service does not import the name of the worksheet.

Validating the Mappings Worksheet

The Standard mapping specification template includes macros to perform validation of the Mappings worksheet.

The Mappings worksheet performs the following validation:

- Models, schemas, tables, and columns are defined on the Models worksheet.
- Functions used in the mapping are supported by Mapping Analyst for Excel.
- Expressions contain columns that are defined on the Models worksheet.
- Rules are defined on the Rules worksheet.

1. Save the mapping specification.
2. On the Mappings worksheet, click **Metadata** > **Update and Check** > **Validate**.

   The validation displays a dialog box listing the number of errors.

3. Click **OK**.

   The mapping specification displays the first worksheet that contains errors. Cells that contain an error are colored red.

4. Move the pointer over a red cell to display the error message for the cell.
5. Correct errors and validate again.

Mappings Worksheet Properties

Use the following sections to configure properties on the Mappings worksheet:

- Mapping. Name and description of the mapping.
- Source and Target. Connections between source and target ports.
- Mapping Specification. Aggregate expressions, non-aggregate expressions, and rules.

Mapping Section

For the Mapping Name column, enter a name for the mapping. The PowerCenter Repository Service assigns this name to the mapping.

Optionally, enter a description for the mapping. The PowerCenter Repository Service does not import this value.

Source and Target Sections

The Source and Target sections contain the source and target data that you want to connect. The PowerCenter Repository Service connects the source and target columns configured on the same row.

The following table describes the properties that you can configure in the Source and Target sections:

Property	Required/ Optional	Description
Model	Required	Name of the source or target model. The name must match the model name defined on the Models worksheet. Enter the model property first. The mapping specification adds lists of valid values to the remaining columns.
Schema	Required	Name of the source or target schema. Select from the list of schemas defined for the model.
Table	Required	Name of the source or target table or file. Select from the list of tables defined for the schema.
Column	Required	Name of the source or target column. Select from the list of columns defined for the table or file.

Mapping Specification Section

Use the Mapping Specification section to configure the following components in the mapping:

- Aggregate expressions
- Non-aggregate expressions
- Rules

Configure an expression or rule to transform the data when the PowerCenter Integration Service moves the data from the source to the target. If you do not configure an expression or rule for a row, then the PowerCenter Integration Service passes the data unchanged from the source to the target.

Aggregate Expressions

You can configure an aggregate expression to perform calculations on multiple values in a port. You configure an aggregate expression in the Mapping Specification Expressions section on the Mappings worksheet. An aggregate expression defined in the mapping specification becomes an Aggregator transformation in a PowerCenter mapping. Use the PowerCenter transformation language to write the aggregate expression.

Use aggregate functions to perform calculations on multiple values in a port. For example, the following aggregate expression evaluates all values in the Quantity port of the SALES table and returns the highest value:

```
MAX(SALES.Quantity)
```

You can use an aggregate function, conditional clauses, and non-aggregate functions in an aggregate expression. You can also nest one aggregate function within another aggregate function, for example:

```
MAX( COUNT( SALES.Quantity))
```

Define an aggregate expression in the Expression column for the row where you want to write return values.

Optionally, add a description for the expression. The PowerCenter Repository Service assigns this value to the description of the expression in the Aggregator transformation.

The following table shows an aggregate expression configured for the StoreSales port of the T_Sales target table:

Source Schema	Source Table	Source Column	Expression	Target Column	Target Table	Target Schema
Sales	Sales	TransactionAmount	SUM(TransactionAmount)	StoreSales	T_Sales	DM

Grouping Values in Aggregate Expressions

In PowerCenter, you use group by ports in the Aggregator transformation to group values for aggregate calculations. The result of an aggregate expression varies based on the group by ports that you configure.

For example, when the PowerCenter Integration Service calculates the following aggregate expression with no group by ports defined, it finds the total sales from all transactions:

```
SUM( SALES.TransactionAmount )
```

However, if you use the same expression, and you group by the StoreID port, the PowerCenter Integration Service returns the total sales for each store ID.

When a PowerCenter developer imports a mapping specification, the PowerCenter Repository Service creates an Aggregator transformation for each aggregate expression defined for the target. Any port in the Aggregator transformation without an aggregate expression becomes a group by port.

Use the Mapping Specification Description column to indicate which group by ports you want to use. After importing the mapping specification, the PowerCenter developer can view that information in the Aggregator transformation and configure group by ports appropriately.

Non-aggregate Expressions

You can configure an expression to calculate values in a single row for each row in a port. You configure an expression in the Mapping Specification Expression section on the Mappings worksheet. An expression defined in the mapping specification becomes an Expression transformation in a PowerCenter mapping. Use the PowerCenter transformation language to write the non-aggregate expression.

Use an expression to calculate values in a single row for each row in a port. For example, the following expression increases the cost of each item by 5 %:

```
INVENTORY.Cost + (INVENTORY.Cost * .05)
```

You can use conditional clauses and non-aggregate functions in non-aggregate expressions. You can also nest non-aggregate functions.

Define an aggregate expression in the Expression column for the row where you want to write return values.

Optionally, add a description for the expression. The PowerCenter Repository Service assigns this value to the description of the expression in the Expression transformation.

Rules

You can enter rules defined on the Rules worksheet. Enter a rule name in the Mapping Specification Expression section of the Mappings worksheet using the following format:

```
%<rule_name>%
```

A rule defined in the mapping specification becomes a Java transformation in a PowerCenter mapping. After defining a rule on the Rules worksheet, you can reuse the rule multiple times on the Mappings worksheet.

Optionally, add a description for the rule. The PowerCenter Repository Service does not import this value for rules.

Related Topics:

- "Configuring the Rules Worksheet" on page 43

CHAPTER 7

Joins, Lookups, and Filters Worksheets

This chapter includes the following topics:

Joins, Lookups, and Filters Worksheets Overview

You can add the following transformations to a mapping specification:

- Joiner transformation. Configure Joiner transformations on the Joins worksheet.
- Lookup transformation. Configure Lookup transformations on the Lookups and Mappings worksheets.
- Filter transformation. Configure Filter transformations on the Filters worksheet.

You can configure multiple transformations for each supported transformation type.

A mapping specification does not define the order of transformations in a mapping. When you import a mapping specification, the PowerCenter Repository Service adds the transformations to the mapping in a specific order. After the source definition, the PowerCenter Repository Service adds the Joiner, Lookup, Filter, Expression, Java, and Aggregator transformations, and then the target definition. A PowerCenter developer can edit the mapping and change the order of the transformations.

Joiner Transformation

You can join source data from two related heterogeneous sources residing in different locations or file systems. You join source data by matching one or more pairs of columns between the two sources. You configure a join on the Joins worksheet. A join in the mapping specification becomes a Joiner transformation in a PowerCenter mapping.

Configuring the Joins Worksheet

Use the Joins worksheet to join source data from two related heterogeneous sources.

1. Click **Metadata** > **Show and Hide** > **Max** to view all properties.
2. Enter the mapping name in the Mapping Name column.

   The PowerCenter Repository Service creates a Joiner transformation in this mapping.
3. In the Left Model column, enter the source model name that owns the left or master table. The name must match the model name defined on the Models worksheet.

   After entering the model name, the mapping specification adds lists to the remaining columns in the Left section with valid values from the Models worksheet.
4. Select the Left Schema, Table, and Column names for the left column that you want to join.
5. Click **Metadata** > **Update and Check** > **Annotate**.

   The mapping specification adds the business name, description, and datatypes for each column as defined on the Models worksheet.
6. Select one of the following join types: Inner, Left, Right, Outer.
7. Repeat steps 3 through 5 to select the Right model, schema, table, and column to join with the left or master table.
8. Optionally, enter a description for the join.

Joins Worksheet Properties

The Joins worksheet defines PowerCenter Joiner transformations.

The PowerCenter Repository Service uses the following syntax to name the Joiner transformation:

```
JNR_<LeftTableName>_join_<RightTableName>
```

The following table describes the properties that you can configure on the Joins worksheet:

Property	Required/ Optional	Description
Mapping Name	Required	Name of the mapping for the join. The PowerCenter Repository Service creates the Joiner transformation in this mapping.
Left Model	Required	Name of the source model that owns the left or master table. The name must match the model name defined on the Models worksheet. Enter the model property first. The mapping specification adds lists of valid values to the remaining columns.
Left Schema	Required	Name of the schema that owns the left or master table. Select from the list of schemas defined for the model.
Left Table	Required	Name of the left table. Select from the list of tables defined for the schema.
Left Column	Required	Name of the left column that you want to join. Select from the list of columns defined for the table.

Property	Required/ Optional	Description
Join Type	Required	Type of join for the Joiner transformation. Enter one of the following types: - LEFT. Left outer join. The PowerCenter Repository Service creates a Joiner transformation with a Master Outer join. - RIGHT. Right outer join. The PowerCenter Repository Service creates a Joiner transformation with a Detail Outer join. - INNER. Inner join. The PowerCenter Repository Service creates a Joiner transformation with a Normal join. - OUTER. Outer join. The PowerCenter Repository Service creates a Joiner transformation with a Full Outer join.
Right Model	Required	Name of the source model that owns the right or detail table. The name must match the model name defined on the Models worksheet. Enter the model property first. The mapping specification adds lists of valid values to the remaining columns.
Right Schema	Required	Name of the schema that owns the right or detail table. Select from the list of schemas defined for the model.
Right Table	Required	Name of the right table. Select from the list of tables defined for the schema.
Right Column	Required	Name of the right column that you want to join. Select from the list of columns defined for the table.
Join Description	Optional	Description for the join. The PowerCenter Repository Service assigns this value to the description of the Joiner transformation.

Lookup Transformation

You can configure a lookup to find data outside the mapping pipeline. A lookup in the mapping specification becomes a connected Lookup transformation in a PowerCenter mapping.

You can perform a lookup on any table defined in the mapping specification.

To create a lookup, define the following information:

- Lookup condition on the Lookups worksheet. The PowerCenter Integration Service finds data in the lookup table with a lookup condition. The lookup condition is similar to the WHERE clause in an SQL query.
- Return values on the Mappings worksheet. When the lookup condition is met, the PowerCenter Integration Service returns values from the lookup table.

Lookup Example

You have the StoreSales source table, Revenue sourcelookup table, and T_Sales target table.

The following table shows the configured sources and targets on the Models worksheet:

Model Name	Model Type	Schema Name	Table Name	Column Name
SalesDB	Oracle	Sales	StoreSales	StoreID
-	-	-	-	TransactionDate
-	-	-	-	TransactionAmount
-	-	-	Revenue	StoreID
-	-	-	-	MonthlyRevenue
DMModel	Oracle	DM	T_Sales	StoreID
-	-	-	-	TransactionDate
-	-	-	-	TransactionAmount
-	-	-	-	MonthlyRevenue

For each StoreID column in the StoreSales table, you want to look up the StoreID column in the Revenue table and return the MonthlyRevenue value to the T_Sales target.

You define the lookup table and lookup condition on the Lookups worksheet. The lookup condition must fully qualify the column in the source table that you want to look up.

The following table shows how to define the lookup table and lookup condition on the Lookups worksheet:

Lookup Model	Lookup Schema	Lookup Table	Lookup Condition
SalesDB	Sales	Revenue	StoreID= SalesDB.Sales.StoreSales.StoreID

On the Mappings worksheet, define the value to return when the lookup condition is met. In the row that defines the MonthlyRevenue target column, select the lookup table column named MonthlyRevenue as the source to connect to the target.

The following table shows how to define the return value on a single row of the Mappings worksheet:

Source Model	Source Schema	Source Table	Source Column	Target Column	Target Table	Target Schema	Target Model
SalesDB	Sales	Revenue	MonthlyRevenue	MonthlyRevenue	T_Sales	DM	DMModel

Configuring the Lookup Condition

Use the Lookups worksheet to configure a lookup condition to find data outside of the mapping pipeline.

1. On the Lookups worksheet, click **Metadata** > **Show and Hide** > **Max** to view all properties.
2. Enter the mapping name in the Mapping Name column.

   The PowerCenter Repository Service creates a Lookup transformation in this mapping.

3. In the Lookup Model column, enter the model name that owns the lookup table. The name must match the model name defined on the Models worksheet.

   After you enter the model name, the mapping specification adds lists to the remaining columns in the Lookup section with valid values from the Models worksheet.
4. Select the Lookup Schema and Table.
5. Click **Metadata** > **Update and Check** > **Annotate**.

   The mapping specification adds the business name, description, and datatypes for each column as defined on the Models worksheet.
6. In the condition column, enter a lookup condition in the following format:

   ```
   <LookupTableColumn><operator><SourceModelName>.<SchemaName>.<TableName>.<ColumnName>
   ```

   You can use one of the following operators or combination of operators:

   ```
   =, <, <=, >, >=, !=
   ```
7. Optionally, enter a description for the lookup.

Configuring the Lookup Return Values

Use the Mappings worksheet to configure the values to return when the lookup condition is met.

1. On the Mappings worksheet, click **Metadata** > **Show and Hide** > **Min** to view the required properties.
2. Enter the target model name for the target where you want to return the lookup values. The name must match the model name defined on the Models worksheet.

   After you enter the model name, the mapping specification adds lists to the remaining columns in the Target section with valid values from the Models worksheet.
3. Select the Target Schema, Table, and Column names for the target column that you want to return the lookup values to.
4. Click **Metadata** > **Update and Check** > **Annotate**.

   The mapping specification adds the business name, description, and datatypes for each column as defined on the Models worksheet.
5. In the same row as the target column, select the Source Model, Schema, Table, and Column name for the lookup table column that you want to return to the target.
6. Click **Metadata** > **Show and Hide** > **Max** to display the Mapping Specification Expression section.
7. Optionally, enter a description for the row explaining that source column values are written to the target column when the lookup condition is met.

Lookups Worksheet Properties

The Lookups worksheet defines the lookup condition for Lookup transformations.

The following table describes the properties that you can configure on the Lookups worksheet:

Property	Required/ Optional	Description
Mapping Name	Required	Name of the mapping for this lookup. The PowerCenter Repository Service creates the Lookup transformation in this mapping.
Lookup Model	Required	Name of the model that owns the lookup table. The name must match the model name defined on the Models worksheet. Enter the model property first. The mapping specification adds lists of valid values to the remaining columns.
Lookup Schema	Required	Name of the schema that owns the lookup table. Select from the list of schemas defined for the model.
Lookup Table	Required	Lookup table name. Select from the list of tables defined for the schema. The PowerCenter Repository Service uses the lookup table name as the name of the Lookup transformation.

Lookup Condition	Required	Lookup condition that uses the following format:
		`<LookupTableColumn><operator><SourceModelName>.<SchemaName>.<TableName>.<ColumnName>`
		You can use one of the following operators or combination of operators:
		`=, <, <=, >, >=, !=`
		You must fully qualify the source column with the owning table, schema, and model name.
		For example, the source data contains a code column. The lookup table contains ID and name columns. You configure the following lookup condition:
		`ID = PersonnelDB.Employment.Company.code`
		For each code, the PowerCenter Integration Service returns the name column from the lookup table. Configure the return values on the Mappings worksheet.

Lookup Description	Optional	Description for the lookup. The PowerCenter Repository Service assigns this value to the description of the Lookup transformation.

Filter Transformation

You can configure a filter to remove source data from the mapping pipeline. You configure a filter on the Filters worksheet. A filter defined in the mapping specification becomes a Filter transformation in a PowerCenter mapping.

Configure a filter by defining a filter condition. The filter condition is an expression that returns TRUE or FALSE. The filter condition must contain the source model, schema, table, and column names in the following format:

```
<SourceModelName>.<SchemaName>.<TableName>.<ColumnName>
```

For example, use the following expression to filter out transactions with negative values, such as returns:

```
PurchasingDB.Sales.SALES.TransactionAmount > 0
```

You can also specify multiple components for a condition using the AND and OR logical operators.

Filters Worksheet Properties

The Filters worksheet defines Filter transformations.

The following table describes the properties that you can configure on the Filters worksheet:

Property	Required/ Optional	Description
Mapping Name	Required	Name of the mapping for this filter. The PowerCenter Repository Service creates the Filter transformation in this mapping.

Filters Condition	Required	Filter condition that returns TRUE or FALSE. The condition must contain the source model, schema, table, and column names in the following format:
		`<SourceModelName>.<SchemaName>.<TableName>.<ColumnName>`
		The names must match the names defined on the Models worksheet.
		For example, use the following expression to filter out transactions with negative values, such as returns:
		`PurchasingDB.Sales.SALES.TransactionAmount > 0`
		You can also specify multiple components for a condition using the AND and OR logical operators.

Filters Description	Optional	Description for the filter. The PowerCenter Repository Service assigns this value to the description of the Filter transformation.

Creating Multiple Joins, Filters, and Lookups Worksheets

You can define multiple Joiner, Filter, and Lookup transformations on a single worksheet. Or, you can create multiple Joins, Filters, or Lookups worksheets to define a single transformation on each worksheet type.

1. Click **Metadata** > **Insert Worksheet** > **Joins, Lookups, or Filters**.

   The mapping specification adds another Joins, Filters, or Lookups worksheet.

2. Rename the worksheet.

   Note: The PowerCenter Repository Service does not import the name of the worksheet.

Validating the Joins, Lookups, and Filters Worksheets

The Standard mapping specification template includes macros to perform validation of the Joins, Lookups, and Filters worksheets.

The Joins, Lookups, and Filters worksheets perform the following validation:

- The mapping name is defined on the Mappings worksheet.
- The model, schema, table, and column names match names defined on the Models worksheet.

1. Save the mapping specification.
2. On the Joins, Lookups, or Filters worksheet, click **Metadata** > **Update and Check** > **Validate**.

   The validation displays a dialog box listing the number of errors.
3. Click **OK**.

   The mapping specification displays the first worksheet that contains errors. Cells that contain an error are colored red.
4. Move the pointer over a red cell to display the error message for the cell.
5. Correct errors and validate again.

CHAPTER 8

Rules Worksheet

This chapter includes the following topics:

Rules Worksheet Overview

Use the Rules worksheet to define reusable rules that you can use as expressions on the Mappings worksheet. You can use rules in a mapping specification to perform simple data cleansing.

When you import a mapping specification that contains a rule defined on the Rules worksheet, the PowerCenter Repository Service creates a Java transformation for the rule. The mapping specification input parameters are the Java transformation input ports, the rule expression is the Java code, and the output results are the output ports.

Note: If you export a PowerCenter mapping that includes a Java transformation, the PowerCenter Repository Service does not export the Java transformation to the mapping specification.

When you configure a rule, you can optionally use domains and enumerations defined on the Domains and Enumerations worksheets. When you specify a value for a parameter on the Rules worksheet, you can select from the list of possible reference values that are defined on the Enumerations worksheet.

The Rules worksheet is optional. If you do not configure rules, you can delete the worksheet type from the mapping specification.

Rule Example

A rule is a conditional statement that takes input parameters, performs a calculation on the parameters, and produces output parameters.

For example, you create a rule named FullName. The rule uses fname and lname as input parameters, concatenates the parameters, and assigns the result to an output parameter named FullName.

The following table shows the required columns that you configure on the Rules worksheet:

Rule Name	Parameter1 Name	Parameter2 Name	Result1 Name	Expression
FullName	fname	lname	FullName	(fname + " " + lname)

On the Mappings worksheet, include the rule name as an expression to connect source columns first_name and last_name to the target column full_name. Enter the rule name using the following format:

```
%FullName%
```

When the PowerCenter Repository Service imports the mapping specification, it creates a Java transformation named FullName. The transformation uses fname and lname as input ports and uses FullName as an output port. The transformation contains the following Java code:

```
if(true) {
    FullName=fname + " " + lname;
}
```

Rule Example Using Domains and Enumerations

You can create a rule that refers to a reference table defined on the Domains worksheet and a list of reference values defined for the domain on the Enumerations worksheet.

For example, you create a domain named CreditStatus on the Domains worksheet.

The following table shows the possible values you define for the domain on the Enumerations worksheet:

Enumeration Name	Enumeration Business Name
1	Bad
2	Good
3	Excellent

You create a rule that checks a credit score and assigns a credit status. When you configure the rule, you select the CreditStatus domain as the datatype of the rule output parameter. In the Result column, you select an enumeration business name value defined for the CreditStatus domain.

The following table shows the required columns that you configure on the Rules worksheet:

Rule Name	Parameter1 Name	Parameter1 Value	Result1 Name	Datatype	Result
GetCreditStatus	credit	>710	creditStatus	CreditStatus	Excellent
-	-	in(650,709)	-	-	Good
-	-	*	-	-	Bad

On the Mappings worksheet, include the rule name as an expression to connect the source column credit to the target column creditStatus. Enter the rule name using the following format:

```
%GetCreditStatus%
```

When the PowerCenter Repository Service imports the mapping specification, it creates a Java transformation named GetCreditStatus. The transformation uses credit as an input port and uses creditStatus as an output port. The transformation contains the following Java code that uses the CreditStatus enumeration names (3, 2, and 1) instead of the business names (Excellent, Good, Bad):

```
if(credit>710) {
    creditStatus=3;
}
else
if(credit==650 || credit==709) {
    creditStatus=2;
}
else
if(true) {
    creditStatus=1;
}
```

Related Topics:

- "Domains and Enumerations Worksheets Overview" on page 23

Configuring the Rules Worksheet

Use the Rules worksheet to configure reusable rules that you can use as expressions on the Mappings worksheet.

1. Click **Metadata** > **Show and Hide** > **Max** to view all properties.
2. Enter the rule name in the Rule name column.
3. Select the source model name from the list of models defined on the Models worksheet.
4. Enter the name, datatype, and value of an input parameter.

   If you defined domains on the Domains worksheet, you can select a domain name for the datatype.
5. Repeat step 4 to configure additional input parameters for the rule.
6. Select the target model name from the list of models defined on the Models worksheet.
7. Enter the name, datatype, and expression of an output result.

   If you defined domains on the Domains worksheet, you can select a domain name for the datatype.
8. Repeat step 7 to configure additional output results for the rule.
9. Click **Metadata** > **Update and Check** > **Annotate**.

   The mapping specification adds the business name, description, and datatypes for each model defined on the Models worksheet and each domain defined on the Domains worksheet.

After you configure the rule, enter the rule name as an expression on the Mappings worksheet for the appropriate source and target row. Use the following format:

```
%<rule_name>%
```

Related Topics:

- "Rules Worksheet Properties" on page 45
- "Rules" on page 32

Formatting the Rules Worksheet

You can format the Rules worksheet to display a single rule at a time with the rule parameter names as column headers. You might want to format the Rules worksheet to make a rule more readable if you have configured multiple rules.

1. Select the single rule that you want to display.
2. Click **Metadata** > **Update and Check** > **Format**.

   The mapping specification displays the selected rule in a more readable format. For example, the default worksheet view includes a Parameter Name column and a Parameter Value column. After formatting, the worksheet lists the parameter name as a column header with the parameter values listed underneath.
3. To display all rules again, click **Metadata** > **Update and Check** > **Format**.

Creating Multiple Rules Worksheets

You can create multiple Rules worksheets. For example, if you configure a large number of rules, you can group the rules on separate worksheets.

1. Click **Metadata** > **Insert Worksheet** > **Rules**.

   The mapping specification adds another Rules worksheet.
2. Rename the worksheet.

   Note: The PowerCenter Repository Service does not import the name of the worksheet.

Validating the Rules Worksheet

The Standard mapping specification template includes macros to perform validation of the Rules worksheet.

The Rules worksheet performs the following validation:

- Rules contain references to source and target models defined on the Models worksheet.
- The datatype for a parameter is valid for the selected model type, or is a user-defined datatype configured on the Domains worksheet.
- Rules contain at least one input parameter and one output result.

1. Save the mapping specification.
2. On the Rules worksheet, click **Metadata** > **Update and Check** > **Validate**.

   The validation displays a dialog box listing the number of errors.
3. Click **OK**.

   The mapping specification displays the first worksheet that contains errors. Cells that contain an error are colored red.
4. Move the pointer over a red cell to display the error message for the cell.
5. Correct errors and validate again.

Rules Worksheet Properties

The Rules worksheet includes the following sections:

- Rule. Name and description of the rule.
- Source. Input parameters.
- Target. Output results.

Rule Section

The Rule section contains the rule name and description.

The following table describes the properties that you can configure in the Rule section:

Property	Required/ Optional	Description
Name	Required	Name of the rule. Each rule name must be unique and cannot be a function or a reserved word in the PowerCenter transformation language. To use the rule in a mapping, enter the rule name within percent characters in the Expression column of the Mappings worksheet. For example: `%<rule_name>%` The PowerCenter Repository Service assigns this name to the Java transformation.
Description	Optional	Description of the rule. The PowerCenter Repository Service does not import this value.

Source Section

The Source section contains the input parameters for the rule. You can configure a maximum of 10 input parameters.

The following table describes the properties that you can configure in the Source section:

Property	Required/ Optional	Description
Model	Required	Source model defined on the Models worksheet that includes the source column that you want to transform with the rule.
Parameter Name	Required	Name of the input parameter. The PowerCenter Repository Service assigns this value to a Java transformation input port.
Parameter Description	Optional	Description of the input parameter. The PowerCenter Repository Service does not import this value.

Property	Required/ Optional	Description
Parameter Datatype	Required	Datatype of the input parameter. Select one of the following options: - Datatype valid for the selected source model. The PowerCenter Repository Service assigns the value to the datatype of the Java transformation input port. - Domain name defined on the Domains worksheet. The PowerCenter Repository Service assigns the datatype, length, and scale defined on the Domains worksheet to the Java transformation input port.
Parameter Length	Required	Length of the input parameter. The PowerCenter Repository Service assigns this value to the precision of the Java transformation input port.
Parameter Scale	Required	Scale of the input parameter. The PowerCenter Repository Service assigns this value to the scale of the Java transformation input port.

Parameter Value	Optional	Value that the input parameter must have to meet the rule condition. If you selected a domain name for the datatype, the mapping specification displays a list of all possible domain values defined on the Enumerations worksheet.
		If blank, all values of the input parameter meet the rule condition.
		The value can contain any of the following:
		- Constant. Enter a number or a string value.
		- Simple expression. Enter a comparison operator such as > or < followed by the value to compare. For example:
		`>25`
		- Complex expression. Enter a complex expression within parentheses. For example:
		`(stateWord.toUpperCase().startsWith("CALI"))`
		The expression must include Java supported operators. The PowerCenter Repository Service uses this value to create an if statement in the Java code of the Java transformation.

Target Section

The Target section contains the output results for the rule. You can configure a maximum of 10 output results.

The following table describes the properties that you can configure in the Target section:

Property	Required/ Optional	Description
Model	Required	Target model defined on the Models worksheet that includes the target column that you want to write the rule result to.
Result Name	Required	Name of the output result. The PowerCenter Repository Service assigns this value to a Java transformation output port.
Result Description	Optional	Description of the output result. The PowerCenter Repository Service does not import this value.

Property	Required/ Optional	Description
Result Datatype	Required	Datatype of the output result. Select one of the following options: - Datatype valid for the selected target model. The PowerCenter Repository Service assigns the value to the datatype of the Java transformation output port. - Domain name defined on the Domains worksheet. The PowerCenter Repository Service assigns the datatype, length, and scale defined on the Domains worksheet to the Java transformation output port.
Result Length	Required	Length of the output result. The PowerCenter Repository Service assigns this value to the precision of the Java transformation output port.
Result Scale	Required	Scale of the output result. The PowerCenter Repository Service assigns this value to the scale of the Java transformation output port.
Result	Required	Value to assign to the output result. If you selected a domain name for the datatype, the mapping specification displays a list of all possible domain values defined on the Enumerations worksheet. The value can contain any of the following: - Constant. Enter a number or a string value. - Expression. Enter an expression within parentheses. For example: `(fname + " " + lname)` The expression must include Java supported operators. The PowerCenter Repository Service assigns this value to the output port in the Java transformation.

CHAPTER 9

Importing and Exporting Mapping Specifications

This chapter includes the following topics:

Importing and Exporting Mapping Specifications Overview

You can use the PowerCenter Repository Manager to import the following information from a mapping specification:

- Source definitions
- Target definitions
- Mappings containing source definitions, target definitions, filter, join, lookup, aggregate, and non-aggregate expressions, and rules

For example, a business analyst creates a mapping specification in Microsoft Excel to define a mapping that includes sources, targets, and Filter and Expression transformations. A PowerCenter developer imports the mapping specification to create the PowerCenter objects.

You can use the PowerCenter Repository Manager to export the following information from a PowerCenter repository to a mapping specification:

- Source definitions
- Target definitions
- A valid mapping containing Filter, Joiner, Lookup, Aggregator, or Expression transformations

The PowerCenter Repository Service exports transformations supported in Mapping Analyst for Excel. If you export a mapping with other transformations, target definitions might not contain all ports in the mapping specification.

You might want to export PowerCenter objects to Microsoft Excel because workflows are running in production. However, no documentation about these workflows or mappings exists. A PowerCenter

developer can export the mappings to mapping specifications for documentation and business analyst review. When you export metadata to Excel, the PowerCenter Repository Service exports the Models, Packages, and Mapping worksheets.

Importing Mapping Specifications

Use the Repository Manager to import a mapping specification. You can import source and target definitions, or you can import an entire mapping. You can also import more than one mapping at a time if the mapping specification contains multiple mappings. Mappings do not have to be complete or valid.

The following table describes the PowerCenter objects that the PowerCenter Repository Service creates when you import a mapping specification based on the Standard mapping specification template:

Mapping Analyst for Excel Component	PowerCenter Repository Object
Source information.	Source definitions.
Filter expression to filter data from the pipeline.	Filter transformation.
Join expression to join two sources. To join more than two sources, you can enter multiple join expressions.	Joiner transformation.
Lookup expression to perform a lookup.	Connected Lookup transformation.
Aggregate data transformation expression.	Aggregator transformation.
Single row data transformation expression.	Expression transformation.
Rule.	Java transformation.
Target information.	Target definitions.
Additional comments or descriptions	Descriptions associated with transformations and source and target definitions. Provides additional notes to the PowerCenter developer.
User-defined properties.	Metadata extensions.

When you import a mapping specification, the PowerCenter Repository Service adds the transformations to the mapping in a specific order. After the source definition, the PowerCenter Repository Service adds the Joiner, Lookup, Filter, Expression, Java, and Aggregator transformations, and then the target definition. The PowerCenter developer can edit the mapping and change the order of the transformations.

After you import a mapping specification, review the imported objects. Most mappings imported from a mapping specification require editing before validation.

1. In the Repository Manager, open a folder and click **Repository** > **Import Metadata**.

   The Tool Selection page appears.
2. In the Source Tool field, select Microsoft Office Excel and click **Next**.

   The Microsoft Office Excel Options page appears.
3. Click the Value field for the File option to find and select the mapping specification to import.

4. Click **Next**.

   The PowerCenter Options page appears.

5. Enter the options.

   The following table describes the options that you need to enter:

Option	Description
Export Objects	Objects that you want to import into the PowerCenter repository. Select one of the following options: - Sources, targets, and mappings. Imports an entire mapping from the mapping specification. Use for mapping specifications that describe an entire mapping. - Tables as sources. Imports all tables or files as sources. - Tables as targets. Imports all tables or files as targets. Default is Tables as sources.
Database type	Database type of the source and target databases. Select a database type, Flat File, or select Auto Detect to allow PowerCenter to determine the database type from the mapping specification. Default is Auto Detect.
Database name	Database name for imported source tables. Overrides the source database name entered in the mapping specification. Optional.
Code page	Name of the PowerCenter Repository Service code page. Default is MS 1252.
Export metadata extensions	Import descriptions and comments as PowerCenter metadata extensions. Select one of the following options: - True. Import descriptions and comments. - False. Do not import descriptions and comments. Default is true.
Path to the Informatica installation	Path to the Informatica PowerCenter client binary files. For example, set the PowerCenter client installation to: `C:\Informatica\PowerCenter <Version number>`. Ensure that the path contains **client** and **java** folders. **Note:** If this parameter is not specified when you import the rules worksheet for a Java transformation, then Mapping Analyst for Excel will not generate Java byte code for business rules.

6. Click **Use Defaults** to revert to default options.
7. Click **Next**.

   In the Import Results dialog box, a message appears if the export from the mapping specification is successful. Error messages appear when the export is not successful.

   Click **Show Details** to view log events. You can also click **Save Log** to save the log events to a file.

8. Click **Next**.

   The Source Selection dialog box displays the sources or targets in the mapping specification with all objects selected by default.

9. Select the objects that you want to import and click **Finish**.

The PowerCenter Repository Service imports the metadata from the mapping specification.

If the import folder contains objects of the same name as those you are importing, the Conflict Resolution Wizard displays. For more information about the conflict, click **Compare Conflict**.

Otherwise for each conflicting object, select the object and choose one of the following resolutions:

- Rename. Change the object name when importing it to the target folder.
- Replace. Replace the existing object in the target folder.
- Reuse. Use the existing object in the target folder.
- Skip. Skip importing the object.

You can also apply the same resolution to all sources or to all conflicts.

10. Click **Next**, and then click **Close** to close the Conflict Resolution Wizard.

Exporting Mappings

Use the Repository Manager to export objects from the PowerCenter repository to a mapping specification. You can export source and target definitions, or you can export a valid mapping containing Filter, Joiner, Lookup, Aggregator, and Expression transformations.

The PowerCenter Repository Service does not export Java transformations to the Rules worksheet of the mapping specification.

The PowerCenter Repository Service exports transformations Mapping Analyst for Excel supports. If you export a mapping with other transformations, target definitions might not contain all ports in the mapping specification.

1. In the Repository Manager, open a folder that contains the repository objects that you want to export.
2. In the Navigator, select the mapping or set of sources or targets that you want to export.
3. Click **Repository** > **Export Metadata**.

   The Tool Selection page appears.
4. For Target Tool, select Microsoft Office Excel and click **Next**.

   The Microsoft Office Excel Options page appears.
5. Enter the options.

   The following table describes the options that you need to enter:

Microsoft Excel Option	Description
File	Name of the mapping specification for the exported metadata. Click the Value field to find and select the mapping specification that you want to use. Or, enter a file name with an .xlsx extension for the wizard to create. You can export metadata to the same file multiple times. When you export to a mapping specification that contains existing data, the PowerCenter Repository Service writes information to empty cells or overwrites existing data.
Format	Name of the mapping specification template that you want to use. Select Standard.

6. Click **Export**.

In the Export Results dialog box, a message appears if the export is successful. Error messages display when the export is not successful.

Click **Show Details** to view log events. You can also click **Save Log** to save the log events to a log file.

7. Click **Finish**.

Configuring the Level of Log Events

When you import a mapping specification or export a mapping, Mapping Analyst for Excel writes log events to a file. You can configure the level of log events that Mapping Analyst for Excel writes to the file.

The LogLevel property in the MirSetup.xml file determines the level of log events.

The following table lists the types of log events displayed for each level:

Log Level	Log Events Displayed
1	Fatal errors.
2	All errors.
3	Errors and warnings.
4	Errors, warnings, and status messages.
5	All messages except for debug messages. Default is 5.
6	All messages including debug messages.

1. Open the MIRSetup.xml file located in the following directory:

   ```
   <PowerCenterClientInstallationDir>\client\bin\mimb\conf
   ```

2. Set the LogLevel property to the appropriate level.

   For example, to display all log events including debug messages, set the property to 6:

   ```
   <LogLevel>6</LogLevel>
   ```

3. Save the MIRSetup.xml file.

Troubleshooting the Import and Export of Mapping Specifications

I exported a mapping that contains an Aggregator transformation. However, the exported mapping specification does not include the aggregate expression on the Mappings worksheet.

In the Designer, verify that the expression port in the Aggregator transformation is connected to a mapping target. To export an aggregate expression to a mapping specification, the expression port in the Aggregator transformation must be connected to a mapping target.

INDEX

P

R

S

T

U

V