Informatica®

Informatica® PowerCenter
10.5.6

Mapping Architect for Visio Guide

Informatica PowerCenter Mapping Architect for Visio Guide
10.5.6
May 2024

Publication Date: 2024-05-29

Table of Contents

Preface

Use *PowerCenter® Mapping Architect for Visio Guide* to learn how to create mapping template that represents PowerCenter mapping. You can specify the extraction logic by configuring rules and parameters in a mapping template. You can define and maintain consistent methodology for your data integration projects. You can also create multiple mappings based on the template.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and 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 find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.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

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

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

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica 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 and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find 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 extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Understanding Mapping Architect for Visio

This chapter includes the following topics:

Understanding Mapping Architect for Visio Overview

Use Mapping Architect for Visio to create mapping templates using Microsoft Office Visio. A mapping template is a drawing that represents a PowerCenter mapping. You can configure rules and parameters in a mapping template to specify the extraction logic. Mapping Architect for Visio is installed with the PowerCenter Client.

You can use the mapping templates to provide consistency and improve productivity in the following cases:

- **Define consistent methodology and structure for data integration projects.** Use mapping templates to standardize error handling or slowly changing dimensions mappings. Distribute mapping templates to developers working on data integration projects to share best practices, standards, and special techniques. In addition, after an implementation is complete, developers can reuse mapping template files for similar projects.
- **Generate multiple mappings at one time.** Create one mapping template to generate multiple mappings that have similar structure. For example, create a mapping template that includes the basic design and uses rules and parameters to define the extraction logic. When you use the mapping template to generate multiple mappings, you can specify different values for the parameters. You can use parameters to define data sources, transformation properties, port names, expressions, and other elements that comprise a mapping.
- **Document data flow.** Use mapping templates to document methods to map or process data.

Components of Mapping Architect for Visio

Mapping Architect for Visio has the following components:

- **Visio.** Use Microsoft Office Visio to create a mapping template that you can use to generate multiple PowerCenter mappings.
- **Mapping template.** A drawing in Visio that represents a PowerCenter mapping. Use rules and parameters in a mapping template to specify the transformation logic. Save and publish a mapping template to generate mapping template files.
- **Mapping template files.** Files that Mapping Architect for Visio generates when you save or publish a mapping template:
  - **Mapping template drawing file ([template name].vsd) or ([template name].vsdx).** A file that Mapping Architect for Visio generates when you save a mapping template. You use this file to generate mappings.
  - **Mapping template XML file ([template name].xml).** An XML file that Mapping Architect for Visio generates when you publish a mapping template, or a mapping template drawing file. You use this file to generate mappings.
  - **Mapping template parameter file ([template name]_param.xml).** An XML file that Mapping Architect for Visio generates when you publish a mapping template, or a mapping template drawing file. You can define and save parameter values in the Mapping template parameter file. Use this file to generate mappings.
- **Import Mapping Template Wizard.** A wizard that you use to import a mapping template XML file into the PowerCenter Designer and generate multiple mappings.
- ***mapgen* command line program.** A command line program that you can use to generate an XML representation of the mappings from mapping template files. You can import the mapping XML file that you created into PowerCenter to view the mappings.
- **Informatica mapping templates.** Predefined mapping templates that cover common data warehousing patterns, such as slowly changing dimensions. Use the Informatica mapping templates instead of creating a mapping template.

Note: You must install Microsoft Visio before you use Mapping Architect for Visio. Informatica does not supply or provide license for Microsoft Visio. To use Microsoft Visio, you must get a separate product license from Microsoft.

Mapping Architect for Visio Interface

Mapping Architect for Visio uses the Microsoft Office Visio interface. Use the Informatica toolbar and the Informatica Stencil to create mapping templates. If you are a Microsoft Office Visio expert, you can also use the Visio tools to help you complete the mapping template easily and quickly.

Mapping Architect for Visio provides online help. To view the Mapping Architect for Visio online help, click the help icon on the Informatica toolbar. You must set a default browser on your system to use the online help.

When you work with a mapping template, you use the following main areas:

- **Informatica toolbar.** Displays buttons for tasks you can perform on a mapping template. It also displays the online help button.

- **Drawing window.** Work area for the mapping template. Mapping Architect for Visio initially displays an empty drawing window.
- **Informatica Stencil.** Displays shapes that represent PowerCenter mapping objects.

Using Mapping Architect for Visio

Mapping Architect for Visio includes the Informatica Stencil and the Informatica toolbar that you can use to create mapping template files. You can import the mapping template files into PowerCenter to generate mappings and workflows. Use the Import Mapping Template Wizard or the *mapgen* command line program to generate mappings in PowerCenter.

To create mapping template files and generate PowerCenter mappings, complete the following steps:

1. Create a mapping template with Mapping Architect for Visio.
2. Publish the mapping template to generate the mapping template XML file and the mapping template parameter file.
3. Generate mappings from the mapping template files.

Step 1. Create a Mapping Template

Use one of the following methods to create a mapping template:

- **Create a mapping template manually**. Use the Informatica Stencil to create a mapping template.
- **Import a mapping template from a PowerCenter mapping.** If you have a mapping that you want to use as a basis for the mapping template, export the mapping to an XML file from PowerCenter. Next, import the mapping XML file into Mapping Architect for Visio to create a mapping template.

After you create a mapping template, you can save and publish the mapping template to create the mapping template files.

Step 2. Publish the Mapping Template

Publish the mapping template or the mapping template drawing file using the Mapping Architect for Visio to create the following files:

- **Mapping template XML file.** Use this file to generate multiple mappings with the Import Mapping Template Wizard.
- **Mapping template parameter file**. Use this file to generate multiple mappings with the *mapgen* command line program. You can also use this file to save the parameter values that you specify with the Import Mapping Template Wizard.

Step 3. Generate Multiple Mappings from Mapping Template Files

Use the mapping template files that you created in the Mapping Architect for Visio to generate mappings in PowerCenter. Use the following methods to generate mappings in PowerCenter:

- **Use the Import Mapping Template Wizard.** A wizard in the Designer that generates multiple mappings from a mapping template.

- **Use the *mapgen* command line program.** A command line program that you use to generate a mapping XML file. Import the mapping XML file in PowerCenter to generate mappings. The mapping XML file contains an XML representation of the generated mappings.

Note: When you generate mappings from a mapping template, PowerCenter removes the unused ports.

Using the Import Mapping Template Wizard

Use the Import Mapping Template Wizard to specify parameters and generate mappings in the repository.

Using the mapgen Command Line Program

You can use *mapgen* to generate mappings from the mapping template files.

To generate mappings using the *mapgen* command line program, complete the following steps:

1. Edit the mapping template parameter file to provide the values for the mapping parameters. You can define each mapping you want to create in the mapping template parameter file.
2. Use the *mapgen* command line program to create a mapping XML file.
3. Use the Import Wizard to import the mapping XML file into the PowerCenter repository.

CHAPTER 2

Mapping Templates

This chapter includes the following topics:

Mapping Templates Overview

A mapping template is a drawing in Visio that represents a PowerCenter mapping. You can configure rules and parameters in a mapping template to specify the transformation logic.

Use the Informatica Stencil and the Informatica toolbar in the Mapping Architect for Visio to create a mapping template. The Informatica Stencil contains shapes that represent mapping objects that you can use to create a mapping template. The Informatica toolbar contains buttons for the tasks you can perform on mapping template.

You can create a mapping template manually, or you can create a mapping template by importing a PowerCenter mapping.

Starting Mapping Architect for Visio

To use Mapping Architect for Visio, install the PowerCenter Client. Mapping Architect for Visio is installed in the same directory as other PowerCenter Client tools.

To start Mapping Architect for Visio, click Start > Programs > Informatica PowerCenter [version] > Client > PowerCenter Mapping Architect for Visio. When you start Mapping Architect for Visio, Visio displays an empty drawing window.

Warning: Do not edit MappingTemplate.vst. If you try to create a mapping template after opening MappingTemplate.vst for editing, you might get unexpected results. If you save the mapping template to MappingTemplate.vst, you can cause inconsistencies in Mapping Architect for Visio.

Macros in Mapping Architect for Visio

Mapping Architect for Visio contains unsigned macros. Set the security level in Visio to Medium so you can enable macros when you start Mapping Architect for Visio. If the security level for Microsoft Office Visio is set to high or very high, you cannot run the Mapping Architect for Visio macros.

To set the security level for the Visio, select Tools > Macros > Security from the menu. On the Security Level tab, select Medium.

When you start Mapping Architect for Visio, Visio displays a security warning about viruses in macros. Click Enable Macros to enable the macros for Mapping Architect for Visio.

Informatica Toolbar

The Informatica toolbar contains buttons for tasks you can perform on a mapping template. Use the buttons on the Informatica toolbar to perform the following tasks:

- **Create a mapping template from a mapping XML.** Creates a mapping template from a PowerCenter mapping XML file. Use this option if you have exported a PowerCenter mapping that you want to use as a basis for a mapping template.
- **Validate a mapping template.** Verifies the validity of a mapping template.
- **Publish a mapping template.** Publishes a mapping template or a mapping template drawing file to create a mapping template XML file and mapping template parameter file.
- **Arrange objects in the drawing window.** Arranges the mapping objects on the drawing window in the best way to show the mapping objects and their links clearly and logically.
- **Show all parameters.** Displays the mapping template parameters in a mapping template. You can configure each mapping template parameter to control how you enter values in the Import Mapping Template Wizard. The mapping template parameter names are enclosed by dollar signs, such as $*ParameterName*$.
- **Declare mapping parameters and variables.** Declares mapping parameters and variables. Use mapping parameters and variables to make mappings more flexible. You can define a value for the mapping parameter or variable before you run the session in a PowerCenter client. Mapping parameter and variable names start with two dollar signs, such as $$*ParameterName*.

Informatica Stencil

The Informatica Stencil contains shapes that you can include in the mapping template. Each shape represents a PowerCenter mapping object. To include a mapping object in the mapping template, drag the mapping object to the drawing window. Double-click the mapping object to edit the mapping object properties.

Creating a Mapping Template Manually

You can use the Informatica Stencil and the Informatica toolbar to create a mapping template. Save and publish a mapping template to create the mapping template files.

To create a mapping template manually, complete the following steps:

1. **Start Mapping Architect for Visio**.
2. **Verify that the Informatica Stencil and Informatica toolbar are available**.
3. **Drag the mapping objects from the Informatica Stencil to the drawing window.** Use the mapping objects to create visual representation of the mapping.
4. **Create links.** Create links to connect mapping objects.
5. **Configure link rules.** Configure rules for each link in the mapping template to indicate how data moves from one mapping object to another. Use parameters to make the rules flexible.
6. **Configure the mapping objects.** Add a group or expression required by the transformations in the mapping template. To create multiple mappings, set a parameter for the source or target definition.
7. **Declare mapping parameters and variables to use when you run sessions in PowerCenter.** After you import the mappings created from the mapping template into PowerCenter, you can use the mapping parameters and variables in the session or workflow.
8. **Validate the mapping template.**
9. **Save the mapping template.** Save changes to the mapping template drawing file.
10. **Publish the mapping template.** When you publish the mapping template, Mapping Architect for Visio generates a mapping template XML file and a mapping template parameter file (param.xml).

If you edit the mapping template drawing file after you publish it, you need to publish again. Do not edit the mapping template XML file.

Importing a Mapping Template from a PowerCenter Mapping

If you have a PowerCenter mapping that you want to use as a basis for a mapping template, export the mapping to a mapping XML file and then use the mapping XML file to create a mapping template.

Note: Export the mapping XML file within the current PowerCenter release. Informatica does not support imported objects from a different release.

To import a mapping template from a PowerCenter mapping, complete the following steps:

1. **Export a PowerCenter mapping.** In the Designer, select the mapping that you want to base the mapping template on and export it to an XML file.
2. **Start Mapping Architect for Visio.**
3. **Verify that the Informatica stencil and Informatica toolbar are available**.
4. **Import the mapping.** On the Informatica toolbar, click the Create Template from Mapping XML button. Mapping Architect for Visio determines the mapping objects and links included in the mapping and adds the appropriate objects to the drawing window.
5. **Verify links.** Create or verify links that connect mapping objects.

6. **Configure link rules.** Configure rules for each link in the mapping template to indicate how data moves from one mapping object to another. Use parameters to make the rules flexible.
7. **Configure the mapping objects.** Add a group or expression required by the transformations in the mapping template. To create multiple mappings, set a parameter for the source or target definition.
8. **Declare mapping parameters and variables to use when you run the session in PowerCenter.** After you import the mappings created from the mapping template into PowerCenter, you can use the mapping parameters and variables in the session or workflow.

   Note: If the PowerCenter mapping contains mapping parameters and variables, it is possible that the mapping parameters and variables ($$*ParameterName*) may not work for all mappings you plan to create from the mapping template. Modify or declare new mapping parameters and variables appropriate for running the new mappings created from the mapping template.
9. **Validate the mapping template.**
10. **Save the mapping template.** Save changes to the mapping template drawing file.
11. **Publish the mapping template.** When you publish the mapping template, Mapping Architect for Visio generates a mapping template XML file and a mapping template parameter file (param.xml).

If you make any change to the mapping template after publishing, you need to publish the mapping template again. Do not edit the mapping template XML file.

Note: Mapping Architect for Visio fails to create a mapping template if you import a mapping that includes an unsupported source type, target type, or mapping object.

Related Topics:

- "Mapping Objects Overview" on page 21

Mapping Template Parameters

Use mapping template parameters to configure values for each mapping. You can create more than one mapping from a mapping template. You can use multiple sources and targets with different port names in the mappings. Each mapping you create from the mapping template uses the same mapping objects, but you may need to configure each mapping object in a different way.

When you define a mapping template parameter, enclose the name in dollar signs. For example, you want to create three mappings from one mapping template. Each mapping uses a different source table. When you configure the source definition in the mapping template, you can set the value of the Source Table property to a parameter, such as $Source$. When you specify parameter values, define the three mappings and set the value of the parameter for the source table for each mapping to the specific source table name.

Note: The mapping template parameter ($*ParameterName*$) is not the same as the PowerCenter mapping parameter ($$*ParameterName*).

Mapping template parameter names and values are case sensitive unless otherwise noted. The mapping template parameter name can contain letters, numbers, or underscores (_). Provide a unique name for each parameter. Mapping Architect for Visio does not validate that parameter names are unique in the mapping template. To view a list of all the parameter values in a mapping template, click the Show Parameters button on the Informatica toolbar.

Create parameters to represent the following mapping template components:

- **Mapping object properties.** To configure a mapping object differently for each mapping, specify parameters when you set the properties.

  Note: If you want to create multiple mappings, set a parameter for either the Source or Target Table property in the source or target definition. You can set parameters for both the Source and Target properties.
- **Expressions.** You can use a parameter as part of an expression or in place of an expression.
- **Rules.** You can use a parameter as part of any rule in a link. You can include one or more rules in a parameter.
- **Group names.** To use different group names for different mappings, specify parameters instead of values when you set the group name.

Configuring Mapping Template Parameters

Configure the mapping template parameters to control how you enter values in the Import Mapping Template Wizard. Before you configure, you must define the mapping template parameters in the mapping template.

To configure mapping template parameters in a mapping template:

1. Click the Show Parameters button on the Informatica toolbar.
2. In the Parameter Name field, select the mapping template parameter you want to configure.
3. In the Parameter field, enter a label name.
4. Select a control that you want to use to select how you want to enter values for the mapping template parameters in the Import Mapping Template Wizard. You can choose Combo box or Edit box.
5. Select the control data. You can select Sources, Targets, or NULL.
6. Enter a description.
7. Click OK.

Mapping Template Example

The following example uses parameters and rules to create a mapping template from a mapping. Complete the following steps:

1. Import a mapping template from a PowerCenter mapping.
2. Define parameters and rules.
3. Validate and publish the mapping template.

Step 1. Importing a Mapping Template from a PowerCenter Mapping

In the Designer, select the mapping that you want to base the mapping template on and export it to an XML file.

To import the mapping template, click the Create Template from Mapping XML button. Mapping Architect for Visio determines the mapping objects and links included in the mapping and adds the appropriate objects to the drawing window.

The mapping template includes mapping objects provided by the Informatica Stencil. The source is a flat file that contains the following employee data: Employee Number, First Name, Last Name, Address, and Phone Number. The mapping loads the employee numbers into the target.

You want to create multiple mappings that use the same mapping objects, but have different sources, targets, or data extraction logic.

Step 2. Define Parameter and Rules

Use parameters and rules in the mapping template to define different logic for each mapping. To define parameters, complete the following steps:

1. Set parameters for the source and target definitions.
2. Set a parameter in a link rule to specify which data to load into the target.

To set parameters for the source and target definitions, open each mapping object and specify a parameter for the source table and target table.

The following table shows the source table definition properties:

Property	Value
Transformation Name	Employees
Source Table	$Source$
Database Name	FlatFile
Database Type	Flat File

Next, set a parameter for the link rule to specify which source data you want to load to the target. To update the rule, double-click the link between the source qualifier and the target.

The following example shows a link rule between the source qualifier and the target:

```
Rule Set Name: Rule 1
Named:EMPLOYEE_NUMBER (TO) EMPLOYEE_NUMBER
```

The link rules that appear are based on the original mapping. Create parameters because the mappings that you plan to generate load different source data into the target.

To create a parameter that specifies the data to load into the target, complete the following steps:

1. Delete link rules between the source qualifier and the target.
2. Create a link rule that includes parameters. The following example shows a link rule that includes parameters for the named port:

```
Starting Port Name: $Source$
Ending Port Name: $Target$
```

The Named Port link rule contains two parameters, one for the source port ($Source$) and one for the target port ($Target$).

When you define the $start$ and end parameter values, you can enter the source and target port names for each mapping in the Import Mapping Template Wizard.

Step 3. Validate and Publish the Mapping Template

After you complete the mapping template, click Validate Template to validate the mapping template. Save changes to the mapping template drawing file, and then publish the mapping template.

To publish the mapping template, click Publish Template. Mapping Architect for Visio creates the mapping template XML file and the mapping template parameter file that you can use to import mappings into the repository.

Informatica Mapping Templates

Informatica mapping templates are predefined mapping templates that cover common data warehousing patterns, such as slowly changing dimensions and remove duplicates.

The following templates provide solutions to the most common issues in data warehousing designs:

- **Slowly Changing Dimensions**. Templates to cover types of slowly changing dimensions.
- **Incremental Load**. Templates to load incremental records from the source.
- **Remove Duplicates**. Templates to capture the logic used for identifying and removing duplicate records from the source.

CHAPTER 3

Mapping Objects

This chapter includes the following topics:

Mapping Objects Overview

Use mapping objects from the Informatica Stencil to create mapping templates. The Informatica Stencil includes the following mapping objects:

- **Source definition.** Represents the source.
- **Target definition.** Represents the target.
- **PowerExchange® source definition.** Represents the PowerExchange source.
- **PowerExchange target definition.** Represents the PowerExchange target.
- **Link.** Connects sources, targets, and transformations, and specifies the rules for data movement.
- **Mapplet.** Represents a mapplet.
- **Transformations.** Represents PowerCenter transformation objects that perform different types of data transformation. You can add the following transformations to a mapping template:
  - Aggregator
  - Custom
  - Application Source Qualifier
  - Custom Transformation
  - Expression
  - Filter
  - Joiner
  - Lookup
  - Pipeline Normalizer

- Rank
- Router
- Sequence Generator
- Sorter
- Source Qualifier
- Stored Procedure
- Transaction Control
- Union
- Update Strategy

Note: The Informatica Stencil does not include mapping objects for all transformations.

Configuring Mapping Objects

Configure mapping object properties in a mapping template the same way as you configure the mapping object properties in a mapping. When you create a mapping template from a mapping and you view the mapping object properties, you see the values you configured in the Designer for the mapping object. If you change the transformation properties in the mapping template and import the mapping template into the Designer, you see the values you configured in Mapping Architect for Visio.

Configure the mapping objects to apply to all the mappings that you plan to create from the mapping template. Specify a name or use a parameter for each mapping object, except for source and target definitions. Specify a rule set name for each link in the mapping template. Property names, property values, parameter names, and parameter values are case sensitive unless otherwise noted.

You may want to use parameters to handle different settings for transformations in multiple mappings. For example, you can use a parameter in the SQL query of a source qualifier. When you configure the source qualifier in the mapping template, set the value of the property to a parameter:

```
SQL Query=$SQL_QUERY$
```

In the mapping template parameter file, set the value of the parameter to the SQL statement appropriate for the mapping:

```
$SQL_QUERY$=SELECT * FROM EMPLOYEE
```

Or, you can include a parameter when you set the SQL query:

```
$SQL_QUERY$=SELECT * FROM $SRC_TABLE$
```

In the mapping template parameter file, set the value of the parameter to the source table name:

```
$SRC_TABLE$=EMPLOYEE
```

To configure a mapping object in the mapping template, double-click the mapping object in the drawing window.

Expressions in Transformations

Create expressions for the output ports when you configure the following transformations:

- Aggregator transformation
- Expression transformation

- Rank transformation

When you configure these transformations, you can enter an expression on the Configuration tab in the Transformation Details window.

Mapping Architect for Visio does not validate the expression. Make sure that you enter a valid expression. When you create an expression, use the following rules and guidelines:

- Do not use semicolons in an expression.
- The precision and scale values must be integers that do not exceed 10 digits.
- The value of the precision must be greater than the scale.

By default, the ports for which you create the expressions are output ports. When you create the expression for a transformation port, you can change the port type to variable.

Using the %ALL% Keyword in an Expression

When you add an expression to a transformation, use the %ALL% keyword in the port name and in the expression. The %ALL% keyword acts as a placeholder for the names of all ports in the transformation. Use the %ALL% keyword if you have a number of ports for which you want to use the same expression. Use a precision of 0 if you want the output ports to have the same precision as the input port from which they are created.

When you use the %ALL% keyword in the port name and expression, Mapping Architect for Visio performs the following tasks:

- **Creates an output port with the same name for each input port in the transformation.** To avoid duplicate ports in the transformation, use a prefix or suffix with the %ALL% keyword. Mapping Architect for Visio creates output ports with the input port names plus the prefix or suffix.
- **Sets up the same expression for all output ports.** The datatype that the expression returns becomes the datatype of the output ports.

The following example shows an Aggregator transformation where the %ALL% keyword is used in an aggregate expression. You configure the following properties:

```
Port Name = AVG_%ALL%_out
Expression = AVG(%ALL%)
```

The input ports for the transformation have the following names:

- PRICE
- COST
- QTY_ON_HAND

The Aggregator transformation will have the following output ports:

Port Name	Datatype	Expression
AVG_PRICE_out	Decimal	AVG(PRICE)
AVG_COST_out	Decimal	AVG(COST)
AVG_QTY_ON_HAND_out	Decimal	AVG(QTY_ON_HAND)

Groups in Multi-Group Transformations

Create groups when you configure the following multi-group transformations:

- Router transformation
- Union transformation

You can create groups on the Configuration tab in the Transformation Details window. Associate the groups you define in the transformation with the rules you create in links that connect to the transformation.

Reusable Transformations

You can configure a transformation to be reusable. Set Reusable to Yes in the mapping object properties. Default is No.

Groups in Multi-Group Sources

Create groups when you configure the PowerExchange sources.

You can create, edit, or delete groups on the Configuration tab of the PowerExchange sources. Associate the groups you define in the PowerExchange source with the rules you create in links that connect to the source.

Source Definitions and Target Definitions

You can configure the following source and target types in Mapping Architect for Visio:

- Flat file
- Relational databases
- PowerExchange sources and targets

The following table describes the required property for the source and target definitions:

Property	Description
Transformation Name	Name of the source or target definition.

PowerExchange Source Definitions and Target Definitions

You can create a mapping template that contains PowerExchange sources and targets. You can also import a mapping template from a PowerCenter mapping that contains PowerExchange sources and targets. Install the PowerExchange product you require before you create mappings and mapping templates that include PowerExchange sources and targets.

Note: To import a PowerCenter mapping template with a PowerExchange target, verify that the PowerExchange target is in the current working folder. Select the PowerExchange target when you import the PowerCenter mapping template.

A PowerExchange data source can contain groups. You can add, edit, or delete the groups in a PowerExchange source. The PowerExchange target definition does not contain groups.

You can add the following PowerExchange sources and targets to a mapping template:

- DB2 for i5/OS
- DB2 for z/OS
- VSAM (z/OS)
- IMS (z/OS)
- Adabas (z/OS)
- Datacom (z/OS)
- IDMS (z/OS)

Automatically Create Targets

You can configure the Import Mapping Template Wizard to automatically create targets for each mapping it generates. In the target definition properties, set Always Create Target to True. If you set the property to True, the active folder must contain the shortcut definition. If you set the property to False, the Import Mapping Template Wizard does not create a target definition unless you do not export the table definition and the table definition does not exist in the tabledefs folder. Default is False.

Shortcuts

You can configure a source definition, target definition, or transformation to use a shortcut. When you import the mapping template, the Designer creates the reference to the shortcut object in the active folder. To configure a shortcut, set IsShortcut to True in the source definition, target definition, or transformation properties. Default is False.

Shortcut to Source Definition and Target Definition

You cannot create a shortcut to an object by specifying the Always Create Target property in target definition. If you set Always Create Target property and IsShortcut property to true, Mapping Architect for Visio creates a mapping with the target object.

The Import Mapping Template Wizard ignores IsShortcut if you use a parameter for the source or the target properties. Instead, you select the shortcut from the active folder when you specify parameter values in the Import Mapping Template Wizard.

If lookup is on shortcut to a target, replace the lookup table name with the shortcut table name in Microsoft Office Visio before you generate the mapping. You can also parameterize the lookup table name in Mapping Architect for Visio.

Shortcut to Transformation

You can create a mapping template that contains a shortcut to a transformation. You can also import a mapping template from a mapping that contains shortcuts to reusable transformations.

When you import a mapping template from a mapping that contains a transformation shortcut, the Is Shortcut property is set to True.

The shortcut to the transformation must be present in the folder in which you generate the mapping. If the shortcut to the transformation is not in the working folder, the Import Mapping Template Wizard creates a reusable or non-reusable transformation, based on the value of the Reusable property. If the Reusable

property is set to YES, the Import Mapping Template Wizard creates a reusable transformation in the working folder. If the Reusable property is set to NO, the Import Mapping Template Wizard creates a non-reusable transformation in the working folder.

Rules and Guidelines for Editing a Transformation Shortcut

Use the following rules and guidelines when you edit a transformation shortcut in a mapping template:

- When you edit a transformation shortcut, set the Is Shortcut property to False. You cannot edit the other properties of the transformation shortcut if the Is Shortcut property is set to True. Set the Is Shortcut property back to True when you finish editing.
- The shortcut in a mapping template must point to the same type of transformation as the shortcut in the PowerCenter repository. Edit the properties of the transformation shortcut to match an existing transformation of the same type in the PowerCenter repository. For example, if you create a shortcut to a Filter transformation in a mapping template, the properties of the shortcut must match a shortcut to a Filter transformation in the PowerCenter repository.
- The name and properties of a transformation shortcut in a mapping template must match the name and properties of a transformation shortcut in the PowerCenter folder where you generate the mapping. If a transformation shortcut that you add to a mapping template does not exist in the PowerCenter repository, you must create the shortcut in the PowerCenter folder where you generate the mapping.

Mapplet

You can create a mapping template from a mapping that includes a mapplet. Or, you can import a mapplet and add the related mapping objects to the mapping template. The mapplet properties are read-only. The mapplet transformations and transformation logic are not visible in Mapping Architect for Visio.

Use mapplets only when you create a mapping template from a mapping. If you manually add a mapplet to a mapping template, or if you use a mapplet as a source, the Import Mapping Template Wizard may not generate the mapping correctly.

The following table describes the required property for a mapplet:

Property	Mandatory field
Transformation Name	Name of the mapplet.

Transformations

Use Mapping Architect for Visio to design high-level data flow patterns in Microsoft Visio. You can create mapping templates by using the transformations that are available on the Informatica Stencil. The Informatica toolbar provides buttons to complete the tasks of creating a mapping template.

Aggregator Transformation

Use the Aggregator transformation to perform aggregate calculations, such as averages and sums.

The following table describes the required properties for the Aggregator transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Expression	Performs a calculation based on values within a single row. For example, based on the price and quantity of a particular item, you can calculate the total purchase price for that line item in an order.
GroupBy	Groups for aggregation.

Custom Transformations and Transformations Built Using Custom Transformations

The Custom transformation shape represents the following types of transformations:

- Custom transformation
- Transformations built using the Custom transformation:
  - Data Masking
  - HTTP
  - Java
  - SQL

The shape can represent a Custom transformation or a transformation built using a Custom transformation based on how you create the mapping template.

If you create a mapping template, the Custom transformation shape represents a Custom transformation. The shape cannot represent one of the transformations built using the Custom transformation.

If you import a mapping template from a mapping that contains Data Masking, HTTP, Java, or SQL transformations, the Custom transformation shape can represent these transformations. When you import a mapping template, Mapping Architect for Visio retains the transformation properties and groups. You cannot import a mapping template from a mapping that contains any other type of Custom transformation.

Custom Transformation

You can include a Custom transformation in a mapping template or import a template from a mapping with a Custom transformation.

You can configure the properties and create input and output groups for the Custom transformation. When you include a Custom transformation in a mapping template, you can set it to an active or passive Custom transformation. Set the IsActive property to Yes for an active Custom transformation, and set the property to No for a passive Custom transformation.

The following table describes the required properties and values for an active or passive Custom transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Language	Language used for the procedure code.

Transformation Property	Description
Runtime Location	Location that contains the DLL or shared library. Default is $PMExtProcDir.
Tracing Level	Amount of detail displayed in the session log for this transformation. Set the value to Normal for both active and passive Custom transformations.
Is Partitionable	Indicates if you can create multiple partitions in a pipeline that uses this transformation. Set the value to No for both active and passive Custom transformations.
Inputs Must Block	Indicates if the procedure associated with the transformation must be able to block incoming data. Set the value to Yes for both active and passive Custom transformations.
IsActive	Indicates if this transformation is an active or passive transformation. Set the value to Yes for an active Custom transformation, and No for a passive Custom transformation.
Update Strategy Transformation	Indicates if this transformation defines the update strategy for output rows. Set the value to No for a passive Custom transformation. For an active Custom transformation, you can set the value to Yes or No.
Transformation Scope	Indicates how the Integration Service applies the transformation logic to incoming data. Set the value to All Input for an active Custom transformation and Row for a passive Custom transformation.
Generate Transaction	Indicates if this transformation can generate transactions. When a Custom transformation generates transactions, it generates transactions for all output groups. Set the value to No for a passive Custom transformation. For an active Custom transformation, you can set the value to Yes or No.
Output is Repeatable	Indicates if the order of the output data is consistent between session runs. Set the value to Never for an active Custom transformation and Based on Input Order for a passive Custom transformation.
Requires Single Thread Per Partition	Indicates if the Integration Service processes each partition at the procedure with one thread. Set the value to Yes for both active and passive Custom transformations.
Output Is Deterministic	Indicates whether the transformation generates consistent output data between session runs. Set the value to Yes for both active and passive Custom transformations.

Transformations Built Using the Custom Transformation

You can import a mapping template from a mapping that contains the following transformations:

- Data Masking
- HTTP
- Java
- SQL

If you create a mapping template in Mapping Architect for Visio, you cannot include these types of transformations.

You can view the common transformation properties such as the type of transformation, transformation name, and module identifier. You cannot edit the properties.

Expression Transformation

Use the Expression transformation to calculate values in a single row.

The following table describes the required properties for the Expression transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Expression	Performs a calculation based on values within a single row. For example, based on the price and quantity of a particular item, you can calculate the total purchase price for that line item in an order.

Filter Transformation

Use the Filter transformation to filter out rows in a mapping.

The following table describes the required properties for the Filter transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Filter Condition	An expression that returns TRUE or FALSE.

Joiner Transformation

Use the Joiner transformation to join data from two sources. One source is the master pipeline. The other is the detail pipeline. In Mapping Architect for Visio, two links connect to a Joiner transformation.

Use the following guidelines when you create links to the Joiner transformation:

- The rules for the links to the Joiner transformation must have unique names. If you assign duplicate names to the rules, mapping generation fails in the Designer.
- The links to the Joiner transformation must end in ports with unique names. When you create the links to the Joiner transformation, verify that the ending ports do not have duplicate names.

For example, two sources have the same port names that you want to link to a Joiner transformation. If both the master and detail links use the All Ports rule, the ending ports will have duplicate names, as shown in the following situation:

- The master link contains starting port names A, B, and C and the link uses the All Ports rule. Therefore, the starting ports link to ending port names A, B, and C.
- The detail link contains starting port names A, B, and E and the link uses the All Ports rule. Therefore, the starting ports link to ending port names A, B, and E.

The input ports in the Joiner transformation have the port names A, B, C, A, B, E. Ports A and B are duplicate port names and are therefore invalid.

If the master and detail links contain the same starting ports, use the Named Port, Pattern, or Dictionary rules to ensure that ending ports will contain unique port names. Avoid using All Ports, Primary Key, Foreign Key, and Datatype rules which link starting ports to ending ports with the same name.

The following table describes the required properties for the Joiner transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Cache Directory	Specifies the directory used to cache master or detail rows and the index to these rows. By default, the cache files are created in a directory specified by the process variable $PMCacheDir. If you override the directory, make sure the directory exists and contains enough disk space for the cache files. The directory can be a mapped or mounted drive.

The following table describes the Pattern rule that you can use to ensure that there are no duplicate port names in the Joiner transformation:

Link	Pattern	Port Names
Master Link	Starting port pattern: [A-Za-z_][A-Za-z_0-9]* Ending port pattern: $0_1	Starting ports A links to ending port A1. Starting ports B links to ending port B1. Starting ports C links to ending port C1.
Detail Link	Starting port pattern: [A-Za-z_][A-Za-z_0-9]* Ending port pattern: $0_2	Starting ports A links to ending port A2. Starting ports B links to ending port B2. Starting ports C links to ending port C2.

Lookup Transformation

Use the Lookup transformation to look up data in a relational database. The Lookup transformation in a mapping template cannot use a flat file or a source qualifier as a lookup source.

If you use a Lookup transformation in a mapping template, you must manually export the lookup source. The Import Mapping Template Wizard does not export the source that the Lookup transformation references.

Configure the following required properties for the Lookup transformation:

Property	Cache Type	Description

Lookup Condition	All	The lookup condition must have the following syntax:
		`<LookupTableColumnName> <operator> <TransformationPort>`
		You can use one of the following operators or combination of operators:
		`=, <, <=, >, >=, !=`
		Do not use any other combination of operators. The following operators are invalid when you create mappings from the mapping template:
		`=>, =<, <>`
		Mapping Architect for Visio does not validate the Lookup condition. Ensure that you use the correct syntax for the condition.

Lookup table name	Table Name	Name of the table from which the transformation looks up and caches values.
Lookup source type	n/a	Source type of the Lookup transformation. The Lookup transformation must use a relational database source.
Input Ports	All	List of input ports. Separate entries with commas.
Comparison Ports	Dynamic Lookup Cache	List of associated input ports that the Integration Service uses to compare values.

Default values appear for many of the properties.

Pipeline Normalizer Transformation

The pipeline Normalizer transformation receives a row that contains multiple-occurring columns and returns a row for each instance of the multiple-occurring data. The transformation processes multiple-occurring columns or multiple-occurring groups of columns in each source row.

For example, you might have a relational table that stores four quarters of sales by store. You need to create a row for each sales occurrence. You can configure a pipeline Normalizer transformation to return a separate row for each quarter.

Use the pipeline Normalizer transformation to process multiple-occurring data from relational tables or flat files. You cannot use a VSAM Normalizer in Mapping Architect for Visio.

The following table describes the required properties for the pipeline Normalizer transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Column Name	Name of the source column.

Transformation Property	Description
Level	Group columns. Columns in the same group occur beneath a column with a lower level number. When each column is the same level, the transformation contains no column groups.
Occurs	The number of instances of a column or group of columns in the source row.
Datatype	The transformation column datatype. Can be String, Nstring, or Number.
Prec	Precision. Length of the column.
Scale	Number of decimal positions for a numeric column.
Key Type	Key type for the column. Can be Primary, Alternate, Primary Duplicate, Alternate Duplicate, or Not a Key.

You must set the All Ports rule to link ports in the previous transformation or source qualifier to ports in a pipeline Normalizer transformation. Depending on the number of occurrences for a field, Mapping Architect for Visio can generate additional ports.

Steps to Create a Pipeline Normalizer Transformation

To create a pipeline Normalizer transformation:

1. Drag the pipeline Normalizer transformation shape from the Informatica Stencil to the drawing window.
2. Link the pipeline Normalizer transformation to the previous transformation or source qualifier in the mapping and set the All Ports rule.
3. In the Configuration tab, click New Column to add a new column.

   By default, all new records are at column-level number 1, which is the Root level.
4. From the list of parent records, select the parent record in which to add a new column.
5. Enter the name, datatype, precision, scale, and keytype.
6. To create a multiple-occurring column, enter the number of occurrences in the Occurs column.
7. To create a group of multiple-occurring columns, select the column and click Level.

   Mapping Architect for Visio adds a NEWRECORD group level column above the selected column. NEWRECORD becomes Level 1. The selected column becomes Level 2. You can rename the NEWRECORD column.

   You can change the column level for other columns to add them to the same group.
8. To make a column the same level as the column above it, select the column and click Level.
9. Change the occurrence at the group level to make the group of columns multiple-occurring.
10. Click Apply to save the columns and create input and output ports.

    Mapping Architect for Visio creates the Normalizer transformation input and output ports. It also creates the generated key columns and a column ID for each multiple-occurring column or group of columns.
11. In the Properties tab, you can change the tracing level, set the Reusable or Is Shortcut properties, or reset the generated key sequence numbers after the next session.

Rank Transformation

Use a Rank transformation to return the largest or smallest numeric value in a port or group.

The following table describes the required properties for the Rank transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Expression	Performs a calculation based on values within a single row. For example, based on the price and quantity of a particular item, you can calculate the total purchase price for that line item in an order.
GroupBy	Groups for ranking.

Router Transformation

Use the Router transformation to evaluate data based on one or more conditions and route the rows of data that meet each condition into a separate output group. Optionally, you can route rows of data that do not meet any of the conditions to a default output group.

When you configure a Router transformation, you can define one or more router groups. These groups have the same ports. When you generate mappings from the mapping template, unique numeric suffixes are assigned to the names of the output ports in each group.

The following table describes the required properties for the Router transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Group Filter Condition	Returns TRUE or FALSE for each row that passes through the transformation, depending on whether a row satisfies the specified condition.

The suffix follows the order in which the groups are listed in the Router configuration window.

For example, the following table describes the groups that you define for a Router transformation:

Group Name	Group Filter Condition
DEFAULT	OutputPortA1 OutputPortB1 OutputPortC1
LONDON_GROUP	City=”London”
SEATTLE_GROUP	City=”Seattle”

When you generate mappings from a mapping template, PowerCenter assigns the following suffixes to the groups:

Group	Suffix	Examples
DEFAULT	1	OutputPortA1 OutputPortB1 OutputPortC1
LONDON_GROUP	2	OutputPortA2 OutputPortB2 OutputPortC2
SEATTLE_GROUP	3	OutputPortA3 OutputPortB3 OutputPortC3

When you create the rules for the links that start from a Router transformation, identify the group associated with the link. The rules must refer to port names with the suffix for the associated group.

If you create rules that refer to port names with the suffix for another group, you cannot generate mappings from the mapping template.

Sequence Generator Transformation

The Sequence Generator transformation generates numeric values. Use the Sequence Generator to create unique primary key values, replace missing primary keys, or cycle through a sequential range of numbers.

The following table describes the required property for the Sequence Generator transformation:

Transformation Property	Mandatory field
Transformation Name	Name of the transformation.

Sorter Transformation

Use the Sorter transformations to sort data. You can sort data in ascending or descending order according to a specified sort key.

The following table describes the required properties for the Sorter transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Transformation Scope	Specifies how the Integration Service applies the transformation logic to incoming data.
Null Treated Low	Treats null values as lower than any other value in a Sorter transformation.
Distinct	Treats output rows as distinct in the Sorter transformation.

Transformation Property	Description
Work Directory	Work directory for the Integration Service to create temporary files while it sorts data.
Sorter Cache Size	Maximum amount of memory that the Integration Service can allocate to perform the sort operation.

Source Qualifier Transformation

When you add a relational or a flat file source definition to a mapping, you need to connect it to a Source Qualifier transformation.

The following table describes the required property for the Source Qualifier transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.

Stored Procedure Transformation

Use a Stored Procedure transformation to automate tasks that are too complicated for standard SQL statements. The following table describes the required properties for the Stored Procedure transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Stored Procedure Name	Name of the stored procedure in the database.
Stored Procedure Type	Determines when the Integration Service calls the stored procedure.

Transaction Control Transformation

PowerCenter lets you control commit and roll back transactions based on a set of rows that pass through a Transaction Control transformation.

The following table describes the required properties for the Transaction Control transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Transaction Control Condition	Conditions to commit and roll back transactions from transactional targets.

Union Transformation

Use the Union transformation to merge data from multiple pipelines or pipeline branches into one pipeline branch. It merges data from multiple sources similar to the UNION ALL SQL statement to combine the results from two or more SQL statements.

The following table describes the required properties for the Union transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.
Module Identifier	Name of the module.
Function Identifier	Name of the procedure in the module.
Transformation Scope	Indicates how the Integration Service applies the transformation logic to incoming data.
Generate Transaction	Indicates if the Union transformation can generate transactions.
Output Is Repeatable	Indicates if the order of the output data is consistent between session runs.
Group Name	Creates multiple input groups and multiple output groups in a transformation.
Group Ports	Creates and deletes ports for input groups and output groups. When you create a port, PowerCenter adds it below the currently selected row or group.

Update Strategy Transformation

Use the Update Strategy transformation to determine how to update the target. You can configure the transformation to insert, delete, update, or reject rows.

The following table describes the required property for the Update Strategy transformation:

Transformation Property	Description
Transformation Name	Name of the transformation.

CHAPTER 4

Mapping Template Rules

This chapter includes the following topics:

Mapping Template Rules Overview

When you create a link between mapping objects, create a set of rules to indicate the movement of data from one mapping object to the next. Every link in a mapping template must have an associated rule set. A rule set can have one or more rules.

The following table describes the rules that you can create in the mapping template:

Rule Name	Description
All Ports	Links all the ports in a mapping object to all the ports in another mapping object.
Datatype	Links all the ports of the selected datatype in a mapping object to all ports of the same datatype in another mapping object.
Dictionary	Links specified starting and ending ports based on the contents of a referenced text file.
Foreign Key	Links all foreign key ports in a mapping object to foreign key ports with the same names in another mapping object.
Named Port	Links all ports with the specified names in a mapping object to ports with the specified name in another mapping object.

Rule Name	Description
Parameter	Replaces the rules on a link with the rules included in the parameter you specify.
Pattern	Links all ports in a mapping object with names that match the specified starting port pattern to all ports in another mapping object with names that match the specified ending port pattern.
Primary Key	Links the primary key port in a mapping object to the primary key port with the same name in another mapping object.
Group Name	Links ports from one group to ports in a different group based on the specfied rules.

Creating and Configuring Rules

Rules determine how mapping objects are linked for each mapping that is generated from the mapping template.

Rule Order

You can set the order in which Mapping Architect for Visio evaluates the rules in a rule set. The order of the rules affects the list of ports for which Mapping Architect for Visio creates links. After you create the rules for a rule set, you can change the order in which you want Mapping Architect for Visio to evaluate them.

Mapping Architect for Visio evaluates the rules from top to bottom and creates one link for each port. Two rules can affect the same port. For example, a Named Port rule or a Foreign Key rule at the bottom of the rule set may override a Named Port rule at the top of the rule set.

The following table shows a sample list of starting ports for a link:

Port Name	Port Type	Datatype
ITEM_ID	Primary key	-
ITEM_NAME	-	String
ITEM_DESC	-	String
PRICE	-	Decimal
DISCONTINUED_FLAG	-	Boolean
MANUFACTURER_ID	Foreign key	-
DISTRIBUTOR_ID	Foreign key	-

Based on the example ports in "Rule Order" on page 38, Mapping Architect for Visio creates links for the sample rule sets.

The following table shows how Mapping Architect for Visio evaluates the rules in a rule set.

This rule set...	Creates these links...
All Ports Named:ITEM_DESC (TO) DESCRIPTION Named:PRICE (TO) ITEM_PRICE DataType:decimal	ITEM_ID links to ITEM_ID ITEM_NAME links to ITEM_NAME ITEM_DESC links to DESCRIPTION PRICE links to PRICE DISCONTINUED_FLAG links to DISCONTINUED_FLAG MANUFACTURER_ID links to MANUFACTURER_ID DISTRIBUTOR_ID links to DISTRIBUTOR_ID **Note:** The Named Port rules for ITEM_DESCRIPTION and PRICE override the All Ports rule. However, the Datatype rule overrides the Named Port rule for PRICE.
Named:ITEM_DESC (TO) DESCRIPTION Named:PRICE (TO) ITEM_PRICE Pattern:[A-Za-z_][A-Za-z_0-9]*_ID$ (TO) NEW_$0 All Ports	ITEM_ID links to ITEM_ID ITEM_NAME links to ITEM_NAME ITEM_DESC links to ITEM_DESC PRICE links to PRICE DISCONTINUED_FLAG links to DISCONTINUED_FLAG MANUFACTURER_ID links to MANUFACTURER_ID DISTRIBUTOR_ID links to DISTRIBUTOR_ID **Note:** The All Ports rule overrides all previous rules.
Pattern:^ITEM.*_.*$ (TO) $0_IN Primary Key Foreign Key	ITEM_ID links to ITEM_ID ITEM_NAME links to ITEM_NAME_IN ITEM_DESC links to ITEM_DESC_IN MANUFACTURER_ID links to MANUFACTURER_ID DISTRIBUTOR_ID links to DISTRIBUTOR_ID **Note:** The Primary Key rule overrides the Pattern rule for ITEM_ID.

Including or Excluding Rules

When you create a rule, you can include or exclude ports that conform to a rule. If you include ports, Mapping Architect for Visio creates a link for all the starting ports that conform to the rule. If you create more than one rule in a rule set, Mapping Architect for Visio creates links for the list of starting ports that conform to rules, based on the order of the rules.

You can also create rules to exclude ports from the list. The order in which an exclusion rule appears in a rule set is important. An exclusion rule can exclude ports only from the list of included ports. Therefore, an exclusion rule is valid only when it follows one or more rules that create links for ports. You cannot set up a rule set that contains only an exclusion rule. You cannot create a rule set where the first rule is an exclusion rule.

Based on the example ports in "Rule Order" on page 38, Mapping Architect for Visio creates links for the sample rule sets.

The following table shows how Mapping Architect for Visio evaluates the rules in rule sets that contain exclusion rules:

This rule set...	Creates these links...
All Ports Named:ITEM_DESC (TO) DESCRIPTION Named:PRICE (TO) ITEM_PRICE DataType:decimal EXCLUDE Foreign Key	ITEM_ID links to ITEM_ID ITEM_NAME links to ITEM_NAME ITEM_DESC links to DESCRIPTION PRICE links to PRICE DISCONTINUED_FLAG links to DISCONTINUED_FLAG **Note:** The All Ports rule creates links for all ports. The Named Port rules for ITEM_DESC and PRICE override the All Ports rule. The Datatype rule overrides the Named Port rule for PRICE. The Exclude Foreign Key rule removes the links for the foreign key ports.
Named:ITEM_DESC (TO) DESCRIPTION Named:PRICE (TO) ITEM_PRICE EXCLUDE Pattern:^ITEM.*$ (TO) $0	PRICE links to ITEM_PRICE **Note:** The Named Port rules create links for ITEM_DESC and PRICE. However, the Exclude Pattern rule removes the link for ports that start with the string *ITEM*.
Pattern:^ITEM.*_.*$ (TO) $0_IN EXCLUDE Primary Key Foreign Key	ITEM_NAME links to ITEM_NAME_IN ITEM_DESC links to ITEM_DESC_IN MANUFACTURER_ID links to MANUFACTURER_ID DISTRIBUTOR_ID links to DISTRIBUTOR_ID **Note:** The Pattern rule creates links for ITEM_ID, ITEM_NAME, and ITEM_DESC. However, the Exclude Primary Key rule removes the link for ITEM_ID. The Foreign Key rule creates links for MANUFACTURER_ID and DISTRIBUTOR_ID.

Multi-Group Transformations

Many transformations have one group of input ports and one group of output ports. You do not have to specify the group with which to associate a rule. Some transformations can have multiple input or output groups.

You can use the following multi-group transformations in a mapping template:

- Router transformation
- Union transformation

When you create a rule on a link for a multi-group transformation, specify the group to which you want to associate the rule.

For example, a Router transformation has one input group and can have multiple output groups. You create a link from a Source Qualifier transformation to a Router transformation and a link from the Router transformation to a target definition. When you create a rule on the link from the Router transformation to the target definition, specify the Router group to associate with the rule.

When you create groups for a transformation, Mapping Architect for Visio adds the group name enclosed in curly brackets ({}) to any rule associated with a group.

For example, if you create an All Ports rule in a link that connects from a Router transformation, Mapping Architect for Visio adds the router group name in front of the rule:

```
{SEATTLE_GROUP}All Ports
```

Multi-Group Sources and Targets

PowerExchange sources have output groups and PowerExchange targets have input groups.

When you create a rule on a link for a PowerExchange source or target, specify the group to which you want to associate the rule.

Creating Rules for a Link

You must create a rule for each link in the mapping template.

To create link rules:

1. In the drawing window, double-click a link.

   The Link Rules window appears.
2. Enter a name for the rule set.

   You cannot include spaces in the Rule Set Name.
3. Click New Rule.

   The Define Link Rule dialog box appears.
4. Optionally, select Exclude to exclude starting ports that conform to the rule.
5. Select the rule you want to add to the rule set:
   - For a Named Port rule, enter the starting and ending port names.
   - For a Datatype rule, select the datatype.
   - For a Pattern rule, enter the starting and ending port patterns.
   - For a Dictionary rule, select the dictionary text file.
   - For a Parameter rule, you can select any defined parameter except source and target parameters.
6. If the link connects to a multi-group transformation, such as a Router transformation or Union transformation, or to a mapplet, enter the group name.
7. Click OK to save the rule.
8. To add another rule, click New Rule on the Link Rules window.
9. After you create all the rules you require for the link, click OK in the Link Rules window to save the rule set.

All Ports

The All Ports rule links all ports in a mapping object to all ports in another mapping object. For example, if you create an All Ports rule on the link from a source definition to a source qualifier, Mapping Architect for Visio creates ports in the source qualifier with the same names and properties as those in the source definition. Mapping Architect for Visio also creates links between ports of the same name.

Syntax

Use the following syntax when you include this rule in a Parameter rule:

```
All Ports
```

Datatype

The Datatype rule links all ports of the selected datatype in a mapping object to all ports of the same datatype in another mapping object. You can select the following datatypes:

- binary
- date/time
- decimal
- double
- integer
- nstring
- ntext
- real
- small integer
- string
- text

For example, if you create a Datatype rule on a link from a source definition to a source qualifier and select the integer datatype, then the source qualifier includes all datatype integer ports of the source definition. The ports on the source qualifier have the same names and properties as the source definition. Links are established between ports of the same name.

Syntax

Use the following syntax when you include this rule in a Parameter rule:

```
DataType:<Datatype>
```

Dictionary

The Dictionary rule links starting ports with specified names to ending ports with the same names. The Dictionary rule looks for the list of port names in a text file. When you create a Dictionary rule, provide the name of the text file. Use the Dictionary rule when you want to use the Named Port rule for a large number of ports.

Unlike the Named Port rule, you cannot use parameters or a combination of parameter and character strings to specify the name of a port in the text file for the Dictionary rule.

Format for the Dictionary Text File

Use the following syntax to specify the starting ports and ending ports in the text file for the Dictionary rule:

```
<StartingPortName> = <EndingPortName>
```

For example, the following list is valid content for a Dictionary rule text file:

```
ITEM_NAME = ITEM_NAME
ITEM_PRICE = ITEM_PRICE_OUT
```

The following list is invalid:

```
$PORT_NAME$ = $PORT_NAME$_OUT
$START_PORT$ = $END_PORT$
```

The name of the text file can be any valid Windows file name. Mapping Architect for Visio does not validate the text file. Verify that the dictionary text file contains rules with valid syntax.

Syntax

Use the following syntax when you include this rule in a Parameter rule:

```
Dictionary:<PathName>\<TextFileName>
```

Foreign Key

The Foreign Key rule links all foreign key ports in a mapping object to foreign key ports with the same names in another mapping object.

For example, if you create a Foreign Key rule on a link from a Filter transformation to a target definition, Mapping Architect for Visio creates all foreign key ports in the target definition with the same names and properties as the Filter Transformation. Mapping Architect for Visio also creates links between foreign key ports of the same name.

Syntax

Use the following syntax when you include this rule in a Parameter rule:

```
Foreign Key
```

Named Port

The Named Port rule links all ports with the specified names in a mapping object to ports with the specified name in another mapping object.

You can use parameters instead of character strings to specify the name. You can also combine a parameter and a character string to specify the name. If you use parameters, provide the parameter value in the parameter file you generate for the mapping template. You cannot include spaces in port names.

Note: If the ending port name is blank, the starting port links to an ending port with the same name.

The following table describes how you can specify port names in a Named Port rule:

Parameter or String	Description
Starting Port: ITEM_NAME Ending Port: <blank>	Starting port named ITEM_NAME links to an ending port ITEM_NAME.
Starting Port: ITEM_NAME Ending Port: PRODUCT_NAME	Starting port named ITEM_NAME links to an ending port PRODUCT_NAME.
Starting Port: ITEM,CODE Ending Port: <blank>	Starting port named ITEM links to an ending port named ITEM. Starting port named CODE links to an ending port named CODE.
Starting Port: ITEM,CODE Ending Port: ITEM_DESC,ITEM_ID	Starting port named ITEM links to an ending port named ITEM_DESC. Starting port named CODE links to an ending port named ITEM_ID.

Parameter or String	Description

Starting Port: $PORT_NAME$
Ending Port: <blank>
	Starting port with a name that matches the value of the $PORT_NAME$ parameter links to the ending port with the same name. You must provide the parameter value in the mapping template parameter file. The parameter value can include a comma-separated list of port names.
	Examples of parameter values:
	`<PARAM NAME="$START_PORT$" VALUE="PORT1" />`
	`<PARAM NAME="$END_PORT$" VALUE="" />`
	Starting port named PORT1 links to an ending port named PORT1.
	`<PARAM NAME="$START_PORT$" VALUE="PORT1,PORT2" />`
	`<PARAM NAME="$END_PORT$" VALUE="" />`
	Starting port named PORT1 links to an ending port named PORT1. Starting port named PORT2 links to an ending port named PORT2.

Starting Port: $PORT_NAME$ Ending Port: $PORT_NAME$	Starting port with a name that matches the value of the $PORT_NAME$ parameter links to the ending port with the same name. You must provide the parameter value in the mapping template parameter file.
Starting Port: $PORT_NAME$ Ending Port: IN_$PORT_NAME$	Starting port with a name that matches the value of the $PORT_NAME$ parameter links to the ending port with the same name prefixed by *IN_* . You must provide the parameter value in the mapping template parameter file. For example, you set the parameter value in the parameter: `<PARAM NAME="$PORT_NAME$" VALUE="ITEM_NAME" />` Starting port named ITEM_NAME links to an ending port named IN_ITEM_NAME.

Starting Port: $START_PORT$
Ending Port: END_PORT
	Starting port with a name that matches the value of the $START_PORT$ parameter links to the ending port with a name that matches the value of the END_PORT parameter. You must provide the parameter value in the mapping template parameter file. The parameter value can include a comma-separated list of port names.
	Examples of parameter values:
	`<PARAM NAME="$START_PORT$" VALUE="ITEM_NAME" />`
	`<PARAM NAME="$END_PORT$" VALUE="PRODUCT_NAME" />`
	Starting port named ITEM_NAME links to an ending port named PRODUCT_NAME.
	`<PARAM NAME="$START_PORT$" VALUE="PORT1,PORT2" />`
	`<PARAM NAME="$END_PORT$" VALUE="PORTA,PORTB" />`
	Starting port named port1 links to an ending PORT1 named PORTA. The starting port named PORT2 links to an ending port named PORTB.

Syntax

Use the following syntax when you include this rule in a Parameter rule:

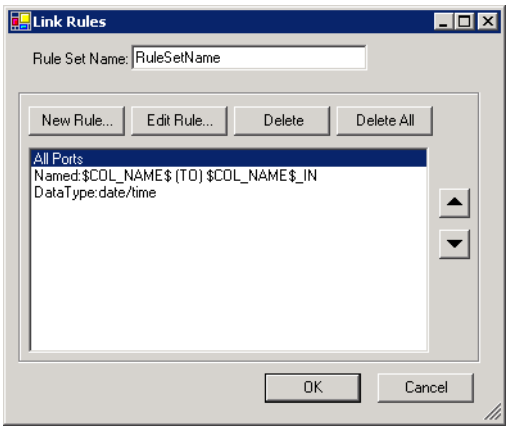
```
Named:<StartingPortName|ParameterName> (TO) <EndingPortName|ParameterName>
```

Parameter

The Parameter rule replaces the rules on a link with the rules included in the parameter you specify. When you create a Parameter rule, select the name of a parameter from the list of mapping template parameters.

The value of the parameter must be a set of rules that you want to use for the link.

The following figure shows how to configure rules with the same syntax on the Link Rules window:



The rules must be separated by semicolons (;).

Use a Parameter rule when the rules for a link between two transformation objects change from mapping to mapping. For example, you can create a parameter named $REPLICATION_RULES$ and provide the following value in the parameter file:

```
<PARAM NAME="$REPLICATION_RULES$"

VALUE="Primary Key;Named:$COL1$ (TO) $COL2$;DataType:date/time" />
```

Mapping Architect for Visio expands the Parameter rule to separate rules and evaluates the rules in the order they are listed.

You can provide different parameter values for the $REPLICATION_RULES$ for each mapping you create from the mapping template.

Limitations

You cannot include the following types of rules in a Parameter rule:

- **Parameter rule.** You cannot include a Parameter rule within a Parameter rule.
- **Rules with associated groups.** You cannot include rules for links that connect to multi-group transformations such as a Router transformation. For example, you cannot include the following rule in a Parameter rule:

  ```
  {GROUPA}Named:ITEM_DESC (TO) DESCRIPTION
  ```
- **Exclusion rule.** You cannot include a rule with the exclude option enabled. For example, you cannot include the following rule in a Parameter rule:

  ```
  EXLUDE Foreign Key
  ```

Pattern

The Pattern rule links all ports in a mapping object with names that match the specified starting port pattern to all ports in another mapping object with names that match the specified ending port pattern. Do not include spaces in port pattern names.

To create a Pattern rule, use a regular expression pattern. A regular expression is a specialized formula for matching text strings that follow a pattern. Create a Pattern rule only if you are familiar with the regular expression syntax.

The following table shows examples of regular expression patterns and the port names that match the patterns:

Regular Expression Pattern	Example Port Names that Match or Do Not Match the Pattern
Starting Port Pattern: _IN$ Ending Port Pattern: <blank>	GENDER_IN links to GENDER. ETHNIC_GROUP_IN links to ETHNIC_GROUP. EMPLOYEE_ID and EMPLOYEE_NAME do not match the starting port pattern.
Starting Port Pattern: ^EMP.*_.*$ Ending Port Pattern: $0	EMPLOYEE_ID links to EMPLOYEE_ID. EMPLOYEE_NAME links to EMPLOYEE_NAME. GENDER and ETHNIC_GROUP do not match the starting port pattern.
Starting Port Pattern: ^EMP.*_.*$ Ending Port Pattern: $0_IN	EMPLOYEE_ID links to EMPLOYEE_ID_IN. EMPLOYEE_NAME links to EMPLOYEE_NAME_IN. GENDER and ETHNIC_GROUP do not match the starting port pattern.
Starting Port Pattern: [\d]$ Ending Port Pattern: $0	EMPLOYEE_NAME1 links to EMPLOYEE_NAME1. GENDER1 links to GENDER1. EMPLOYEE_ID and ETHNIC_GROUP do not match the starting port pattern.
Starting Port Patterns: - [A-Za-z_][A-Za-z_0-9]* - [A-Za-z_0-9]+ - [A-Za-z_][A-Za-z_0-9]+ Ending Port Pattern: $0 _x	EMPLOYEE_ID links to EMPLOYEE_ID_x. EMPLOYEE_NAME links to EMPLOYEE_NAME_x. GENDER links to GENDER_x. ETHNIC_GROUP links to ETHNIC_GROUP_x.

Syntax

Use the following syntax when you include this rule in a Parameter rule:

```
Pattern:<StartingPortPatternString> (TO) <EndingPortPatternString>
```

Primary Key

The Primary Key rule links the primary key port in a mapping object to the primary key port with the same name in another mapping object.

For example, if you create a Primary Key rule on the link from a source definition to a source qualifier, Mapping Architect for Visio creates the primary key port in the source qualifier with the same name and properties as the source definition. Mapping Architect for Visio also creates links between ports of the same name.

The link between primary key and foreign key does not appear in Mapping Architect for Visio. However, the Import Mapping Template creates a mapping with these links.

Syntax

Use the following syntax when you include this rule in a Parameter rule:

```
Primary Key
```

CHAPTER 5

Using the Import Mapping Template Wizard

This chapter includes the following topics:

Using the Import Mapping Template Wizard Overview

Use the Import Mapping Template Wizard to create mappings based on published mapping templates you created in Mapping Architect for Visio. The Import Mapping Template Wizard uses the mapping template XML file to generate mappings.

Use the Import Mapping Template Wizard to perform the following tasks:

- Select the published mapping template.
- Specify parameter values.
- Select mappings to generate.
- Import mappings into the repository.

Before You Begin

To use the Import Mapping Template Wizard, verify the following prerequisites:

- **Source and target definition exist in the repository.** The Import Mapping Template Wizard uses an XML version of the source and target to create mappings from the mapping template. If the Import Mapping Template Wizard cannot access the source and target definitions, it does not create the mappings.
- **Source definition database name matches in mapping object properties and in the active repository folder.** The Import Mapping Template Wizard cannot create the mapping if the database name in the mapping source definition properties and the active repository folder are different.
- **Shortcut destination exists in a shared folder.** If you import a mapping template that includes a shortcut to a source or target definition, the destination source or target must exist in a shared folder.
- **Target definition is linked to at least one transformation.** The Import Mapping Template Wizard does not generate a mapping if the mapping template does not have at least one transformation linked to a target definition.
- **Mapplet or shortcut exists in the active repository folder.** If you import a mapping template that includes a mapplet or a shortcut, the mapplet or shortcut must exist in the active folder.

Export Source and Target Definitions

The Import Mapping Template Wizard looks for the source and target definitions in the following location:
`<PowerCenterInstallationDir>\client\MappingTemplate\tabledefs`

You can export source and target definitions to this folder in the following ways:

- Use the PowerCenter Export Wizard to export the source and target definitions.
- Use the Import Mapping Template Wizard to export source and target definitions based on parameter values.

The Import Mapping Template Wizard prompts you to export source or target definitions that are specified in parameters to the default location. If a source or target definition exists in the table definition location and you do not want to overwrite the current exported version, you can choose not to export the source or target definition. When you export the table definitions, ensure that you save the definitions in the correct location.

Note: If you use a Lookup transformation in a mapping template, you must manually export the lookup source. The Import Mapping Template Wizard does not export the source that the Lookup transformation references.

Step 1. Select the Mapping Template

In the first step of the Import Mapping Template Wizard, select the mapping template that you want to use.

To start the Import Mapping Template Wizard:

1. Open the Designer and connect to the repository.
2. Open the folder that contains the source and target objects you want to use in the imported mappings. Or, open the folder that contains a shortcut to the source and target objects.
3. Select Mappings > Import Mapping Template.

   The Import Mapping Template Wizard appears.
4. Click Browse. to the locate the mapping template XML file.

5. Click Next.

Step 2. Specify Parameter Values

This step displays all parameters for the selected mapping template. Specify the parameter values for each mapping you want to generate. Parameter values are case sensitive unless otherwise noted.

To configure parameter values for each mapping:

1. If you want to use a mapping template parameter file, click Use Existing to locate the parameter file.

   If the selected parameter file does not include the same parameters as the parameters configured in the mapping template, the following error appears:

   ```
   <filename> is not valid for current mapping template.
   ```

   The Import Mapping Template Wizard displays the mapping template parameters that you configured in the mapping template.
2. Click the Add Mapping button to add a mapping and specify values for the mapping template parameters.
3. Specify a mapping name and description, and enter values for all parameters used in the mapping template. The mapping name is not case sensitive. It cannot exceed 79 characters and cannot be a duplicate of a mapping in the active folder. It cannot include a tab, newline character, or the following special characters:

   ```
   , + " \ < > ; / * % ?
   ```
4. Enter values for each of the mapping template parameters. The way you enter values depends on how you configured the mapping template parameters in Mapping Architect for Visio. For example, you set parameters for the source definitions as $Source$, and selected Combo box for the Control field. The Import Mapping Template Wizard allows you to select a source from a list of available sources in the active folder. You can select flat file or relational source definitions.
5. Repeat steps 1 to 4 for each mapping you want to create.

   -or-

   Use the Copy and Paste buttons to replicate mapping configurations. Add and edit the parameter values for each new mapping you want to define.
6. Click Next.

   Note: Define at least one mapping before you proceed to the next step.

Step 3. Select Mappings to Generate

This step displays the list of mappings you configured. You can select the mappings that you want to generate.

If you select the Save parameter values for the selected mapping option, you can reuse the mapping template parameter file to create more mappings. The wizard saves the parameter values for each mapping you selected in the mapping template parameter file.

Note: Save the parameter values to a mapping template parameter file. If you do not save the parameter file and an error occurs when generating the mappings, you cannot retrieve the parameter value settings for each mapping you configured.

To select the mappings you want to generate:

1. Select the mappings you want to generate.
2. Optionally, click Save Parameter Values for the Selected Mappings to save the parameter values for the selected mappings. Click Browse to navigate to the folder where you want to save the parameter file.

   By default, the Publish Template function creates a mapping template parameter file in the same location as the mapping template XML file. You can choose to overwrite the existing file or create a new mapping template parameter file.
3. Optionally, click Create Single Mapping for Selected Mappings to create one mapping with multiple pipelines. Enter the name for the mapping. Default is the first mapping name in the list of mappings. Select this option if you do not want to create workflows for each mapping. For example, select this option if you want to create one connection to Oracle LogMiner.
4. Click Next.

   The wizard prompts you to export table definitions.
5. Click Yes.

   The Export Object dialog box appears and the table definitions are exported to the default location.
6. Click Close on the Export Objects dialog box.

Step 4. Import Mappings

This step displays the number of mappings that the wizard generated and the status of each mapping. You can generate workflows and sessions for the mappings you generated.

To import the mappings:

1. Review the list to verify that the wizard generated the correct number of mappings.

   By default, the option to create workflows and sessions for the mappings is selected.
2. To launch the Workflow Generation Wizard, click Next.

   -or-

   If you choose to disable the option to create workflows and sessions for the mappings, click Finish.

   The generated mappings appear in the mappings node of the selected repository folder.
3. Click Repository > Save.

Import Mapping Template Wizard Example

In this example, you use the mapping template XML file to create two mappings. To create multiple mappings from a mapping template, complete the following steps:

1. Select the mapping template.
2. Specify parameter values.

3. Select mappings to generate, and import the mappings.

Step 1. Select the Mapping Template

In the Designer, connect to the repository and select the folder that contains the source and target definitions required for the mappings you want to generate.

Then click Mappings > Import Mapping Template. After the Import Mappings Template Wizard appears, select the mapping template XML file that you want to use.

Step 2. Specify Parameters in the Mapping Template

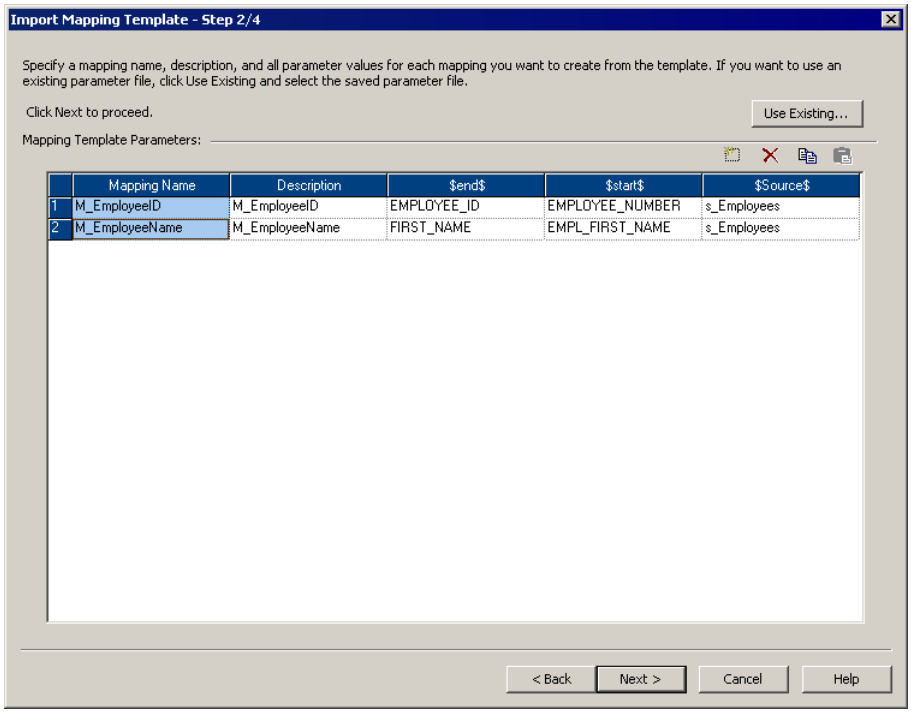
After you select the mapping template, use parameters to create two mappings. The parameters that you defined in the mapping template appear in the wizard.

You configured the following parameters in the mapping template:

- **end**. This parameter defines the target port.
- **$start$**. This parameter defines the source port that contains the data required in the target.
- **$Source$**. This parameter defines the source table.

Create two mappings with different values for the end and $start$ fields. Enter the same value for the $source$ field.

The following example shows the settings for the two mappings you want to create, one listing for M_EmployeeID and another for M_EmployeeName:



These settings create two mappings that use the same source table, but extract different data and load the data to different ports of the same target table.

The Source table list shows available sources in the following format:

```
DBDName:SouceTableName
```

Step 3. Generate and Import Mappings

The wizard displays all the mappings you configured. To generate both mappings, leave both mappings selected.

Before you import the mappings, the wizard asks you to export the table definitions. Since you did not manually export the source or target definitions to the tabledefs folder, you must export the definitions when prompted.

After you import the mappings, the mappings appear in the mapping folder of the selected repository folder.

Mapping Status Messages

This section describes common mapping status messages.

TableDefinitionNotFoundException Error: <Table Name>

Explanation:	Source or target definitions are not available in the tabledefs folder.
User Response:	Export the source or target definition to the tabledefs folder. Then, use the Import Mapping Template Wizard to create the mappings.

RuleNotSupportedException: Unknown Rule <Rule Name>

Explanation:	The parameter values specified for the rules are invalid.
User Response:	Review and fix the parameter values settings. Then, use the Import Mapping Template Wizard to create the mappings. If no message appears and no mapping is generated, source or target values may have been set incorrectly when you defined parameter values. Review and fix the parameter values settings. Then, use the Import Mapping Template Wizard to create the mappings.

Value of JAVA_HOME Environment Variable in Configurations.xml file is not set.

Explanation:	You did not set the path for the JAVA_HOME environment variable in the Configurations.xml file.
User Response:	Set the path for the JAVA_HOME environment variable. By default, the JAVA_HOME environment variable is in following directory: <PowerCenter installation directory>/java

Value of MAPFWK_HOME Environment Variable in Configurations.xml file is not set.

Explanation:	You did not set the path for the MAPFWK_HOME environment variable in the configurations.xml file.
User Response:	Set the path for the MAPFWK_HOME environment variable. By default, the MAPFWK_HOME environment variable is in following directory: <PowerCenter installation directory>/MappingSDK

CHAPTER 6

Using the mapgen Command Line Program

This chapter includes the following topics:

Using the mapgen Command Line Program Overview

Use *mapgen* to create mappings based on published mapping templates you created in Mapping Architect for Visio. *mapgen* uses the mapping template XML file to generate mappings.

To create mappings from a mapping template, complete the following steps:

1. **Export the sources and targets, shortcuts, and lookup sources.** In the Designer or Repository Manager, export the files for objects in the mapping template that *mapgen* requires to generate the mapping.
2. **Edit the parameter file for the mapping template.** The Publish Template function in Mapping Architect for Visio creates a mapping template parameter file. The parameter file must provide the values for all parameters used in the mapping template. Define each mapping you want to create in the parameter file.
3. **Create mapping file.** *mapgen* creates a mapping file that you can import into the PowerCenter repository. The mapping file contains an XML representation of the mappings defined in the mapping template and parameter file.
4. **Import the mapping file into the repository.** After *mapgen* creates the mapping file, go to the Designer or Repository Manager to import the mapping file into the repository.

mapgen File Requirements

mapgen requires the following files:

- **Mapping template drawing file.** After you create the mapping template drawing file in Mapping Architect for Visio, save it as a Visio drawing file with a .vsd extension for Microsoft Visio 2010 and previous versions. You can save the file with a .vsdx extension for Microsoft Visio 2016. *mapgen* reads the Visio drawing file to determine the transformations and link rules for the mappings it creates based on the mapping template.
- **Mapping template parameter file.** When you publish the mapping template, Mapping Architect for Visio creates a mapping template parameter file. Edit the parameter file to provide the values for the mapping parameters. *mapgen* reads the mapping template parameter file to determine the values of the parameters used in the mapping template.
- **Source and target definition files.** The mapping template does not include the definitions for sources and targets. Export the source and target definitions from PowerCenter. *mapgen* reads the source and target definition files to determine the source and target table properties and columns to use in the mappings it creates based on the mapping template.

Source, Target, and Shortcut Files

When *mapgen* evaluates the rules in the mapping template, it requires the names and properties of the columns available in the source and target tables. *mapgen* uses the column names and properties to determine the ports in the mapping transformations that conform to the rules. *mapgen* also requires the name, object type, repository name, and folder name of the shortcuts to sources, targets, and transformations used in the mapping template. Export the sources, targets, and shortcuts to make them available to *mapgen*.

Export the following objects to the tabledefs folder:

- source definition
- target definition
- shortcut to source definition
- shortcut to target definition
- shortcut to transformation
- lookup source for Lookup transformation

If you do not export shortcuts to targets and transformations, *mapgen* creates the targets or transformations based on the rules for the links to the target or transformation.

Use the following rules and guidelines when you export the source and target definitions and shortcuts to sources, targets, and transformations:

- *mapgen* looks for all the files in one folder. By default, it looks for the files in the Mapping Architect for Visio table definition folder:

  `\<PowerCenterInstallationDir>\Client\MappingTemplate\tabledefs`

  If you export the files into a different folder, specify the folder path name when you run *mapgen*.
- *mapgen* requires all XML files in the folder to be valid PowerCenter XML files that comply with powrmart.dtd. *mapgen* reads all XML files in the directory. To ensure that *mapgen* can read all XML files in the folder, keep only valid PowerCenter XML files in the table definition folder.

- Since *mapgen* reads all files in the table definition folder, you can export multiple source definitions to one XML file. For example, the folder can contain one XML file that describes five sources or five XML files that each describe one source definition. Similarly, you can export shortcuts to multiple transformations to one XML file.
- *mapgen* uses the first definition it finds in the table definition folder. For example, if you have ASources.xml that has a source definition for EMPLOYEE and BSources.xml that also has a source definition for EMPLOYEE, *mapgen* uses the first EMPLOYEE source definition it finds. If the source definitions are from different folders or have different columns and properties, *mapgen* can use the wrong table definition for the mapping.

  Create separate folders for table definitions with the same names but different properties to ensure that *mapgen* uses the correct table definition for the mapping. Then supply the appropriate folder name when you run *mapgen*.

Manually Updating Mapping Template Parameters

When you publish a mapping template, Mapping Architect for Visio creates a mapping template XML file and a mapping template parameter file. The mapping template parameter file contains the parameters set up in the mapping template and is located in the same folder as the mapping template file.

Modify the mapping template parameter file to provide the values for the mapping template parameters. If you want to create more than one mapping from the mapping template, modify the parameter file so that it contains as many mapping definitions as you require. Then provide the parameters values for each mapping.

The mapping template parameter file has an associated Document Type Definition (DTD) file called parameters.dtd. When you create the mappings based on the mapping template, the parameter file is validated against parameter.dtd. When you modify the parameter file, make sure that the XML file conforms to the structure of parameter.dtd.

Parameter File Requirements

The following text shows a sample mapping template parameter file:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE PARAMETERS SYSTEM "parameters.dtd">
<PARAMETERS REPOSITORY_NAME="" REPOSITORY_VERSION="" REPOSITORY_CODEPAGE=""
REPOSITORY_DATABASETYPE="">
  <MAPPING NAME="" FOLDER_NAME="" DESCRIPTION="">
    <PARAM NAME="$TGT$" VALUE="" />
    <PARAM NAME="$SRC$" VALUE="" />
    <PARAM NAME="$DBD_NAME$" VALUE="" />
    <PARAM NAME="$DB_TYPE$" VALUE="" />
</MAPPING>
</PARAMETERS>
```

The mapping template parameter file includes the following information:

- **Parameter properties.** Contains the repository name. It is required for each parameter file.
- **Mapping properties.** Contains the name, description, and location of the mapping to be created. It is required for each mapping in the parameter file.
- **Parameter values.** Values required for each mapping in the parameter file.

Parameter Properties

The parameter file requires the name of the repository where you plan to import the mappings defined in the parameter file. You set the REPOSITORY_NAME property once for each parameter file. All mappings defined in the parameter file will have the same repository name.

Mapping Properties

Set the following properties for each mapping in the parameter file:

- **NAME.** Name of the mapping.
- **FOLDER_NAME.** Name of the repository folder where you plan to import the mapping.
- **DESCRIPTION.** Description of the mapping.

All mapping properties are required.

Parameter Values

Use mapping template parameters to set different values for properties, expressions, port names, and other mapping elements that may vary for each mapping. Set the value for each mapping template parameter that you define.

Each PARAM element defines a parameter name-value pair. Set the parameter name in the mapping template. Set the value for the parameter in the parameter file.

Modifying the Parameter File

The mapping template parameter file is an XML file. Modify it in the same way you modify HTML and other text files.

When you install PowerCenter, the DTD for the mapping template parameter file is installed. To view the DTD for the parameter file, locate the *parameter.dtd* file in the following directory:

```
<PowerCenterInstallationDir>\client\MappingTemplate\dtd
```

The number of MAPPING elements in the parameter file equals the number of mappings you want to create from the mapping template.

To modify the parameter file:

1. Locate the mapping template parameter file for the mapping template you want to use.
2. Open the mapping template parameter file with a text editor.
3. Search for the following string in the file:

   ```
   <PARAMETERS REPOSITORY_NAME=
   ```

4. Set the REPOSITORY_NAME property to the name of the repository where you plan to import the mappings.
5. Search for the following string in the file:

   ```
   <MAPPING NAME=
   ```

6. Set the following required mapping properties:
   - Name of the mapping
   - Folder name in the repository where you plan to import the mappings
   - Description of the mapping

7. Search for the following string in the file:

   ```
   <PARAM NAME=
   ```

8. Set the mapping template parameters to values appropriate for the mapping.
9. If you want to create more than one mapping from the same mapping template, add as many copies of the MAPPING element to the parameter file as you require and set the parameters to the values appropriate for each mapping.
10. Save the parameter file and close it.

Running the mapgen Command Line Program

Use the *mapgen* command line program to create the mappings from a mapping template.

mapgen uses the following syntax:

```
mapgen
<-t> VisioDrawingFile
<-p> ParameterFile
<-o> MappingFile
[-d] TableDefinitionDir
```

The following table describes the *mapgen* options and arguments:

Option	Argument	Required/ Optional	Description
-t	VisioDrawingFile	Required	Name of the Visio drawing file for the mapping template. The file has a .vsd extension for Microsoft Visio 2010 and previous versions. You can save the file with a .vsdx extension for Microsoft Visio 2016. To create the file, save the mapping template in Mapping Architect for Visio as a Visio drawing file. If the file is not in the current working folder, add a path name in front of the file name.
-p	ParameterFile	Required	Name of the mapping template parameter file. Mapping Architect for Visio automatically creates the parameter file when you publish the mapping template. The filename is <mapping template name>_param.xml. If the file is not in the current working folder, add a path name in front of the file name.
-o	MappingFile	Required	Name of the PowerCenter mapping file used to create mappings from the mapping template. The file has a .xml extension. If you do not want to create the file in the current working folder, add a path name in front of the file name.
-d	TableDefinitionDir	Conditional	Name of the folder where the source and target definition files are located. Required if the files are not located in the default folder. By default, the source and target definition files are stored in the following location: `\< PowerCenterInstallationDir >\Clien t\ MappingTemplate\tabledefs`

Before you run *mapgen*, verify that you have all the files required by the program.

To run the mapgen program:

1. Close the referenced Visio drawing file.

   If the .vsd or .vsdx file is open when you run *mapgen*, *mapgen* generates a warning that the file is open for modification.

2. In the Designer or Repository Manager, export the source and target definitions for the sources and targets you want to use in the mappings to be created by the *mapgen* program.

   By default, *mapgen* looks for the source and target definition files in the following folder:

   ```
   <PowerCenterInstallationDir>\client\MappingTemplate\tabledefs
   ```

   If you export the files to another folder, provide the folder name when you run the *mapgen* program.

3. Open a command line window on the machine where Mapping Architect for Visio is installed.

4. Go to the Mapping Architect for Visio folder.

   By default, Mapping Architect for Visio is installed in the following location:

   ```
   <PowerCenterInstallationDir>\client\MappingTemplate\
   ```

5. Run mapgen.exe.

   After you generate the mapping file, import the file into the repository to view all the generated mappings.

mapgen Command Line Program Example

In this example, you use *mapgen* to generate two mappings from a mapping template. To generate PowerCenter mappings and import them into the repository, complete the following steps:

1. Export source and target definitions to the tabledefs folder.
2. Edit the parameter file.
3. Run the *mapgen* command line program.
4. Import the mapping file into the repository.

Step 1. Export Source and Target Definitions

Open the Designer or Repository Manager and export the source and target definitions that you plan to use in the mapping template.

In this example, you export s_Employees and t_Employees to the following default location:

```
\<PowerCenterInstallationDir>\Client\MappingTemplate\tabledefs
```

Step 2. Edit the Parameter File

When you publish the mapping template, Mapping Architect for Visio creates a parameter file in the same location as the Mapping Template XML file. Locate the parameter file and edit it to include the parameter values required for the mappings you want to generate.

In the mapping template you specified the following parameters:

- **end**. This parameter defines the target port.
- **$start$**. This parameter defines the source port that contains the data required in the target.
- **$Source$**. This parameter defines the source table.

Update the parameter file to create two mapping. Both mappings use the same source, but they extract data different source ports and load data to different ports of the same target. The following code appears in the completed parameter file:

```
<?xml version='1.0'?>
<!DOCTYPE PARAMETERS SYSTEM "parameters.dtd">
<PARAMETERS REPOSITORY_NAME="HR">
  <MAPPING NAME="M_EmployeeID" FOLDER_NAME="CRM" DESCRIPTION="M_EmployeeID">
    <PARAM NAME="$end$" VALUE="EMPLOYEE_ID" />
    <PARAM NAME="$start$" VALUE="EMPLOYEE_NUMBER" />
    <PARAM NAME="$Source$" VALUE="s_Employees" />
  </MAPPING>
<MAPPING NAME="M_EmployeeName" FOLDER_NAME="CRM" DESCRIPTION="M_EmployeeName">
    <PARAM NAME="$end$" VALUE="FIRST_NAME" />
    <PARAM NAME="$start$" VALUE="EMPL_FIRST_NAME" />
    <PARAM NAME="$Source$" VALUE="s_Employees" />
  </MAPPING>
</PARAMETERS>
```

Step 3. Run the mapgen Command Line Program

Before you run *mapgen*, close and save the mapping template parameter and drawing files.

You have the following mapping template files in the folder:

```
<PowerCenterInstallationDir>\client\MappingTemplate\
```

- mt_employee.vsd or mt_employee.vsdx
- mt_employee.param.xml

Go to the command line and perform the following steps:

1. Change the directory to the following location:

   ```
   < PowerCenter InstallationDir >\client\MappingTemplate\
   ```

2. Enter the following command for Microsoft Visio 2010 and previous versions:

   ```
   mapgen -t mt_employees.vsd -p mt_employees_param.xml -o output_employees.xml
   ```

   Enter the following command for Microsoft Visio 2016:

   ```
   mapgen -t mt_employees.vsdx -p mt_employees_param.xml -o output_employees.xml
   ```

mapgen creates the mapping file named output_employees.xml. Use this file to import the new mappings into the repository.

Step 4. Import the Mappings into the Repository

In the Designer or Repository Manager, connect to the repository. Select Repository > Import Objects. The Import Wizard appears.

To import the mappings, select the mapping file output_employees.xml that *mapgen* created. After you complete the steps in the Import Wizard, the mappings appear in the repository.

CHAPTER 7

Using Informatica Mapping Templates

This chapter includes the following topics:

Using Informatica Mapping Templates Overview

Informatica mapping templates are predefined mapping templates that cover common data warehousing patterns, such as slowly changing dimensions and incremental load. These templates provide solution for the most common issues in the data warehousing designs. You can use the predefined mapping templates to document methods to map or process data.

Informatica provides predefined mappings for the following types of templates:

- **Type 1 Slowly Changing Dimensions**. Loads a slowly changing dimensions table by inserting new dimensions and overwriting existing dimensions.
- **Type 2 Slowly Changing Dimensions.** Loads a slowly changing dimensions table with new and changed dimensions. There are three types of type 2 slowly changing dimensions.
- **Type 3 Slowly Changing Dimensions**. Loads a slowly changing dimensions table by inserting new dimensions and updating values in existing dimensions.
- **Remove Duplicates**. Duplicate records are frequently found in source data. The Remove Duplicates Mapping template helps you remove duplicate data from source tables.
- **Incremental Load**. You can use the Incremental Load template when you want to identify and capture the data that has been added, changed, or deleted in a database table.

  For more information on creating a mapping from one of these mapping templates, see *PowerCenter Designer Guide*.

Type 1 Slowly Changing Dimensions Template

The Type 1 Slowly Changing Dimensions Template filters source rows based on user-defined comparisons and inserts only those found to be new dimensions to the target. Rows containing changes to the existing dimensions are updated in the target by overwriting the existing dimension. In the Type 1 Dimension mapping, all rows contain current dimension data.

Use the Type 1 Slowly Changing Dimensions Template to update a slowly changing dimensions table when you do not need to keep any previous versions of dimensions in the table.

For example, you might have a site dimension table with store code, location, and overhead that you update after the company opens a new store. This dimension is used for sales and overhead calculations. Since you do not need to know the previous address of the same store or the store overhead from the previous year, you do not need previous dimension data in the table. With the Type 1 Dimension mapping, you can keep current data without a historical log.

Parameters

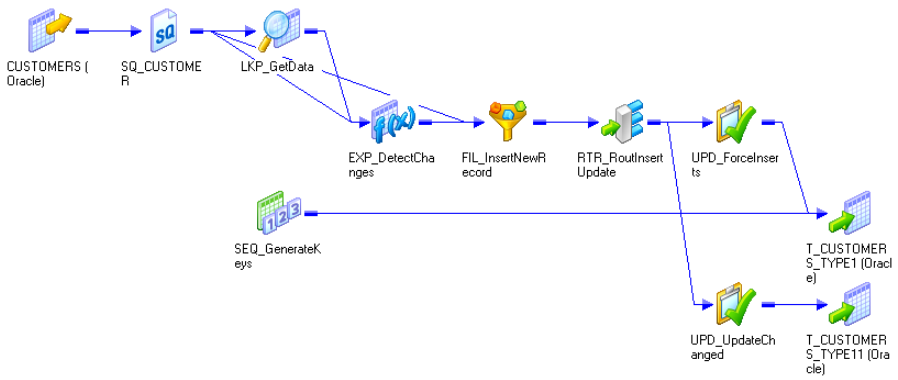
When you use the Type 1 Slowly Changing Dimensions Wizard, you must configure the parameters.

The following table describes the parameters for the Type 1 Slowly Changing Dimensions Wizard:

Parameter	Name	Description
$CREATE_TGT$	Create Target Table	Creates a target table instead of using an existing one. The target database type will be the same as the source database type.
$INCR_COND$	Incremental Extract Condition	Loads all rows from the database source table. If this is not specified, all rows will be loaded.
TGT	Target Table	Selects a target table from the target tables in the current working folder.
SRC	Source Table	All available source definitions of the current working directory.
$CKEY$	Comparison Key Fields	Set of fields that identify a changed row between the source and target tables.
$LKEY$	Logical Key Fields	Columns of the source table that identify a particular entity.
DBD	Database Types	Database type for the new target.
$SKEY$	Surrogate Key	Column in the target table that can act as a primary key.
$DICT$	Set Association	Association between the source table columns and target table columns.

Understanding the Mapping

The following figure shows a mapping that the Slowly Changing Dimensions Wizard creates when you select the Type 1 Dimension option:



The Type 1 Dimension mapping uses Lookup and Expression transformations to compare source data against existing target data. When you step through the Slowly Changing Dimensions Wizard, you enter the lookup conditions and source columns that you want the Integration Service to compare against the existing target.

For each source row without a matching primary key in the target, the Expression transformation marks the new row. For each source row with a matching primary key in the target, the Expression compares user-defined source and target columns. If those columns do not match, the Expression marks the row changed. The mapping then splits into two separate data flows.

The first data flow uses the Router transformation, RTR_RoutInsertUpdate, to pass only new records to the Updates Strategy transformation, UPP_ForceInserts. The Updates Strategy transformation, UPD_ForceInserts, inserts new rows into the target, and a Sequence Generator, SEQ_GenerateKeys, creates a primary key for each row.

In the second data flow, the Router transformation, RTR_RoutInsertUpdate, passes changed rows to the Update Strategy transformation, UPD_UpdateChanged. The Update Strategy transformation, UPD_UpdateChanged, replaces existing rows in the target with the updated source rows.

You can create the following transformations by using the Type 1 Slowly Changing Dimensions Wizard:

- Lookup transformations
- Filter transformations
- Update Strategy transformations

Type 2 Slowly Changing Dimensions Template

In the Type 2 Dimension mapping, the slowly changing dimensions table is updated with new and changed dimensions. There are three types of Type 2 Slowly Changing Dimensions:

- **Version Data Mapping**. The Type 2 Dimension/Version Data mapping filters source rows based on user-defined comparisons and inserts both new and changed dimensions into the target. Changes are tracked in the target table by versioning the primary key and creating a version number for each dimension in the table. In the Type 2 Dimension/Version Data target, the current version of a dimension has the highest version number and the highest incremented primary key of the dimension.

  Use the Type 2 Dimension/Version Data mapping to update a slowly changing dimensions table when you want to keep a full history of dimension data in the table. Version numbers and versioned primary keys track the order of changes to each dimension.

- **Flag Current Mapping**. The Type 2 Dimension/Flag Current mapping filters source rows based on user-defined comparisons and inserts both new and changed dimensions into the target. Changes are tracked in the target table by flagging the current version of each dimension and versioning the primary key. In the Type 2 Dimension/Flag Current target, the current version of a dimension has a current flag set to 1 and the highest incremented primary key.

  Use the Type 2 Dimension/Flag Current mapping to update a slowly changing dimensions table when you want to keep a full history of dimension data in the table, with the most current data flagged. Versioned primary keys track the order of changes to each dimension.

- **Effective Date Range Mapping**. The Type 2 Dimension/Effective Date Range mapping filters source rows based on user-defined comparisons and inserts both new and changed dimensions into the target. Changes are tracked in the target table by maintaining an effective date range for each version of each dimension in the target. In the Type 2 Dimension/Effective Date Range target, the current version of a dimension has a start date with no corresponding end date. Use the Type 2 Dimension/Effective Date Range mapping to update a slowly changing dimensions table when you want to keep a full history of dimension data in the table. An effective date range tracks the chronological history of changes for each dimension.

For example, you might have a dimension table with product information, such as product name, product ID, year, and product price. When the price of the product changes, a new row is added to the table with latest price information and the previous row is retained by adding a new column with a version\date\flag mapping. When the product price changes continuously, the complete history of the changes are stored.

Parameters

In addition to the parameters described in "Parameters" on page 62, the Type 2 Slowly Changing Dimensions wizard uses additional parameters.

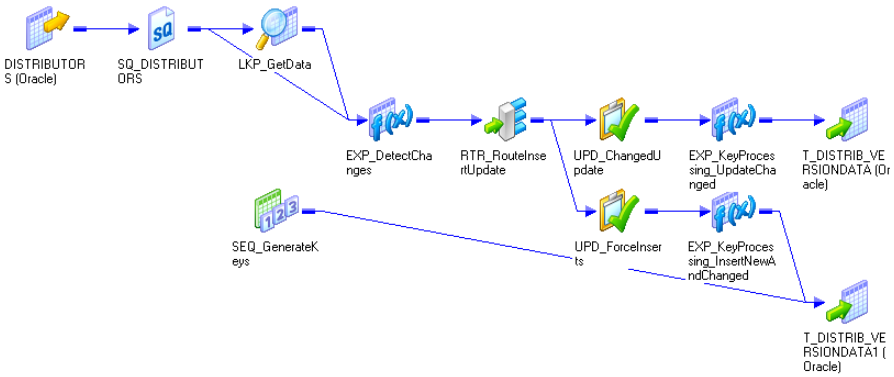
The following table describes the additional parameters for the Type 2 Slowly Changing Dimensions wizard:

Parameter	Name	Description
$FKEY$	Current Flag	Distinguishes old and new rows. New row has value equal to 1 and old row has value equal to 0.
$SDATE$	Effective Start Date	Column in the target table to store the start date of the period of comparison.
$EDATE$	Effective End Date	Column in the target table to store the end date of the period of comparison.

Parameter	Name	Description
$DDATE$	Default End Date	End date value. Use the database-specific date function to specify the date.
$VKEY$	Version Key Field	Tracks the history of all changes. Every time a row is modified, the version number value is incremented by one.

Understanding the Mapping

The following figure shows a mapping that the Slowly Changing Dimensions Wizard creates when you select the Type 2 Dimension or Version Data option:



The Type 2 Dimension/Version Data mapping uses Lookup and Expression transformations to compare source data against the existing target data. When you step through the Slowly Changing Dimensions Wizard, you enter the lookup conditions and source columns that you want the Integration Service to compare against the existing target.

For each source row without a matching primary key in the target, the Expression transformation marks the new row. For each source row with a matching primary key in the target, the Expression compares user-defined source and target columns. If those columns do not match, the Expression marks the row changed. The mapping then splits into two data flows.

The first data flow uses the Router transformation, RTR_RouteInsertUpdate, to pass only new rows to the Update Strategy transformation, UPD_ForceInserts. The Update Strategy transformation, UPD_ForceInserts, inserts new rows to the target. A Sequence Generator, SQ_GenerateKeys, creates a primary key for each row. The Expression transformation, EXP_KeyProcessing_InsertNewAndChanged, increases the increment between keys by 1,000 and creates a version number of 0 for each new row.

In the second data flow, the Router transformation, RTR_RouteInsertUpdate, passes only changed rows to pass to the Update Strategy transformation, UPD_ChangedUpdate. The Update Strategy transformation, UPD_ChangedUpdate, inserts changed rows to the target. The Expression transformation, EXP_KeyProcessing_UpdateChanged, increments both the key and the version number by one.

You can create the following transformations by using the Type 2 Slowly Changing Dimensions Wizard:

- Sequence Generator transformations
- Lookup transformations
- Router transformations
- Expression transformations
- Update Strategy transformations

Type 3 Slowly Changing Dimensions Template

The Type 3 Dimension mapping filters source rows based on user-defined comparisons and inserts only those found to be new dimensions to the target. Rows containing changes to the existing dimensions are updated in the target. When updating an existing dimension, the Integration Service saves existing data in different columns of the same row and replaces the existing data with the updates. The Integration Service optionally enters the system date as a timestamp for each row it inserts or updates. In the Type 3 Dimension target, each dimension contains current dimension data.

Use the Type 3 Dimension mapping to update a slowly changing dimensions table when you want to keep only current and previous versions of column data in the table. Both versions of the specified column or columns are saved in the same row.

When you use this option, the Designer creates additional fields in the target:

- **PM_PREV_ColumnName**. The Designer generates a previous column corresponding to each column for which you want historical data. The Integration Service keeps the previous version of dimension data in these columns.
- **PM_PRIMARYKEY**. The Integration Service generates a primary key for each row written to the target.
- **PM_EFFECT_DATE**. This is an optional field. The Integration Service uses the system date to indicate when it creates or updates a dimension.

For example, you might have a site dimension table with store code, location, and overhead that you update after the company changes the location of a store. This dimension is used for sales and overhead calculations. In this case, you need only the more recent two versions of the location information to track the changes and do the calculations. You do not need to know the complete history of all previous locations. With the Type 3 Dimension mapping, you can keep current data and the previous data without having a complete history.

Parameters

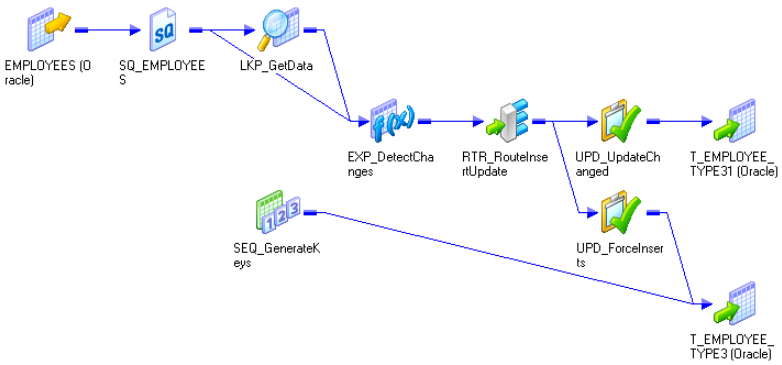
In addition to the parameters described in "Parameters" on page 62, the Type 3 Slowly Changing Dimensions Wizard uses an additional parameter.

The following table describes the additional parameter for the Type 3 Slowly Changing Dimensions wizard:

Parameter	Name	Description
$EFDATE$	Effective Date Field	Date from which the parameter is effective.

Understanding the Mapping

The following figure shows a mapping that the Slowly Changing Dimensions Wizard creates when you select the Type 3 Dimension option:



The Type 3 Dimension mapping uses Lookup and Expression transformations to compare source data against existing target data. When you step through the Slowly Changing Dimensions Wizard, you enter the lookup conditions and source columns that you want the Integration Service to compare against the existing target. The Designer creates additional columns for the change columns to hold historic data.

For each source row without a matching primary key in the target, the Expression transformation marks the new row. For each source row with a matching primary key in the target, the Expression compares user-defined source and target columns. If those columns do not match, the Expression marks the row changed. The mapping then splits into two data flows.

The first data flow uses the Router transformation, RTR_RouteInsertUpdate, to pass only new rows to the UPD_ForceInserts Update Strategy transformation. UPD_ForceInserts inserts new rows to the target. A Sequence Generator, SEQ_GenerateKeys, creates a primary key for each row. If you select the Effective Date option in the mapping wizard, the Designer also creates a new field, PM_EFFECT_DATE, in the Expression Transformation, EXP_DetectChanges. The Integration Service uses the system date to indicate when it creates a new row.

In the second data flow, the Router transformation, RTR_RouteInsertUpdate, allows only the changed rows to pass to the Update Strategy transformation, UPD_UpdateChanged. UPD_UpdateChanged inserts changed rows to the target. If you select the Effective Date option in the mapping wizard, the Designer also creates a new field, PM_EFFECT_DATE, in the Expression Transformation, EXP_DetectChanges. The Integration Service uses the system date to indicate when it updates a row.

You can create the following transformations by using the Type 3 Slowly Changing Dimensions Wizard:

- Lookup transformations
- Filter transformations
- Expression transformations
- Router transformations
- Update Strategy transformations

Remove Duplicates Template

Duplicate records are frequently found in source data. The Remove Duplicates Mapping template helps you to remove duplicate data from source tables. This template helps you to maintain only one version of the data in the target database.

For example, your organization acquires another organization and both the organizations have many common customers. When you update the database, you need to remove the duplicate information, based on attributes like name and address of the common customers. You can use the remove duplicates template to remove duplicate information of the common customers and keep the database updated.

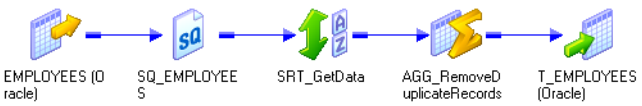
Parameters

The following table describes the parameters for the Remove Duplicates Mapping template:

Parameter	Name	Description
TGT	Target Table	Selects a target table from the target tables in the current working folder.
SRC	Source Table	All available source definitions of the current working directory.
$SKEY$	Surrogate Key	Surrogate key is the primary key.
$DICT$	Set Association	Association between the source table columns and target table columns.

Understanding the Mapping

The following figure shows a mapping that the Remove Duplicates Mapping Templates Wizard creates when you select the Remove Duplicates option:



The Remove Duplicates Dimension mapping uses the Sorter transformation, SRT_GetData, to sort the data. The Aggregator transformation, AGG_RemoveDuplicateRecords, removes the duplicate records and passes only one version of the data to the target table.

You can create the following transformations by using the Remove Duplicates Mapping template:

- Sorter transformations
- Aggregator transformations

Incremental Load Template

You can use the Incremental Load template when you want to identify and capture the data that has been added, changed, or deleted in a database table. The Incremental Load template can be used in the following ways:

- **Use Last Refresh Time.** You can update the target database by selecting all the rows in a table where the date in the create or modified date fields equates SYSDATE-1.
- **Use Session Time Stamp**. You can update the target database by loading the latest data based on the session time stamp.

For example, Karen works in a sales environment where the target database is updated at the end of each day to include the whole history of transactions. The database is updated by running a workflow. Karen uses the Incremental Load template to update the target database. This template ensures that only the changed data, and not the entire data, is loaded into the target database.

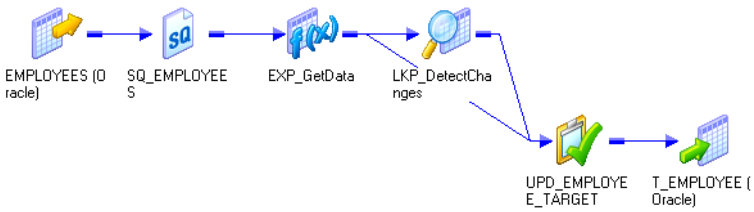
Parameters

The following table describes the parameters for the Incremental Load Template Wizard:

Parameter	Name	Description
TGT	Target Table	Selects a target table from the target table list to use an existing target table.
SRC	Source Table	Selects a source table from the source table list.
$LKEY$	Logical Key Fields	Selects the columns in the source table to lookup data in the target table.
$UPDATE_TS$	Source Timestamp Field	Time at which the row in the table is updated.
$DICT$	Set Association	Association between the source table columns and target table columns.

Understanding the Mapping

The following figure shows a mapping that the Incremental Load Wizard creates when you select the Incremental Load option:



The Incremental Load mapping uses Lookup and Expression transformations to compare source data against existing target data. When you step through the Slowly Changing Dimensions Wizard, you enter the lookup conditions and source columns that you want the Integration Service to compare against the existing target. The Designer creates additional columns for the change columns to hold historic data.

For each source row without a matching primary key in the target, the Expression transformation marks the row new. For each source row with a matching primary key in the target, the Expression compares user-defined source and target columns. If those columns do not match, the Expression marks the row changed. The Update Strategy transformation, UPD_EMPLOYEE_TARGET, passes only the new rows to the target.

You can create the following transformations using the Incremental load templates wizard:

- Lookup transformations
- Expression transformations
- Update Strategy transformations

APPENDIX A

Glossary

drawing window

Work area for the mapping template. Mapping Architect for Visio initially displays an empty drawing window.

Import Mapping Template Wizard

A wizard in the PowerCenter Client that generates mappings from mapping template files and imports them into the repository.

Informatica stencil

A template that includes shapes that represent PowerCenter mapping objects such as source definitions, target definitions, link, and transformations.

Informatica toolbar

A toolbar in Microsoft Visio that displays buttons for tasks you can perform on a mapping template, such as create a template from a mapping XML file, validate a template, publish a template, and declare mapping parameters and variables.

link

A mapping object that connects sources, targets, and transformations, and specifies the rules for data movement.

mapgen command line program

A command line program that generates an XML file from mapping template files. Import the XML file into PowerCenter to create the mappings.

mapping template

A drawing in Visio that represents a mapping. Use rules and parameters in a mapping template to specify the transformation logic for a PowerCenter mapping.

mapping template drawing file

A file that Mapping Architect for Visio generates when you save a mapping template. The file name is in the format [template name].vsd or [template name].vsdx based on the Microsoft Visio version.

mapping template files

Files that Mapping Architect for Visio generates when you save or publish a mapping template. Mapping template files include mapping template drawing files, mapping template XML files, and mapping template parameter files.

mapping template parameter file

A file that Mapping Architect for Visio generates when you publish a mapping template. Mapping Architect for Visio also generates mapping template XML file. Use the mapping template parameter file to define and save parameter values. The filename is in the format [template name]_param.xml.

mapping template XML file

A file that Mapping Architect for Visio generates when you publish a mapping template, or a mapping template drawing file. Mapping Architect for Visio also generates mapping template parameter file. The file name is in the format [template name].xml.

mapping XML file

A file that the Designer generates when you export a mapping. In Mapping Architect for Visio, you can create a mapping template from the mapping XML file.

PowerCenter mapping template

A predefined mapping template in the PowerCenter Client that covers a common data warehousing pattern, for example, slowly changing dimensions.

rule set

Set of rules that indicates how data is moved from one mapping object to the next. Every link in a mapping template must have an associated rule set. A rule set can have one or more rules.

INDEX

M

N

O

P

R

S

T

U

V

W

Z