How-To Library

Informatica

Creating a Parameter File for Informatica Developer

Abstract

A parameter file is an .xml file that lists user-defined parameters and their assigned values. Parameter files provide the flexibility to change parameter values each time that you run a mapping or a workflow. Generate a parameter file based on a mapping or workflow using the Developer tool or the command line. Edit the contents of the file to create unique parameter files that you can use for different runs of the same mapping or workflow. This article provides a description of the elements in a parameter file as well as the steps to generate and edit a parameter file.

Supported Versions

- Informatica 10.x

Table of Contents

Overview

A parameter file is an .xml file that lists user-defined parameters and their assigned values. You can use a parameter file when you run a mapping or a workflow, or when you deploy an application that contains mapping or workflow objects to flexibly change the parameter values that the Data Integration Service applies.

The parameter values in a parameter file define properties for the mapping or workflow. The Data Integration Service reads the parameter values in the parameter file at the start of the mapping or workflow run, resolves the parameters, and applies the parameter values.

The following steps describe how to create a parameter file:

1. Create parameters in the Developer tool.
2. Generate a parameter file. To generate a parameter file, you can export the parameters from the Developer tool, or you can list the parameters in the command line and generate the parameter file from the list of values.

3. Edit the parameter values in the parameter file or edit the parameter file structure to create unique sets of parameters in each parameter file.

After you create the parameter file, you can define parameters in the following ways:

- Define parameters for multiple mappings or workflows in a single parameter file.
- Define parameters for the same mapping or workflow in different parameter files and use a different parameter file each time that you run the mapping or workflow to use different parameter values.
- Define parameter values for mappings or workflows within an application. Define multiple applications to set different parameter values for a mapping or workflow depending on which application is deployed.

Use Case - Creating Multiple Parameter Files

You connect to a database that stores data on employees at your organization. You want to run a mapping that ingests data on the employees to different targets. Each target should contain employee data based on the employee's department and the salary that they earn.

To filter the data to separate targets, you want to configure the following combinations of mapping parameters:

Department Parameter	Salary Parameter
Support	$Salary < 50000
Support	50000 < $Salary
Development	$Salary < 50000
Development	50000 < $Salary

In the Developer tool, you configure the mapping. The mapping contains a Read transformation, two Filter transformations, and a Write transformation. In the first Filter transformation, you configure a parameter to filter the data by department. In the second Filter transformation, you configure a parameter to filter the data by salary.

You then configure the parameter files. Each parameter file contains one of the department and salary parameter combinations. The following sample files describe the parameter combinations that you can configure in each parameter file:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root xmlns="http://www.informatica.com/Parameterization/1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema" version="2.0">
    <!-- File 1    -->
    <project name="MyProject">
        <mapping name="MyMapping">
            <parameter name="Department">Support</parameter>
            <parameter name="Salary">$Salary<50000</parameter>
        </mapping>
    </project>
</root>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root xmlns="http://www.informatica.com/Parameterization/1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema" version="2.0">
    <!-- File 2    -->
    <project name="MyProject">
        <mapping name="MyMapping">
            <parameter name="Department">Support</parameter>
            <parameter name="Salary">50000<$Salary</parameter>
        </mapping>
```

```
    </project>
</root>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root xmlns="http://www.informatica.com/Parameterization/1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema" version="2.0">
    <!-- File 3    -->
    <project name="MyProject">
        <mapping name="MyMapping">
            <parameter name="Department">Development</parameter>
            <parameter name="Salary">$Salary<50000</parameter>
        </mapping>
    </project>
</root>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root xmlns="http://www.informatica.com/Parameterization/1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema" version="2.0">
    <!-- File 4    -->
    <project name="MyProject">
        <mapping name="MyMapping">
            <parameter name="Department">Development</parameter>
            <parameter name="Salary">50000<$Salary</parameter>
        </mapping>
    </project>
</root>
```

You run the mapping. Each time that you run the mapping, you use a different parameter file. The target contains the filtered data according to the parameter values in the parameter file.

Parameter File Structure

The Data Integration Service reads a parameter file by identifying the parameter file structure. The parameter file structure allows the Data Integration Service to locate the parameters and their defined values.

The following table describes the elements that you can use in a parameter file:

Element	Description
parameter	Defines the parameter value that corresponds to a parameter.
mapping	Defines a mapping in a project or an application.
workflow	Defines a workflow in a project or an application.
project	Defines a project in the Model repository that contains objects that use parameters.
folder	Defines a folder within a project in the Model repository.
application	Defines a deployed application that contains objects that use parameters.

Parameter Element

The parameter element defines the parameter value that corresponds to a parameter. The parameter element must exist within a mapping or a workflow element. The mapping or workflow element defines the mapping or workflow where the Data Integration Service applies the parameter value listed in the parameter element.

For example, you can specify a parameter value within the following element:

```
<parameter name="MyParameter">default_value</parameter>
```

Mapping Element

The mapping element defines a mapping in a project or an application. A mapping element contains one or more parameter elements that define parameter values for the mapping or for any non-reusable transformation, mapplet, or data object in the mapping that accepts parameters.

If you want to specify reusable object parameter values, expose the reusable object parameters as mapping parameters. Specify the mapping parameter values in the parameter file.

For example, you can specify parameters for a mapping within the following elements:

```
<mapping name="Mapping1">
        <parameter name="Parameter1">value1</parameter>
</mapping>
```

Workflow Element

The workflow element defines a workflow in a project or an application. A workflow element contains one or more parameter elements that define parameter values for the workflow.

For example, you can specify parameters for a workflow within the following elements:

```
<workflow name="MyWorkflow">
    <parameter name="MyParameter">default_value</parameter>
</workflow>
```

Project Element

The project element defines a project in the Model repository that contains objects that use parameters. The project element identifies the parameter values to use when you run a specific mapping or workflow that is stored in the project. The project element can include mapping or workflow elements. It can also include a folder element if the mapping or workflow is stored in a folder in the project.

For example, you want to specify a parameter for a mapping in a parameter file. In the Model repository, the mapping `MyMapping` is stored under a folder `MyFolder` within a project `MyProject`. Define the parameter within the following elements:

```
<project name=”MyProject”>
    <folder name=”MyFolder”>
        <mapping name=”MyMapping”>
            <parameter name=”MyParameter”>default_value</parameter>
        </mapping>
    </folder>
</project>
```

Multiple projects can contain mappings that use the same name. To differentiate mappings with the same name, use the fully-qualified name by specifying the projects and folders where the mappings are stored. For example, you can define a mapping `MyMapping` in two different projects within the following elements:

```
<project name=”Project1”>
    <folder name=”Folder1”>
        <mapping name=”MyMapping”>
            <parameter name=”MyParameter”>default_value</parameter>
        </mapping>
    </folder>
</project>

<project name=”Project2”>
    <folder name=”Folder2”>
        <mapping name=”MyMapping”>
            <parameter name=”MyParameter”>default_value</parameter>
        </mapping>
    </folder>
</project>
```

Folder Element

The folder element defines a folder within a project in the Model repository. Use a folder element if objects are organized in multiple folders within the project. A folder element can contain a mapping or a workflow element, or another folder element.

For example, you want to specify a parameter for a mapping in a parameter file. In the Model repository, the mapping `MyMapping` is stored under multiple levels of folders within a project `MyProject`. Define the parameter within the following elements:

```
<project name="MyProject">
    <folder name="Folder1">
        <folder name="Folder2">
            <folder name="Folder3">
                <mapping name="MyMapping">
                    <parameter name="MyParameter">default_value</parameter>
                </mapping>
            </folder>
        </folder>
    </folder>
</project>
```

Application Element

The application element defines parameters for mapping or workflow objects in a deployed application. The application element defines the deployed application and the parameter values that the Data Integration Service applies to the mappings and workflows in the application. If a mapping or a workflow appears in multiple applications, use the application element to define different sets of parameters for the mapping or workflow depending on which application runs.

The application element can contain mapping or workflow elements. It can also contain project and folder elements if the mappings or workflows are organized in multiple projects or folders in the application.

For example, you want to specify a parameter for a mapping in a parameter file. The mapping `MyMapping` is part of a deployed application `MyApplication`. Define the parameter within the following elements:

```
<application name="MyApplication">
    <mapping name="MyMapping">
        <parameter name="MyParameter">default_value</parameter>
    </mapping>
</application>
```

If the mapping is stored under a project `MyProject` and a folder `MyFolder` within the application, define the parameter within the following elements:

```
<application name="MyApplication">
    <project name="MyProject">
        <folder name="MyFolder">
            <mapping name="MyMapping">
                <parameter name="MyParameter">default_value</parameter>
            </mapping>
        </folder>
    </project>
</application>
```

You can also specify the same mapping element in multiple applications to use a different parameter value for the mapping depending on which application runs on the Data Integration Service. For example, you want to specify a parameter value for the parameter `MyParameter` in the mapping `MyMapping` when the application `Application1` runs. But, you want to specify a different parameter value for the same mapping when the application `Application2` runs. Define the parameter values within the following elements:

```
<application name="Application1">
    <mapping name="MyMapping">
        <parameter name="MyParameter">value1</parameter>
    </mapping>
```

```
</application>
<application name="Application2">
    <mapping name="MyMapping">
        <parameter name="MyParameter">value2</parameter>
    </mapping>
</application>
```

When `Application1` runs, the Data Integration Service applies the parameter value `value1` in the mapping. When `Application2` runs, the Data Integration Service applies the parameter value `value2` in the mapping.

How to Generate a Parameter File

Generate a parameter file to reuse the same parameter file as a template to create and customize multiple parameter files with different parameter values depending on the mapping or workflow that you run, or the application that you deploy.

You can generate a parameter file in one of the following ways:

- Export mapping or workflow parameters as a parameter file in the Developer tool.
- List mapping or workflow parameters in the command line. Use the list of parameters to create a parameter file.

Exporting a Parameter File from the Developer Tool

Export a parameter file from the Developer tool to generate a parameter file based on the parameters configured for a mapping or workflow. You can specify the parameter file name and choose where to save the file.

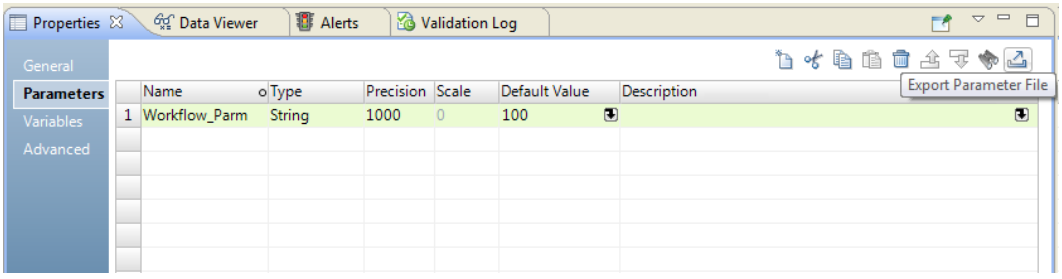
To export a parameter file, complete the following tasks:

1. In the Developer tool, define the parameters and the parameter default values for a mapping or a workflow.
2. In the **Properties** view, navigate to the **Parameters** tab.

   The Parameters tab lists the defined parameters.
3. Click **Export Parameter File**.

   When you export a parameter file, the Developer tool creates a parameter file that contains the mapping or workflow parameters.

   For example, the following image shows the **Export Parameter File** option on the Parameters tab for a workflow:



   When you export the workflow parameter `Workflow_Parm`, the Developer tool creates the following parameter file:

```
<?xml version="1.0" encoding="UTF-8"?>
<root version="2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema" xmlns="http://
www.informatica.com/Parameterization/1.0">
    <project name="Orders">
        <workflow name="Customer_Workflow">
            <parameter name="Workflow_Parm">100</parameter>
```

```
        </workflow>
    </project>
</root>
```

Creating a Parameter File from the Command Line

You can create a parameter file from the command line by listing the parameters used in a mapping or a workflow. Use the infacmd ms ListMappingParams command to list the parameters in a mapping, or use the infacmd wfs ListWorkflowParams command to list the parameters in a workflow. You can use the output of this command to create a parameter file.

To list the parameters and create a parameter file, complete the following tasks:

1. Run one of the following commands:
   - Run the infacmd ms ListMappingParams command to list the parameters in a mapping.
   - Run the infacmd wfs ListWorkflowParams command to list the parameters in a workflow.

   The -o option sends the command output to an .xml file.
2. If you do not specify the -o option, you can copy the command output to an .xml file.
3. Save the .xml file.

How to Edit a Parameter File

After you generate a parameter file, you can use the same parameter file as a template to create unique parameter files. The generated parameter file uses the following default structure:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root xmlns="http://www.informatica.com/Parameterization/1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema" version="2.0">
    <!--
                        Parameter file elements
    -->
</root>
```

Edit the parameter values in the generated file to define different parameters for a mapping or workflow. Save each set of parameters as a separate parameter file. You can also edit the parameter file structure to redefine the parameter file elements. If you define additional parameter values in a parameter file, the parameters associated with the parameter values must be configured in the mapping or workflow. If the parameters are not configured in the mapping or workflow, the Data Integration Service ignores the parameters.

Editing a Parameter Value in a Parameter File

Once you have a parameter file, you can reuse the parameter file as a template to define different parameter values for a mapping or a workflow. Save the parameters as separate parameter files.

To edit the parameter values in a parameter file, complete the following tasks:

1. Open the XML file.
2. Navigate to the mapping or workflow element that contains the parameter element that you want to change.
3. Edit the value of the parameter element.

   For example, you can edit the parameter defined by the element

   ```
   <parameter name="MyParameter">value1</parameter>
   ```

   To edit the default value from `value1` to `value2`, make the following change:

   ```
   <parameter name="MyParameter">value1</parameter>
   ```

The value `value2` is the parameter value that the Data Integration Service applies to the mapping or workflow if the mapping or workflow runs using the parameter file.

Editing the Parameter File Structure

In the generated parameter file, you can also edit the structure to redefine the parameters that the parameter file contains.

For example, you generate a parameter file that contains parameters for a workflow. The parameters might be defined within the following elements:

```
<project name=”MyProject”>
<workflow name=”MyWorkflow”>
    <parameter name=”Parameter1”>value1</parameter>
    <parameter name=”Parameter2”>value2</parameter>
</workflow>
</project>
```

Then, you decide that you want to use these parameter values only when the workflow runs in a specific application. To specify the application, you can add an application element. Define the parameters within the following elements:

```
<application name="MyApplication">
    <project name=”MyProject”>
        <workflow name=”MyWorkflow”>
            <parameter name=”Parameter1”>value1</parameter>
            <parameter name=”Parameter2”>value2</parameter>
        </workflow>
    </project>
</application>
```

Rules and Guidelines for Parameter Files

The following rules and guidelines apply when you create parameter files:

- The application element contains mapping or workflow parameters that apply only to the specified application when the application is deployed. If you run a mapping with a parameter file from the Developer tool and do not deploy the mapping as an application, do not specify an application element. Specify the mapping within a project element.
- Parameter values cannot be empty. For example, the Data Integration Service fails the mapping run if the parameter file contains the following entry:

  ```
  <parameter name="Param1"> </parameter>
  ```
- Within an element, artifact names are not case-sensitive. Therefore, the Data Integration Service interprets <parameter name="SrcDir"> and <parameter name="Srcdir"> as the same application.
- A parameter that identifies a reference table must use a forward-slash (/) to separate folder names in a repository folder path.
- You cannot define system parameter values in a parameter file.

Author

Margarita Pelyushenko