

---

---

## How Do I Handle Flat Files with Variable Names?

### Scenario

Your company relies on a legacy system that writes data to a flat file on a daily basis and assigns a unique name to the file based on the date and time of its creation. You would like to create a mapping that uses the generated flat files as a source and transforms and loads the data to a relational database. However, mappings require files to have permanent names and, in this situation, the name of the source file changes each time the file is created.

### Solution

In Warehouse Builder, you can design a process flow that locates the generated file in a specific directory, renames it to a permanent name you designate, and launches a dependent mapping. You can now use the permanent flat file name as the source for your mapping.

### Case Study

This case study describes how to create a process flow and a mapping to extract data from a legacy system that generates flat files with variable names. The process flow relies on the use of an external process activity. Assume the following information for the purposes of this case study:

- **Generated Flat File:** The legacy system generates a flat file containing sales data on a daily basis. It saves the file to the `c:\staging_files` directory and names the file based on the time and date, such as `sales010520041154.dat`. Every generated file is saved to the same directory and begins with the word `sales`, followed by the timestamp information.
- **Permanent Flat File Name:** You decide to rename the generated file name to `s_data.dat`. This is the name you reference as the flat file source in the mapping.
- **Process Activity:** You design a process flow named `OWF_EXT` to execute batch commands in DOS that copies the generated file, saves it as `s_data.dat`, and deletes the originally generated file.

Your objective is to create logic that ensures the generated flat file is renamed appropriately before it triggers the execution of a mapping.

**To extract data from a generated flat file with name that varies with each generation, refer to the following sections:**

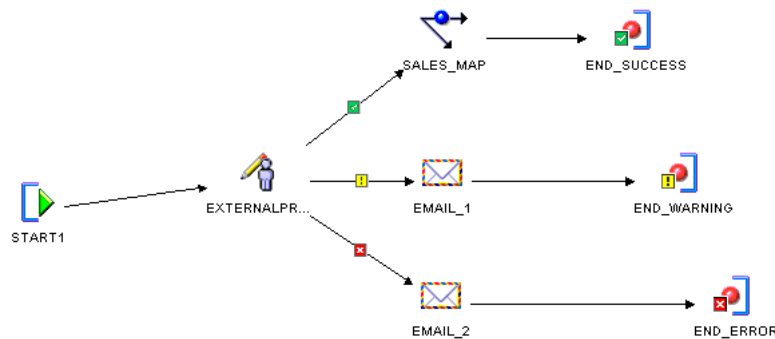
1. ["Creating the Process Flow"](#) on page 7-2

2. ["Setting Parameters for the External Process Activity"](#) on page 7-2
3. ["Configuring the External Process Activity"](#) on page 7-4
4. ["Designing the Mapping"](#) on page 7-5
5. ["Deploying and Executing"](#) on page 7-5

## Creating the Process Flow

Create a process flow that launches a mapping on the condition that the external process activity completes successfully. Your process flow should resemble [Figure 7-1](#).

**Figure 7-1 Process Flow with External Process Transitioning to a Mapping**



## Setting Parameters for the External Process Activity

This section describes how to specify the DOS commands for renaming the generated file. The DOS commands you issue from the external process activity should be similar to the following:

```
copy c:\staging_files\sales*.* c:\staging_files\s_data.dat
del c:\staging_files\sales*.*
```

The first command copies the temporary file into a file with a fixed name `s_data.dat`. The second command deletes the originally generated file.

You can either direct Warehouse Builder to a file containing the script of commands or you can store the commands in the Warehouse Builder user interface. Choose one of the following methods:

- [Method 1: Write a script within Warehouse Builder](#)
- [Method 2: Call a script maintained outside of Warehouse Builder](#)

### Method 1: Write a script within Warehouse Builder

Choose this method when you want to maintain the script in Warehouse Builder. Consider using this method when the script is small and need not be very flexible.

For this method, write or copy and paste the script into the `VALUE` column of the `SCRIPT` parameter. In the `COMMAND` parameter, type the path to the DOS shell command such as `c:\winnt\system32\cmd.exe`. Also, type the `${Task.Input}` variable into the `VALUE` column of the `PARAMETER_LIST` parameter. Your Activity View should resemble [Figure 7-2](#).

**Figure 7-2 External Process Parameters When Script Maintained in Warehouse Builder**

ACTIVITY	PARAMETER	DATATYPE	DIR...	BIND...	VALUE
EXTERNALPROCESS	COMMAND	STRING	IN		c:\winnt\system32\cmd.exe
	PARAMETER_L...	STRING	IN		?*\${Task.Input}*?
	SUCCESS_TH...	INTEGER	IN		0
	SCRIPT	STRING	IN		copy c:\staging_files\sales*. * c:\staging_files\sales_data.dat

Although this case study does not illustrate it, you can use substitution variables in the script when you maintain it in Warehouse Builder. This prevents you from having to update activities when server files, accounts, and passwords change.

Table 7-1 lists the substitute variables you can type for the external process activity. *Working* refers to the machine hosting the Runtime Service, the *local* machine in this case study. Remote refers to a server other than the Runtime Service host. You designate which server is remote and local when you configure the activity as described in "Configuring the External Process Activity" on page 7-4. These values are set when you register the locations at deployment.

**Table 7-1 Substitute Variables for the External Process Activity**

Variable	Value
\${Working.Host}	The host value for the location of the Runtime Service host.
\${Working.User}	The user value for the location of the Runtime Service host.
\${Working.Password}	The password value for the location of the Runtime Service host.
\${Working.RootPath}	The root path value for the location of the Runtime Service host.
\${Remote.Host}	The host value for a location other than the Runtime Service host.
\${Remote.User}	The user value for a location other than the Runtime Service host.
\${Remote.Password}	The password value for a location other than the Runtime Service host.
\${Remote.RootPath}	The root path value for a location other than the Runtime Service host.
\${Deployment.Location}	The deployment location.

## Method 2: Call a script maintained outside of Warehouse Builder

If extra maintenance is not an issue, you can point Warehouse Builder to a file containing a script including the necessary commands. This method is more flexible as it allows you to pass in parameters during execution of the process flow.

The following example shows how to call an external process script outside of Warehouse Builder and illustrates how to pass parameters into the script during execution of the process flow. This example assumes a Windows operating system. For other operating systems, issue the appropriate equivalent commands.

### To call a script outside the external process activity:

1. Write the script and save it on the file directory. For example, you can write the following script and save it as `c:\staging_files\rename_file.bat`:

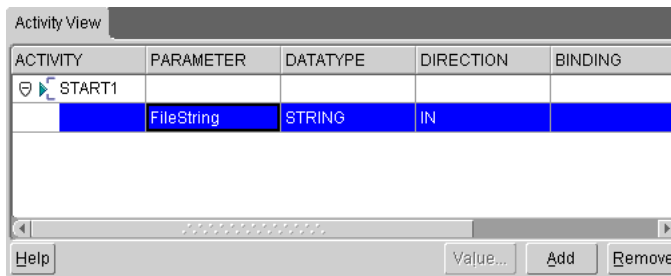
```
copy c:\staging_files\%1*.dat c:\staging_files\s_data.dat
del c:\staging_files\%1*.dat
```

In this sample script, we pass a parameter %1 to the script during the execution of the process flow. This parameter represents a string containing the first characters of the temporary file name, such as sales010520041154.

2. Select the start activity on the canvas to view and edit activity parameters in the Activity View displayed in the lower left panel of the Process Flow Editor.

To add a start parameter, click **Add** on the lower right corner of the Activity View. Create a start parameter named FILE\_STRING as shown in [Figure 7-3](#). During execution, Warehouse Builder will prompt you to type a value for FILE\_STRING to pass on to the %1 parameter in the rename\_file.bat script.

**Figure 7-3 Start Activity in the Activity View**



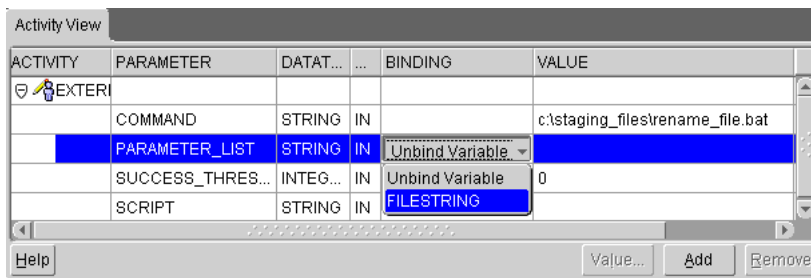
3. Select the external process activity on the canvas and edit its parameters.

For the COMMAND parameter, type the path to the script in the column labelled VALUE. If necessary, use the scroll bar to scroll to the right and reveal the column. For this example, type c:\staging\_files\rename\_file.bat.

For PARAMETER\_LIST, click the column labelled BINDING and select the parameter you defined for the start activity, FILE\_STRING.

Accept the defaults for all other parameters for the external process. Your Activity View for the external process activity should resemble [Figure 7-4](#).

**Figure 7-4 External Process Parameters When Calling an Outside Script**



## Configuring the External Process Activity

When you apply conditions to the outgoing transitions of an external process, you must define the meaning of those conditions when you configure the external process activity.

**To configure the external process activity:**

1. Right-click the process flow on the navigation tree and select **Configure**.

2. Expand the external process activity and the Path Settings. Warehouse Builder displays the configuration settings.
3. Complete this step if you wrote the script in the Warehouse Builder user interface using the substitution variables related to Remote Location, Working Location, and Deployment Location as listed in [Table 7-1](#) on page 7-3. Use the drop down list to select the location for each.

Because this case study does not use substitution variables, accept the defaults values.

4. Set the Deployed Location to the machine where you deploy the process flow.
5. Set **Use Return as Status** to true.

This ensures that the process flow uses the external process return codes for determining which outgoing transition to activate. For the process flow in this case study, shown in [Figure 7-1](#) on page 7-2, if the external process returns a success value, the process flow continues down the success transition and executes the downstream mapping.

## Designing the Mapping

Now you can design a mapping with `s_data.dat` as the source. You can create a PL/SQL mapping or a SQL\*Loader mapping. For a PL/SQL, map the flat file source to an external table and design the rest of the mapping with all the operators available for a PL/SQL mapping. For SQL\*Loader, map the flat file source to a staging table and limit the mapping to those operators permitted in SQL\*Loader mappings.

## Deploying and Executing

Deploy the mapping. Also, deploy the process flow package or module containing the process flow `OWF_EXT`.

When you execute the process flow, Warehouse Builder prompts you to type values for the parameter you created to pass into the script, `FILE_STRING`. For this case study, type `?sales` where the question mark is the separator, as shown in [Figure 7-5](#). The external activity then executes the command `rename_file.bat sales`.

**Figure 7-5 External Process Activity in the Activity View**

