
How Do I Transfer Remote Files?

Scenario

Developers at your company designed mappings that extract, transform, and load data. The source data for the mapping resides on a server separate from the server that performs the ETL processing. You would like to create logic that transfers the files from the remote machine and triggers the dependent mappings.

Solution

In Warehouse Builder, you can design a process flow that executes file transfer protocol (FTP) commands and then launches a mapping. For the process flow to be valid, the FTP commands must involve transferring data either from or to the server with the Runtime Service installed. To move data between two machines, neither of which host the Runtime Service, first transfer the data to the Runtime Service host machine and then transfer the data to the second machine.

You can design the process flow to launch different activities depending upon the success or failure of the FTP commands.

Case Study

This case study describes how to transfer files from one machine to another and launch a dependent mapping. The case study provides examples of all the necessary servers, files, and user accounts.

- **Data host machine:** For the machine hosting the source data, you need a user name and password, host name, and the directory containing the data. In this case study, the machine hosting the data is a UNIX server named `salesrv1`. The source data is a flat file named `salesdata.txt` located in the `/usr/stage` directory.
- **Runtime Service host machine:** In this case study, Warehouse Builder and the Runtime Service are installed on a machine called `local` with a Windows operating system. `Local` executes the mapping and the process flow.
- **Mapping:** This case study assumes there is a mapping called `salesresults` that uses a copy of `salesdata.txt` stored on `local` at `c:\temp` as its source.
- **FTP Commands:** This case study illustrates the use of a few basic FTP commands on the Windows operating system.

Your objective is to create logic that ensures the flat file on `salesrv1` is copied to the `local` machine and then trigger the execution of the `salesresults` mapping.

To transfer files and launch a dependent mapping, refer to the following sections:

1. ["Defining Locations"](#) on page 2-2 for details.

2. "Creating the Process Flow" on page 2-2
3. "Setting Parameters for the FTP Activity" on page 2-2
4. "Configuring the FTP Activity" on page 2-5
5. "Registering the Process Flow for Deployment" on page 2-6

After you complete the instructions in the above sections, you can run the process flow.

Defining Locations

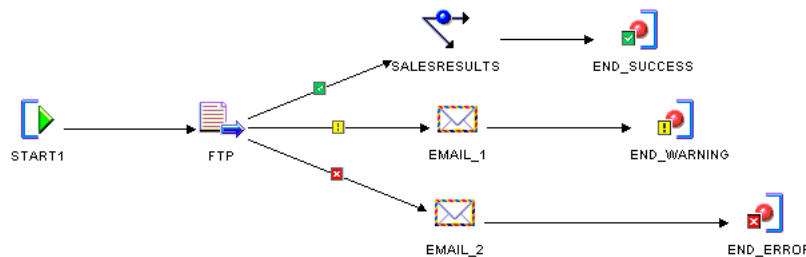
Locations are logical representations of the various data sources and destinations in the warehouse environment. In this scenario, the locations are the logical representations of the host and path name information required to access a flat file. Warehouse Builder requires these definitions for deploying and running the process flow. When you deploy the process flow, Warehouse Builder prompts you to type the host and path name information associated with each location. You must define locations for each machine involved in the data transfer.

To define locations, right-click the appropriate Locations node on the navigation tree and right-click or press **Ctrl+N** to create a new location. For `salesrv1`, right-click **File Location** and create a location named `REMOTE_FILES`. Repeat the step for `local` and create the location `LOCAL_FILES`.

Creating the Process Flow

Use the Process Flow Editor to create a process flow with an FTP activity that transitions to the `salesresults` mapping on the condition of success. Your process flow should appear similar to [Figure 2-1](#).

Figure 2-1 Process Flow with FTP Transitioning to a Mapping



Setting Parameters for the FTP Activity

This section describes how to specify the commands for transferring data from the remote server, `salesrv1`, to the `local` machine. You specify the FTP parameters by typing values for the FTP activity parameters on the Activity View as displayed in [Figure 2-2](#).

Warehouse Builder offers you flexibility on how you specify the FTP commands. Choose one of the following methods:

- **Method 1: Write a script in Warehouse Builder:** Choose this method when you want to maintain the script in Warehouse Builder and/or when password security to servers is a requirement.

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 FTP executable such as `c:\winnt\system32\ftp.exe`. Also, type the `Task.Input` variable into the VALUE column of the PARAMETER_LIST parameter.

- **Method 2: Call a script maintained outside of Warehouse Builder:** If password security is not an issue, you can direct Warehouse Builder to a file containing a script including the FTP commands and the user name and password.

To call a file on the file system, type the appropriate command in PARAMETERS_LIST to direct Warehouse Builder to the file. For a Windows operating system, type the following: `?-s:<file path\file name>?"`

For example, to call a file named `move.ftp` located in a temp directory on the C drive, type the following: `?-s:c:\temp\move.ftp"?`

Leave the SCRIPT parameter blank for this method.

Example: Writing a Script in Warehouse Builder for the FTP Activity

The following example illustrates Method 1 described above. It relies on a script and the use of substitution variables. The script navigates to the correct directory on `salessrv1` and the substitution variables are used for security and convenience.

This example assumes a Windows operating system. For other operating systems, issue the appropriate equivalent commands.

To define a script within the FTP activity:

1. Select the FTP 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.
2. For the **COMMAND** parameter, type the path to the FTP executable in the column labelled `Value`. If necessary, use the scroll bar to scroll to the right and reveal the column labelled `Value`.

For windows operating systems, the FTP executable is often stored at `c:\winnt\system32\ftp.exe`.

3. For **PARAMETER_LIST**, type the `Task.Input` variable.

When defining a script in Warehouse Builder and using Windows FTP, you must type `?-s:${Task.Input}?"` into **PARAMETER_LIST**.

For UNIX, type `?"${Task.Input}?"`.

4. Navigate and highlight the **SCRIPT** parameter. Your Activity View should display similar to [Figure 2-2](#).

Figure 2–2 Activity View for FTP Activity Using a Script

ACTIVITY	PARAMETER	DATATYPE	DIREC...	BINDING	VALUE
FTP					
	COMMAND	STRING	IN		c:\winnt\system32\ftp.exe
	PARAMETER_LIST	STRING	IN		?*-s:\${Task.Input}?
	SUCCESS_THRESHOLD	INTEGER	IN		0
	SCRIPT	STRING	IN		open \${Remote.Host}

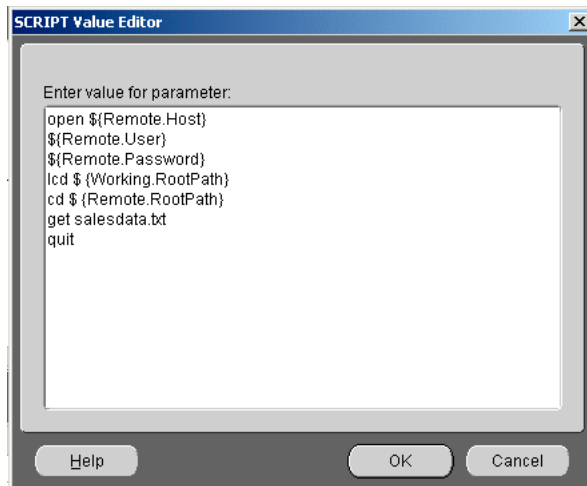
5. Click **Value...** displayed at the bottom of the Activity View.

Warehouse Builder displays the **SCRIPT Value Editor**. Write or copy and paste FTP commands into the editor.

[Figure 2–2](#) shows a script that opens a connection to the remote host, changes the directory to the local machine, changes the directory to the remote host, transfers the file, and closes the connection.

Notice that the script in [Figure 2–2](#) includes `${Remote.User}` and `${Remote.Password}`. These are substitution variables. Refer to "[Using Substitution Variables](#)" for more details.

Figure 2–3 SCRIPT Value Editor Using Substitution Variables



Using Substitution Variables

Substitution variables are available only when you choose to write and store the FTP script in Warehouse Builder.

Use substitution variables to prevent having to update FTP activities when server files, accounts, and passwords change. For example, consider that you create 10 process flows that utilize FTP activities to access a file on `salessrv1` under a specific directory. If the file is moved, without the use of substitution variables, you must update each FTP activity individually. With the use of substitution variables, you need only update the location information as described in "[Defining Locations](#)" on page 2-2.

Substitution variables are also important for maintaining password security. When Warehouse Builder executes an FTP activity with substitution variables for the server passwords, it resolves the variable to the secure password you provided for the associated location.

[Table 2-1](#) lists the substitute variables you can provide for the FTP activity. `Working` refers to the machine hosting the Runtime Service, the *local* machine in this case study. `Remote` refers to the other server involved in the data transfer. You designate which server is remote and local when you configure the FTP activity. For more information, see "[Configuring the FTP Activity](#)" on page 2-5.

Table 2-1 *Substitute Variables for the FTP Activity*

Variable	Value
<code>\${Working.RootPath}</code>	The root path value for the location of the Runtime Service host.
<code>\${Remote.Host}</code>	The host value for the location involved in transferring data to or from the Runtime Service host.
<code>\${Remote.User}</code>	The user value for the location involved in transferring data to or from the Runtime Service host.
<code>\${Remote.Password}</code>	The password value for the location involved in transferring data to or from the Runtime Service host.
<code>\${Remote.RootPath}</code>	The root path value for the location involved in transferring data to or from the Runtime Service host.

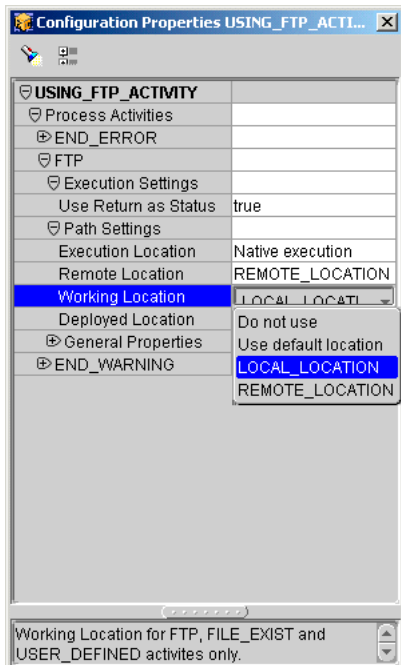
Configuring the FTP Activity

As part of configuring the complete process flow, configure the FTP activity.

To configure the FTP Activity:

1. Right-click the process flow on the navigation tree and select **Configure**.
2. Expand the FTP activity and the Path Settings. Warehouse Builder displays the configuration settings as shown in [Figure 2-4](#).

Figure 2–4 Configuration Settings for the FTP Activity



3. Set Remote Location to `REMOTE_LOCATION` and Working Location to `LOCAL_LOCATION`.
4. Set **Use Return as Status** to true. This ensures that the process flow uses the FTP return codes for determining which outgoing transition to activate. For the process flow in this case study, shown in [Figure 2–1](#) on page 2-2, if FTP returns a success value of 1, the process flow continues down the success transition and executes the `salesresults` mapping.

Registering the Process Flow for Deployment

After you complete these instructions, you can deploy and run the process flow. To deploy the process flow, launch the Deployment Manager by right-clicking and selecting **Deploy** from either the process flow module or package on the navigation tree. The Deployment Manager prompts you to register the `REMOTE_LOCATION` and the `LOCAL_LOCATION`.

[Figure 2–5](#) shows the registration information for the `REMOTE_LOCATION`. For the `LOCAL_FILES`, only the root path is required.

Figure 2–5 Example Location Registration Information

The image shows a dialog box titled "Location Registration: REMOTE_FILES". The text inside the dialog reads: "Type in the values for the File System location REMOTE_FILES." Below this text are four input fields:

- User Name: sales
- Password: *****
- Host Name: salesrv1
- Root Path: /usr/stage

At the bottom of the dialog, there are three buttons: "Help", "OK", and "Cancel".

Now you can run the process flow.

