BEA WebLogic Integration™

xFile Plug-in User Guide

Release 7.1
Release Date: June 2003
Revision Date: April 2004

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA WebLogic Integration xFile Plug-in User Guide

Part Number	Date	Software Version
N/A	Released: June 2003 Revised: April 2004	7.1

Contents

About This Document

This document explains how to deploy and use the BEA WebLogic Integration xFile Plug-in. It is organized as follows:

- Chapter 1, "Introducing the BEA WebLogic Integration xFile Plug-in," introduces the BEA WebLogic xFile Plug-in and provides a brief description of the xFile Plug-in.
- Chapter 2, "Deploying the xFile Plug-in," contains instructions for deploying the plug-in with WebLogic Integration 2.1 Service Pack 2 (SP2) and WebLogic Integration 7.0 SP2.

  Note: The plug-in is included with WebLogic Integration 7.0 SP5. If you have installed that release, the plug-in is automatically deployed on server startup.
- Chapter 3, "Using the xFile Plug-in," describes how to use the xFile Plug-in features and describes the workflow process in detail.
- Chapter 4, "Configuring the xFile Plug-in for a Migrated Domain," provides details on configuring the xFile Plug-in for a migrated domain.
- Chapter 5, "xFile Plug-in Example," provides a step-by-step example of using xFile Plug-in to transfer a file to a remote file system.

What You Need to Know

This document is intended for workflow designers and system integrators who develop client interfaces between file systems and other applications.

The information provided in this document requires you to have in-depth knowledge of Workflow Design and Workflow Templates, File Transfer Protocol (FTP), and WebLogic Integration Studio. Additionally, it is assumed that you know Web technologies and have a general understanding of Microsoft Windows and UNIX systems.

If you do not have the required knowledge of workflows or the WebLogic Integration Studio, see the following documents:

- *Using the WebLogic Integration Studio* at http://edocs.bea.com/wli/docs70/studio/index.htm
- *Learning to Use BPM with WebLogic Integration* at http://edocs.bea.com/wli/docs70/bpmtutor/index.htm

e-docs Web Site

BEA Product Documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the "e-docs" Product Documentation page at http://edocs.bea.com.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File→Print option on your Web browser.

A PDF version of this document is available on the WebLogic Integration documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Integration documentation Home page, click the PDF files button, and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com.

Related Information

The following BEA WebLogic Integration documents contain information that is relevant to using this product:

- *Using the WebLogic Integration Studio* at http://edocs.bea.com/wli/docs70/studio/index.htm
- *Learning to Use BPM with WebLogic Integration* at http://edocs.bea.com/wli/docs70/bpmtutor/index.htm

Contact Us!

Your feedback on the BEA WebLogic Integration documentation is important to us. Send us e-mail at docsupport@beasys.com if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the WebLogic Integration documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Integration xFile Plug-in 7.1 release.

If you have any questions about this version of xFile Plug-in, or if you have problems installing and running the product, contact BEA Customer Support through BEA WebSupport at www.bea.com. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and filenames and their extensions.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Examples*: LPT1 SIGNON OR
`{}`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

Convention	Item
`[]`	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: `buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...`
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
. . .	Indicates one of the following in a command line: ■ That an argument can be repeated several times in a command line ■ That the statement omits additional optional arguments ■ That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: `buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...`
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER

1 Introducing the BEA WebLogic Integration xFile Plug-in

This section introduces the BEA WebLogic Integration xFile Plug-in and describes how the xFile Plug-in functions in a Business Process Management (BPM) workflow. It includes the following topics:

About the xFile Plug-in

The BEA WebLogic Integration xFile Plug-in extends the functionality of the BEA WebLogic Integration BPM Studio. Specifically, the xFile Plug-in provides basic actions for working with files and transferring files between local and remote file systems. The applicable data formats include `ASCII`, `XML`, and `Binary`. Actions are the basic units of work, and are performed during BPM functions. They define workflow and task behavior.

For more information, see *Programming BPM Plug-ins for WebLogic Integration* at http://edocs.bea.com/wli/docs70/devplug/index.htm.

What the xFile Plug-in Does

The xFile Plug-in has the following functionality:

- Read a file and assigning the content to a workflow variable
- Write to a file using the content of a workflow variable
- Append the content of a workflow variable to a file
- Create a file using the content of a workflow variable
- Copy a file from one location to another
- Move a file from one location to another
- Delete a file
- Transfer a file to and from a remote file system, through FTP
- Start a workflow
- Resume a workflow
- Manipulate rows

What You Need to Know

This document is written for workflow designers and system integrators who develop client interfaces between file systems and other applications. The information provided in this document requires that you are have in-depth knowledge of workflow design and workflow templates, File Transfer Protocol (FTP), and WebLogic Integration Studio. Additionally, it is assumed that you know Web technologies and have a general understanding of Microsoft Windows and UNIX systems.

If you do not have the required knowledge of workflows or the WebLogic Integration Studio, see the following documents:

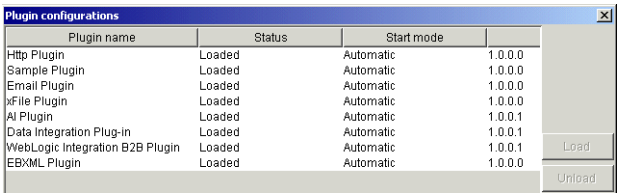
- *Using the WebLogic Integration Studio* at http://edocs.bea.com/wli/docs70/studio/index.htm.
- *Learning to Use BPM with WebLogic Integration* at http://edocs.bea.com/wli/docs70/bpmtutor/index.htm.

xFile Plug-in and the Plug-in Framework

Like other plug-ins that extend BPM functionality, the xFile Plug-in adheres to the BPM Plug-in specification. For more information, see *Programming BPM Plug-ins for WebLogic Integration* at http://edocs.bea.com/wli/docs70/devplug/index.htm. The Plug-in Configurations window displays the plug-ins and their status, as shown in the following figure.

For details on how to see the Plug-in Configurations window, see "Verifying Deployment," in Chapter 2, "Deploying the xFile Plug-in."

Figure 1-1 Plug-in Configurations Window



2 Deploying the xFile Plug-in

If you have installed WebLogic Integration 7.0 Service Pack 5, the plug-in is included in your installation and is deployed on server start up. See "Verifying Deployment" on page 2-12 to verify deployment.

If you have an earlier release of WebLogic Integration installed and do not wish to upgrade the latest WebLogic Integration 7.0 Service Pack, you can install the plug-in as described in the following sections:

- Understanding the Representation of Paths
- Deploying on WebLogic Integration 7.0 SP2
- Deploying on WebLogic Integration 2.1 SP2
- Updating the BEA License
- Verifying Deployment

You can deploy the xFile Plug-in on WebLogic Integration 7.0 (running on WebLogic Server 7.0) or WebLogic Integration 2.1 (running on WebLogic Server 6.1).

Note: The procedures provided in "Deploying on WebLogic Integration 7.0 SP2" and "Deploying on WebLogic Integration 2.1 SP2" assume that you are using an already existing database.

Understanding the Representation of Paths

The location of files in the WebLogic Integration environment depends on options selected during installation and configuration. Therefore, the following conventions are used throughout to represent paths.

- *BEA_HOME* represents the BEA Home directory specified for your WebLogic installation.

  If you install the product in the default location on a Windows system, *BEA_HOME* represents `C:\bea`.

- *WLI_HOME* represents the root of your WebLogic Integration installation.
  - If you install WebLogic Integration 2.1 in the default location on a Windows system, *WLI_HOME* represents `C:\bea\wlintegration2.1`.
  - If you install WebLogic Integration 7.0 in the default location on a Windows system, *WLI_HOME* represents `C:\bea\weblogic700\integration`.

  Note: *WLI_HOME* and *BEA_HOME* also represent the corresponding Windows and UNIX environment variables. For example, the literal interpretation of *WLI_HOME* is `%WLI_HOME%` for Windows and `$WLI_HOME` for UNIX.

- *localhost* represents the IP address of the machine running the WebLogic Server.

Deploying on WebLogic Integration 7.0 SP2

The xFile Plug-in is included with WebLogic Integration 7.0 Service Pack 5. If you have WebLogic Integration 7.0 Service Pack 5 installed, skip to "Verifying Deployment" on page 2-12 to verify deployment.

If you do not wish to upgrade to the latest WebLogic Integration 7.0 Service Pack for some reason, you can deploy the xFile Plug-in on WebLogic Integration 7.0 SP2 by copying the `.jar` files to your WebLogic installation directory, editing the `application.xml` and `config.xml` files, copying a `.sql` script to the appropriate location, and running a command to create the `FILEPOLLX` table in your database.

If you downloaded the xFile Plug-in, the following files are all located in the.zip file you downloaded. If you received the xFile Plug-in on a CD, they are located on the CD.

- `xfileplugin-ejb.jar`
- `plugin-shared.jar`
- `db2\fp_schema.sql`
- `mssql\fp_schema.sql`
- `pointbase\fp_schema.sql`
- `oracle\fp_schema.sql`
- `sybase\fp_schema.sql`
- `cloudscape\fp_schema.sql`
- `xfilesetupdb.cmd`
- `xfilesetupdb.sh`

To deploy the xFile Plug-in on WebLogic Integration 7.0, do the following:

1. From the source location (the directory where you unzipped the `.zip` file or the product CD), copy the following files into the *`BEA_HOME`*`\weblogic700\integration\lib` directory:
   - `xfileplugin-ejb.jar`
   - `plugin-shared.jar`
2. From the *`BEA_HOME`*`\weblogic700\integration\lib\META-INF` directory, open the `application.xml` file and add the lines in bold to the existing configuration, at the location shown here:

```
<application>
.
.
.
   <module>
      <ejb>xfileplugin-ejb.jar</ejb>
   </module>
.
.
.
   <!--BPM Initialization Bean must be deployed after
   BPM plug-ins-->
   <module>
      <ejb>bpm-init-ejb.jar</ejb>
```

```
   </module>
</application>
```

3. Save the file and close it.

4. From the *BEA_HOME*\weblogic700\samples\integration\config\samples directory, open the config.xml file and add the lines in bold to the existing configuration, at the location shown here.

   Note: If you have already created a specific domain, open the config.xml file from that domain. The scripts given here assume that the default domain is samples.

```
<Application
   Deployed="true"
   Name="WLI"
   Path="D:\bea\weblogic700\integration\lib"
   TwoPhase="true">
   <EJBComponent
      Name="WLI-BPM Plugin Manager"
      Targets="myserver"
      URI="wlpi-master-ejb.jar"/>
.
.
.
   <EJBComponent
      Name="WLI-BPM xFile Plug-in"
      Targets="myserver"
      URI="xfileplugin-ejb.jar"/>
.
.
.
</Application>
.
.
.
<JMSServer
   Name="WLIJMSServer"
   Store="JMSWLIStore"
   Targets="myserver"
   TemporaryTemplate="TemporaryTemplate">
.
.
.
   <JMSQueue
      JNDIName="com.bea.wli.bpm.EventQueue"
      Name="WLI_BPM_Event"
      Template="WLI_JMSTemplate"/>
   <JMSQueue
```

```
        JNDIName="com.bea.wli.bpm.xFilePluginQueue"
        Name="WLI_BPM_FPx"
        Template="WLI_JMSTemplate"/>
.
.
.
</JMSServer>
```

5. Save the file and close it.
6. Copy the `fp_schema.sql` file for your database from the source location (the directory where you unzipped the `.zip` file or product CD) to the location listed in the table that follows:

Table 2-1 Location for fp_schema.sql Files for Various Databases

For This Database	Copy the `fp_schema.sql` File to This Location
DB2	*WLI_HOME*\dbscripts\db2\
MS SQL	*WLI_HOME*\dbscripts\mssql\
Oracle	*WLI_HOME*\dbscripts\oracle\
Pointbase	*WLI_HOME*\dbscripts\pointbase\
Sybase	*WLI_HOME*\dbscripts\sybase\

7. Go the location where you unzipped the xFile file and open the file that creates the `FILEPOLLX` database required for the xFile Plug-in.
   - If your operating system is Windows, the file is `\scripts\win32\xfilesetupdb.cmd`.
   - If your operating system is UNIX, the file is `/scripts/unix/xfilesetupdb.sh`.
8. Edit the `.cmd` or `.sh` file to modify the line that starts `- If/I "`*WLI_HOME*`" == "" call...\setEnv.cmd`, to set the correct path.
9. Execute the `xfilesetupdb.cmd` or `xfilesetupdb.sh` command to create the `FILEPOLLX` database.
10. Start the WebLogic Integration Server.

Note: This step assumes that you are using the default domain. If you are using a specific domain, you must run `wliconfig.cmd` and restart the WebLogic Integration Server after you update the relevant schema.

Warning: Running `wliconfig.cmd` for an existing database will drop all the tables and create the tables again, which will lose any saved data.

Deploying on WebLogic Integration 2.1 SP2

To deploy the xFile Plug-in, you copy the `.jar` files to your WebLogic installation directory, edit the `config.xml` file, copy a `.sql` script to the appropriate location, and run a command to create the `FILEPOLLX` table in your database.

If you downloaded the xFile Plug-in, the following files are all located in the.zip file you downloaded. If you received the xFile Plug-in on a CD, they are located on the CD.

- `xfileplugin-ejb.jar`
- `plugin-shared.jar`
- `db2\fp_schema.sql`
- `mssql\fp_schema.sql`
- `pointbase\fp_schema.sql`
- `oracle\fp_schema.sql`
- `sybase\fp_schema.sql`
- `cloudscape\fp_schema.sql`
- `xfilesetupdb.cmd`
- `xfilesetupdb.sh`

To deploy xFile Plug-in in WebLogic Integration 2.1, do the following:

1. From the source location, copy the following files into the *`BEA_HOME`*`\wlintegration2.1\lib` directory:
   - `xfileplugin-ejb.jar`
   - `plugin-shared.jar`

2. From the *BEA_HOME*\wlintegration2.1\config\samples directory, open the config.xml file and add the lines in bold to the existing configuration, at the location as shown here.

   Note: If you have already created a specific domain, open the config.xml file from that domain. The scripts given here assume that the default domain is samples.

```
<Application
   Deployed="true" Name="WLI"
   Path="D:\bea\weblogic700\integration\lib"
   TwoPhase="true">
   <EJBComponent
      Name="WLI-BPM Plugin Manager" Targets="myserver"
      URI="wlpi-master-ejb.jar"/>
.
.
.
   <EJBComponent
      DeploymentOrder="10"
      Name="WLI-BPM xFile Plugin"
      Targets="myserver"
      URI="xfileplugin-ejb.jar"/>
.
.
.
</Application>
.
.
.
<JMSServer
   Name="WLIJMSServer"
   Store="JMSWLIStore"
   Targets="myserver"
   TemporaryTemplate="TemporaryTemplate">
.
.
.
   <JMSQueue
      JNDIName="com.bea.wli.bpm.EventQueue"
      Name="WLI_BPM_Event"
      Template="WLI_JMSTemplate"/>
   <JMSQueue
      JNDIName="com.bea.wli.bpm.xFilePluginQueue"
      Name="WLI_BPM_FP"/>
.
.
```

```
.
</JMSServer>
```

3. Save the file and close it.
4. Copy the `fp_schema.sql` file for your database from the source location (the directory where you unzipped the `.zip` file or product CD) to the location listed in the table that follows:

Table 2-2 Location for fp_schema.sql Files for Various Databases

For This Database	Copy the `fp_schema.sql` File to This Location
DB2	*WLI_HOME*\dbscripts\db2\
MS SQL	*WLI_HOME*\dbscripts\mssql\
Oracle	*WLI_HOME*\dbscripts\oracle\
Cloudscape	*WLI_HOME*\dbscripts\cloudscape\
Sybase	*WLI_HOME*\dbscripts\sybase\

5. Go the location where you unzipped the xFile file and open the file that creates the `FILEPOLLX` database required for the xFile Plug-in.
   - If your operating system is Windows, the file is `\scripts\win32\xfilesetupdb.cmd`.
   - If your operating system is UNIX, the file is `/scripts/unix/xfilesetupdb.sh`.
6. Edit the `.cmd` or `.sh` file to modify the line that starts `- If/I "`*WLI_HOME*`" == "" call...\setEnv.cmd`, to set the correct path.
7. Execute the `xfilesetupdb.cmd` or `xfilesetupdb.sh` command to create the `FILEPOLLX` database.
8. From the *BEA_HOME*`\wlintegration2.1\config\samples` directory, open the `startWeblogic.cmd` file and add `SET SVRCP =%SVRCP%;`*BEA_HOME*`"\lib\plugin-shared.jar;`
9. Save the file and close it.
10. Start the WebLogic Integration Server.

Note: This step assumes that you are using the default domain. If you are using a specific domain, you must run `wliconfig.cmd` and restart the WebLogic Integration Server after you update the relevant schema.

Warning: Running `wliconfig.cmd` for an existing database will drop all the tables and create the tables again, which will lose any saved data.

Precautionary Step to Avoid Errors During Use

The following are steps you could take to avoid exception messages or errors while working with the WebLogic Integration Studio. These steps are, however, not mandatory as part of deployment.

When you open the WebLogic Integration Studio while connected to the WebLogic Server running on either Red Hat Linux 7.2 or Red Hat Enterprise Linux AS 2.1, you may get the following exception message:

```
"ClassNotFoundException"
```

To avoid this, edit the `studio.cmd` file on the machine running the WebLogic Integration Studio, by adding the following lines in bold, at the location shown below:

- If the machine is on Windows:

```
set CP=WLI_HOME\lib\plugin-shared.jar;
WLI_HOME\lib\wlpi-studio.jar;
%WLICP%

CP=%CP%;
WLI_HOME\lib\ebxml-bpm-plugin.jar;
WLI_HOME\lib\wlc-wlpi-plugin.jar;
WLI_HOME\lib\xfileplugin-ejb.jar;
WLI_HOME\lib\wlai-plugin-ejb.jar;
WLI_HOME\lib\sampleplugin-ejb.jar;
WLI_HOME\lib\wlxtpi.jar;
WLI_HOME\lib\ebxml-bpm-plugin.jar;
WLI_HOME\lib\wlai-plugin-ejb.jar;
WLI_HOME\lib\httpplugin-ejb.jar;
WLI_HOME\lib\emailplugin-ejb.jar;
WLI_HOME\lib\mdb-generator.jar;
WLI_HOME\lib\wliserver.jar;
WLI_HOME\lib\wlpi-aux.jar;
WLI_HOME\lib\wlpi-ejb.jar;
WLI_HOME\lib\wlpi-mdb-ejb.jar;
```

```
start %JAVA_HOME%\bin\javaw %COMM_CLIENT_VM% -classpath "%CP%"
"-Dwli.samples=%SAMPLES_HOME%"
"-Dwli.bpm.studio.help=WLI_HOME\docs\help"
"-Durl=http://172.19.138.45:7001"
com.bea.wlpi.client.studio.Studio
```

- If the machine is on UNIX/Red Hat Linux:

```
set CP=WLI_HOME/lib/plugin-shared.jar:
WLI_HOME/lib/wlpi-studio.jar:
$WLICP

CP=$CP:
WLI_HOME/lib/ebxml-bpm-plugin.jar:
WLI_HOME/lib/wlc-wlpi-plugin.jar:
WLI_HOME/lib/wlai-plugin-ejb.jar:
WLI_HOME/lib/sampleplugin-ejb.jar:
WLI_HOME/lib/wlxtpi.jar:
WLI_HOME/lib/ebxml-bpm-plugin.jar:
WLI_HOME/lib/wlai-plugin-ejb.jar:
WLI_HOME/lib/xfileplugin-ejb.jar:
WLI_HOME/lib/httpplugin-ejb.jar:
WLI_HOME/lib/emailplugin-ejb.jar:
WLI_HOME/lib/mdb-generator.jar:
WLI_HOME/lib/wliserver.jar:
WLI_HOME/lib/wlpi-aux.jar:
WLI_HOME/lib/wlpi-ejb.jar:
WLI_HOME/lib/wlpi-mdb-ejb.jar

start $JAVA_HOME/bin/javaw $COMM_CLIENT_VM -classpath "$CP"
"-Dwli.samples=$SAMPLES_HOME"
"-Dwli.bpm.studio.help=WLI_HOME/docs/help"
"-Durl=http://172.19.138.45:7001"
com.bea.wlpi.client.studio.Studio
```

Updating the BEA License

If you have installed WebLogic Integration Service Pack 5, the plug-in is licensed as part of WebLogic Integration. No further action is required.

If you install the plug-in with an earlier release of WebLogic Integration, you must obtain a valid software license and update your `license.bea` file as described in the following procedure. If you have downloaded the plug-in for evaluation, you must

obtain an evaluation license as described on the plug-in download page. If you have purchased a license for the plug-in, the license file is typically sent to you as an e-mail attachment.

When you have obtained a valid license for the plug-in, update your `license.bea` file by completing the following steps:

1. Save the license file that you obtained with a name other than `license.bea`, in the *`BEA_HOME`* directory. For example, save the file as `xfile_plugin_license.bea`. Use this file as the *`license_update_file`* in step 4 of this procedure.

   Warning: Do not overwrite or change the name of the existing `license.bea` file.

2. Perform the step appropriate for your platform:

   - On a Windows system, open an MS-DOS session and go to the *`BEA_HOME`* directory.
   - On a UNIX system, go to the *`BEA_HOME`* directory.

3. If it is not already included, add the JDK to your `PATH` variable by executing the command appropriate to your system:

   - On a Windows system:

     ```
     set PATH=BEA_HOME\jdk131_03\bin;%PATH%
     ```

   - On a UNIX system:

     ```
     PATH=BEA_HOME/jdk131_03/bin:$PATH
     export PATH
     ```

4. Merge the license update file into your existing license by executing the command appropriate to your system:

   - On a Windows system:

     ```
     UpdateLicense license_update_file
     ```

   - On a UNIX system:

     ```
     sh UpdateLicense.sh license_update_file
     ```

   Here, *`license_update_file`* is the name you gave the license update file in step 1.

5. Save a copy of your updated `license.bea` file in a safe place outside the WebLogic Integration and application installation directories.

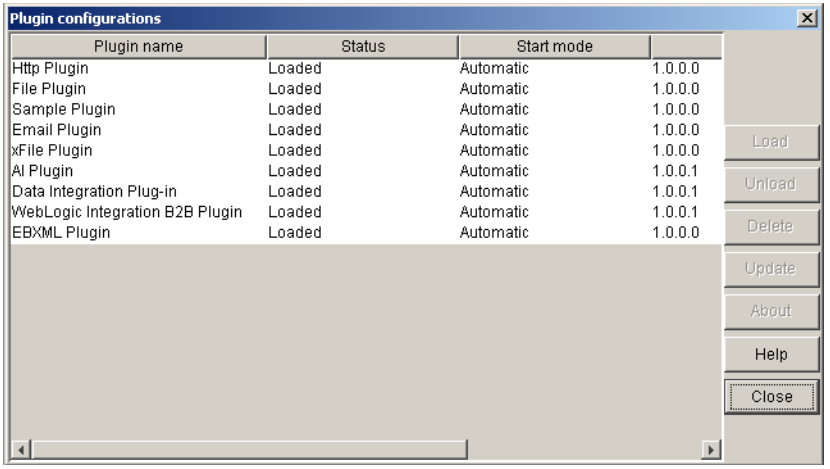
Verifying Deployment

After you have completed the steps for deploying the xFile Plug-in, you need to verify whether it has been deployed correctly.

To verify the deployment, do the following:

1. Open WebLogic Integration Studio.
2. Choose Configuration→Plug-ins.The Plug-in Configurations dialog box is displayed.

Figure 2-1 Plug-in Configurations Dialog Box



3. In the Plug-in Configurations dialog box, under Plug-in names, locate xFile Plug-in. Its presence confirms that the xFile Plug-in has been deployed correctly.

CHAPTER

3 Using the xFile Plug-in

This section provides information on using the xFile Plug-in. It contains the following topics:

These topics contain step-by-step instructions for setting up the xFile Plug-in actions. It is assumed that you already know how to design WebLogic Integration workflows. For an example of using the xFile Plug-in to transfer a file to a remote server, see Chapter 5, "xFile Plug-in Example."

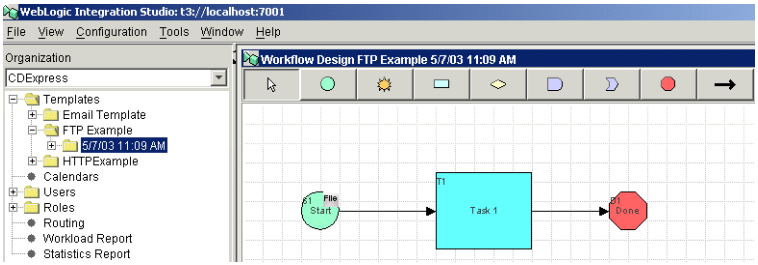
Overview

This topic provides information you need to know before using the xFile Plug-in services. It contains the following sub-topics:

- Defining Workflow Variables
- Setting Task Properties
- Workflow Expressions

You can set up the xFile Plug-in using the Workflow Design window in the WebLogic Integration Studio, as shown in the following figure.

Figure 3-1 WebLogic Integration Studio - Workflow Design Window



In the Workflow Design window, you can construct workflows and set properties that define workflows. You must set the variable properties and task properties.

Defining Workflow Variables

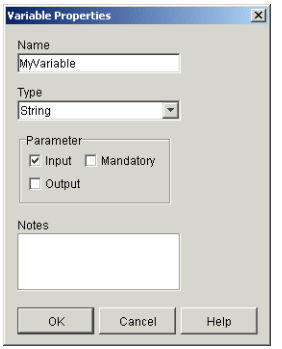
Some of the xFile Plug-in actions may require variables to hold file content. You must define workflow variables before defining the workflow's task properties, because the file content is bound to a workflow variable.

To define the variables used by the workflow actions:

1. In the left pane of the WebLogic Integration Studio, double-click the Templates folder, right-click the Variables node, and select Create Variable. The Variable Properties dialog box is displayed.

   Note: You can create variables only for existing templates. To know how to create templates and template definitions, see "Setting Up the Workflow."

Figure 3-2 Variable Properties Dialog Box



2. Set the following properties:

Table 3-1 Variable Properties

Field	Description	Example
Variable Name	Enter the name of the variable.	`MyVariable`
Type of Variable	Select the type of variable from the drop-down list.	`String`
Parameter	Select the relevant check box(es), depending upon the purpose: ■ Input - To create an input variable ■ Output- To create an output variable ■ Mandatory - To make the variable mandatory	

3. Click OK. The Variable Properties dialog box closes.

For more information on variables, see the "Working with Variables" section of "Defining Workflow Templates" in *Using the WebLogic Integration Studio* at http://edocs.bea.com/wli/docs70/studio/ch5.htm.

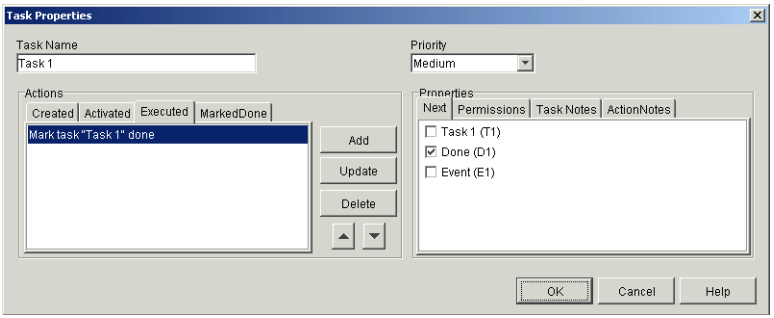
Setting Task Properties

You set the Task Properties on the Task node in the Workflow Design window.

To set the workflow properties for a task:

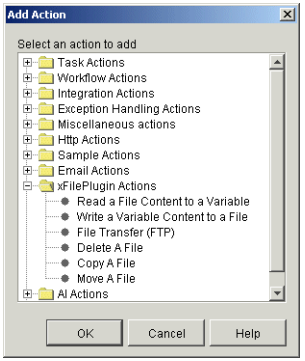
1. In the Workflow Design window, right-click a Task node, and select Properties. The Task Properties dialog box is displayed.
2. Select the Executed tab, as shown in the following figure.

Figure 3-3 Task Properties Dialog Box



3. To add an xFile Plug-in action, click Add. The Add Action dialog box is displayed.

Figure 3-4 Add Action Dialog Box



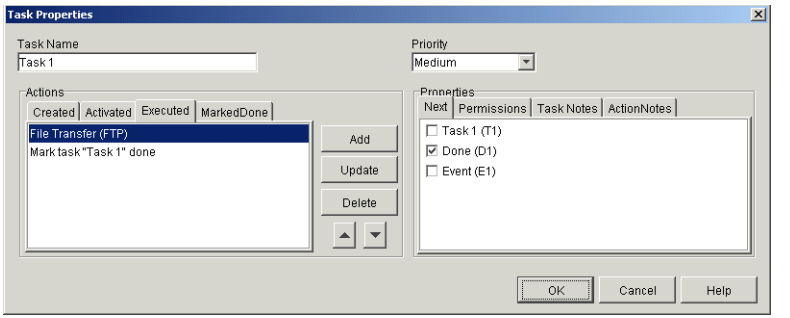
4. Double-click the xFile Plug-in Actions folder, select the required xFile Plug-in action, and click OK. The corresponding dialog box, to define the properties of the action, is displayed.

Note: For details on defining the properties for the selected Plug-in action, see the sections that follow.

5. After defining the properties, click OK. The Task Properties dialog box is displayed, with the Plug-in action displayed on the Executed tab.

6. To make the displayed action the first to be executed, select the action, and click the Up arrow. The action moves to the top position, as shown in the following figure.

Figure 3-5 Task Properties Dialog Box with Selected Action



Note: Use the Up and Down arrows to move selected tasks higher or lower in the list, depending upon the workflow requirements. You can use this option when there are multiple tasks that require a sequence for their execution.

7. Click OK. The Task Properties dialog box closes.

Workflow Expressions

A workflow expression is an algebraic expression that defines a calculation that the system performs at run time, and is made up of literals, such as strings, integers and other constants, workflow variables, operators, and workflow functions. Workflow expression syntax allows you to manipulate strings, test for relationships and conditions, perform arithmetic calculations, use functions that obtain run-time information from workflows or `XML` messages, and so on.

The result of an expression may be a string, integer, double, date/time value, or either of the Boolean (logical) values true and false. Expressions that yield a Boolean result are referred to as conditional expressions or conditions.

The Expression button  next to a field in a Studio dialog box means that the field requires that an entry be formulated in the workflow expression language. You can either enter an expression within quotes in the field, or click the Expression button to formulate the expression.

In the xFile Plug-in actions, the following properties require expressions:

- File Name
- Source File Name
- Destination File Name
- Directory Name
- Time Interval - Must use an integer
- Archive Directory
- FTP Server Name
- Remote File Name
- Local File Name

You can obtain values for these properties from the Expression Builder and XPath Wizard, which return a string value. The following values are available:

- Constant strings, such as `"d:\\read\\read.xml"`
- Workflow variables in String type, such as `$file_name`
- Complex expressions that return string values, such as `$a+$b+$c+"a.txt"`

Notes: The string value for the file or directory name must contain the full path to the file or directory. Two formats for the path are available:

- DOS, such as `"d:\\read\\read.xml"`
- UNIX, such as `"/home/workflow/input.txt"`

For more information about expressions, see "Using Workflow Expressions" in *Using the WebLogic Integration Studio* at http://edocs.bea.com/wli/docs70/studio/index.htm.

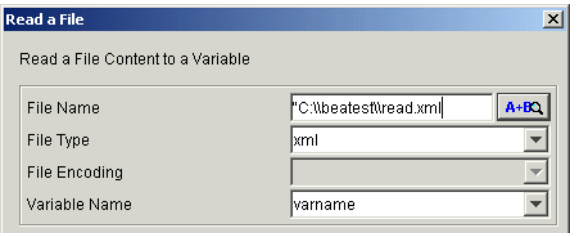
Reading a File

Using the xFile Plug-in, you can read the contents of a file and assign the contents to a workflow variable.

To read a file, do the following:

1. In the Add Action dialog box, double-click the xFile Plug-in Actions folder, and select Read a File Content to a Variable. The Read a File dialog box is displayed.

Figure 3-6 Read a File Dialog Box



2. Set the following properties:

Table 3-2 Reading a File Properties

Field Name	Description	Example
File Name	Enter the path and name of the file or click A+B to select an expression. For details, see Workflow Expressions.	"C:\\Dir\\FileTest.txt"
File Type	Select the type of file from the drop-down list. The choices are XML, text, and binary. **Note:** The file type Text corresponds to the variable type string.	XML Text Binary
File Encoding	Select the type of encoding from the drop-down list, required to read the contents of the file. **Note:** Text files are the only type that can be encoded.	Cp1252

Table 3-2 Reading a File Properties (Continued)

Field Name	Description	Example
Variable Name	Enter the variable to hold the contents of the file. **Note:** The file type `Text` corresponds to the variable type `string`.	`xmlVar`

3. Click OK. The Read a File dialog box closes and the Task Properties dialog box is displayed.

Run-Time Exceptions

The following exceptions may occur during the workflow process:

- File does not exist
- Transaction times out

Take appropriate action, based on the exception message.

Writing to a File

Using the xFile Plug-in, you can write contents of a workflow variable to a file. You can write to a file in three different ways:

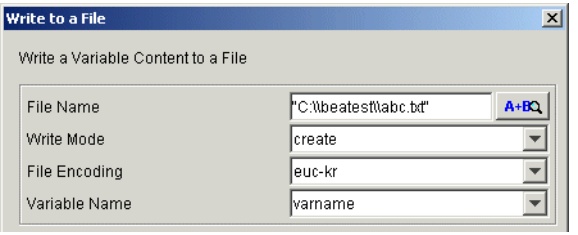
- Create a new file and write the contents of a workflow variable to it.
- Append the contents of a workflow variable to an existing file.
- Overwrite an existing file with the contents of a workflow variable.

Caution: Writing to the same file through multiple workflow instances might corrupt the data. To prevent multiple threads from modifying the same file at the same time, ensure that only one workflow instance is allowed to write to a file at any time.

To write to a file, do the following:

1. In the Add Action dialog box, double-click the xFile Plug-in Actions folder, and select Write a VariableContent to a File. The Write to a File dialog box is displayed.

Figure 3-7 Write to a File Dialog Box



2. Set the following properties:

Table 3-3 Writing to a File Properties

Field Name	Description	Example
File Name	Enter the path and name of the file or click A+B to select an expression. For details, see Workflow Expressions.	"C:\\Dir\\FileTest.txt"
Write Mode	Select the Mode from the drop-down list. ■ Select create to write contents to a new file. ■ Select append to append the contents to an existing file. ■ Select overwrite to replace the contents of the file.	create append overwrite
File Encoding	Select the type of encoding from the drop-down list, required to write contents to the file. **Note:** Write action supports encoding for `text` and `xml` files.	`Cp1252`
Variable Name	Enter the name of the variable that holds the contents of the file.	`strVar`

3. Click OK. The Write to a File dialog box closes, and the Task Properties dialog box is displayed.

Run-Time Exceptions

Exceptions may occur in the following conditions:

- If you create a file with a filename that already exists.
- If the transaction times out.

Take appropriate action, based on the exception message.

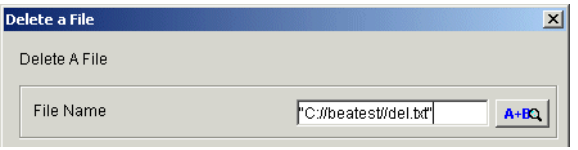
Deleting a File

Using the xFile Plug-in, you can delete a file from a file system.

To delete a file, do the following:

1. In the Add Action dialog box, double-click the xFile Plug-in Actions folder, and select Delete a File. The Delete a File dialog box is displayed.

Figure 3-8 Delete a File Dialog Box



2. In the File Name field, specify the path and name of the file that you want to delete, or click A+B to select an expression (for details, see Workflow Expressions).

3. Click OK. The Delete a File dialog box closes and the Task Properties dialog box is displayed.

Run-Time Exception

An exception occurs if the specified file exists, but delete permissions are restricted.Take appropriate action, based on the exception message.

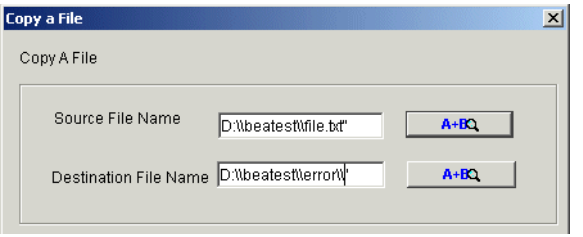
Copying a File

Using the xFile Plug-in, you can copy a file from one location to another. When you copy the file, you can either rename it or append a timestamp to the existing file name.

To copy a file, do the following:

1. In the Add Action dialog box, double-click the xFile Plug-in Actions folder, and select Copy a File. The Copy a File dialog box is displayed.

Figure 3-9 Copy a File Dialog Box



2. Set the following properties:

Table 3-4 Copying a File Properties

Field Name	Description	Example
Source File Name	Enter the path and name of the source file or click A+B to select an expression.	"C:\\Dir\\FileTest.txt"

Table 3-4 Copying a File Properties (Continued)

Field Name	Description	Example
Destination File Name	Enter the path and name of the destination file or click  to select an expression. For details, see Workflow Expressions. ■ To copy the file and append a timestamp to the existing file name, specify just the destination path. ■ To copy the file with a new name, specify both the destination path and the new file name.	To append a timestamp to the existing file name, enter: "D:\\Dir" To rename a file, enter: "D:\\Dir\\FileTest2.txt" Renames the file.

3. Click OK. The Copy a File dialog box closes, and the Task Properties dialog box is displayed.

Run-Time Exceptions

Exceptions may occur in the following conditions:

- If the file to be copied does not exist in the source directory.
- If the file to be copied already exists in the destination directory (with the same name).

Take appropriate action, based on the exception message.

Moving a File

Using the xFile Plug-in, you can move a file from one location to another. When you move the file, you can either rename it or append a timestamp to the existing file name. This allows you to:

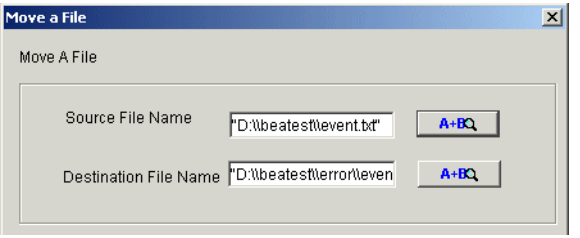
- Archive a file with a timestamp appended to its original name.

- Rename a file and place it at a different location.

To move a file, do the following:

1. In the Add Action dialog box, double-click the xFile Plug-in Actions folder, and select Move a File. The Move a File dialog box is displayed.

Figure 3-10 Move a File Dialog Box



2. Set the following properties:

Table 3-5 Moving a File Properties

Field Name	Description	Example
Source File Name	Enter the path and name of the source file or click A+B to select an expression.	"C:\\Dir\\FileTest.txt"
Destination File Name	Enter the path and name of the destination file or click A+B to select an expression. For details, see Workflow Expressions. ■ To move the file and append a timestamp to the existing file name, specify just the destination path. ■ To move the file with a new name, specify both the destination path and the new file name.	To move a file appended with a timestamp, enter: "D:\\Dir" To rename and move a file, enter: "D:\\Dir\\FileTest2.txt"

3. Click OK. The Move a File dialog box closes, and the Task Properties dialog box is displayed.

Run-Time Exceptions

Exceptions may occur in the following conditions:

- If the file to be moved does not exist in the source directory.
- If the file to be moved already exists in the destination directory (with the same name).

Take appropriate action, based on the exception message.

Accessing a File Through File Transfer Protocol

Using the xFile Plug-in, you can access and transfer files (in `ASCII` or `binary` format) between a local system and a remote system via the File Transfer Protocol (FTP).

You can use one of the following features to access and transfer files:

- GET Files: retrieves a file from a remote system for additional processing, such as starting a workflow, triggering an event, or reading information into a workflow.
- PUT Files: transfers a file from a local system to a remote system, for archiving or triggering an application on the remote system.

Caution: A GET action overwrites a file if the same file name exists in the local system, and a PUT action overwrites a file if the same file name exists in the remote system. To prevent overwriting files, ensure that the file names are different before you GET or PUT files.

If you need to perform additional file operations or read and write Excel spreadsheet contents and Comma Separated Variable (`CSV`) formats, you may want to use the BEA WebLogic Adapter for File.

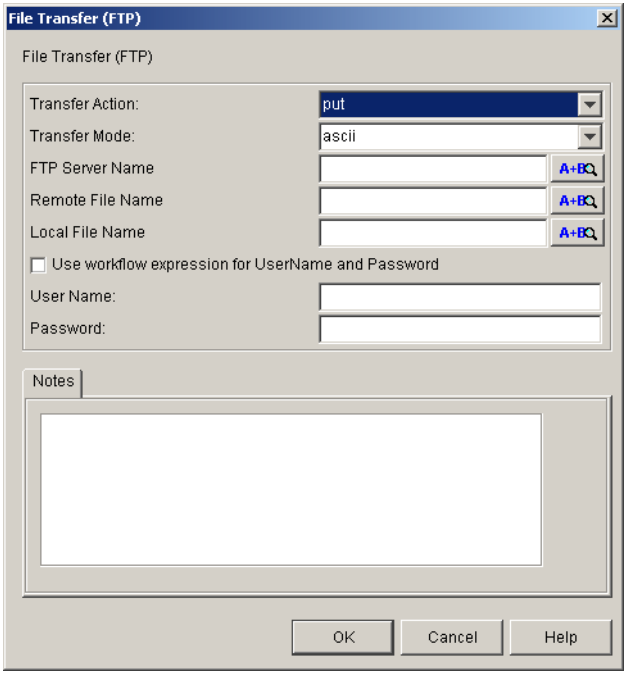
For more information about the BEA WebLogic Adapter for File, see the documentation for the adapter at the following URL:

http://edocs.bea.com/wladapters/file/docs70sp2/index.htm.

To access a file through the File Transfer Protocol (FTP), do the following:

1. In the Add Action dialog box, double-click the xFile Plug-in Actions folder, and select File Transfer (FTP). The File Transfer (FTP) dialog box is displayed.

Figure 3-11 File Transfer (FTP) Dialog Box



2. Set the following properties:

Table 3-6 Transferring a File Properties

Label	Description	Example
Transfer Action	Select the transfer action from the drop-down list.	put get
Transfer Mode	Select the mode of transfer from the drop-down list.	`ASCII` `Binary`

Table 3-6 Transferring a File Properties (Continued)

Label	Description	Example
FTP Server Name	Enter the name of the FTP Server and the port number (server:port), or click A+B to select an expression. For details, see Workflow Expressions.	`172.16.16.168:21`
Remote File Name	Enter the path and name of the file on the remote machine, or click A+B to select an expression. ■ For a GET action, this is the name of the file being retrieved *from* the remote file system. ■ For a PUT action, this is the name of the file being written *on* the remote system.	"`C:\\get\\test.txt`"
Local File Name	Enter the path and name of the file on the local machine, or click A+B to select an expression. ■ For a GET action, this is the name of the file being written *on* the local file system. ■ For a PUT action, this is the name of the local file being written *to* the remote system.	"`D:\\put\\test.txt`"
Use workflow expression for User Name and Password (check box)	Check to specify an expression for the User Name and Password. When checked, the expression builder is added to the User Name and Password fields:  **Note:** This option is available in the xFile Plug-in packaged with WebLogic Integration 7.0 SP5; it is not included in the original 7.1 release of the BEA WebLogic Integration Plug-in for xFile. Use this feature with caution. When an expression is used to specify a password, the password is exposed in clear text within the system.	Unchecked
User Name	Enter the user name needed to log on to the FTP server.	susan
Password	Enter the password needed to log on to the FTP server.	
Notes	Optional notes or description of the action.	

3. Click OK. The File Transfer (FTP) dialog box closes, and the Task Properties dialog box is displayed.

Starting a Workflow upon Detecting a File Creation

You can start a workflow whenever a file is created or moved into a specific directory. The xFile Plug-in supports these data types: `text`, `xml`, and `binary`.

The file movement that starts a workflow could be a new file arriving into a specified directory, or a file being copied from one directory to another. When a file arrives in the specified directory, it is processed according to the workflow action settings. In addition to specifying the location where files will arrive, you can also set the time interval for the system to periodically look for new files arriving in a specified directory.

The application searches for a file in a specified directory at periodic intervals, and upon detecting the file, it starts a workflow. You can set the polling interval to control the number of workflows that can be started at one time. If you are not sure of the exact file name to search for, you can specify a file mask (search criteria, which can include a wildcard) to be used instead.

If the application detects a `XML` or `Binary` file, it moves the file to the archive directory and copies its contents to the variable selected during design-time. The application cannot process individual rows for these types of files.

If the file is of type `ASCII`, you can either move it directly to the archive directory or process it first before storing it. Processing consists of:

- Specifying the number of rows to be ignored at the beginning and end of the file. These rows correspond to headers and footers and do not contain business information.
- Specifying the number of rows the system should read at one time.
- Specifying the number of documents that can be created at once. This prevents the WebLogic Integration Server from being overloaded.

The output of row processing is in the form of an XML variable. The schema for the XML variable is given below:

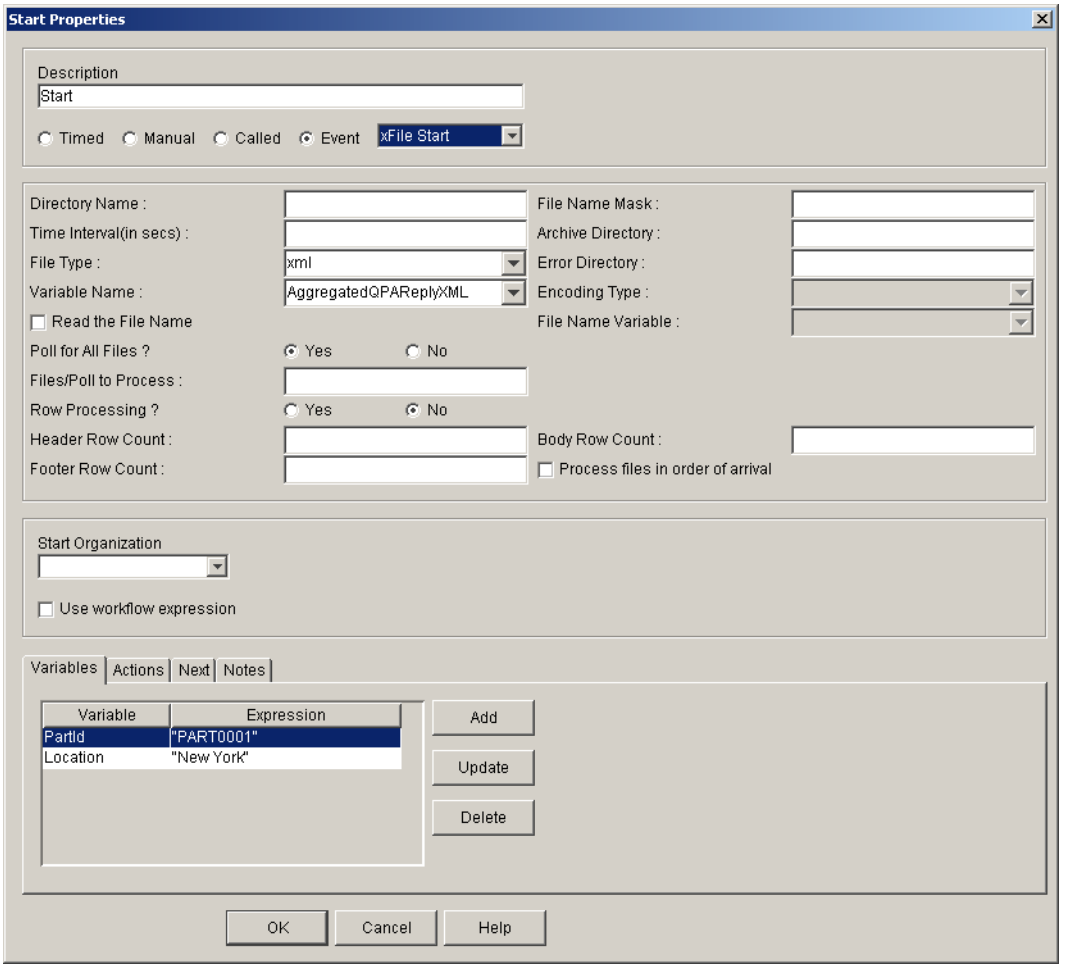
```
<?xml version=encoding="1.0" encoding="UTF-8"?>
<FILE>
   <HEADER>Header Content</HEADER>
   <BODY>Body Content
   Body Content</BODY>
   <FOOTER>Footer Content</FOOTER>
</FILE>
```

If the application encounters errors in reading the file, it stores the file in the error directory.

To define the properties of the Start node, do the following:

1. In the Workflow Design window, right-click the Start node, and select Properties. The Start Properties dialog box is displayed.
2. Click the Event option button, and from the Event drop-down list, select xFile Start. The fields pertaining to the xFile Start Properties appear, as shown in the following figure.

Figure 3-12 Start Properties Dialog Box



3. Set the following properties:

Table 3-7 Start Properties

Field Name	Description	Example
Directory Name	Enter the directory path.	"D:\\beatest"
File Name Mask	Enter the file name or the mask of the file. **Note:** A file mask is an extension, by which a file is distinguished.	cop10MB.txt *.xml or *.txt
Time Interval	Enter the polling interval in integer seconds.	120 30
Archive Directory	Enter the path of the archive directory.	"C:\\archive"
File Type	Select the type of file from the drop-down list.	Text XML Binary
Error Directory	Enter the path and name of the directory where you want to save the file if there is an error.	"D:\\error"
Variable Name	Select the name of the variable to hold the content from the drop-down list.	xmlVar
Read the File Name (check box)	Check to read the file name. When checked, the File Name Variable drop-down list is active to allow you to select the variable to hold the file name. **Note:** This option is available in the xFile Plug-in packaged with WebLogic Integration 7.0 SP5; it is not included in the original 7.1 release of the BEA WebLogic Integration Plug-in for xFile.	
File Name Variable	Select the name of the variable to hold the file name. The drop-down list is only active when the Read the File Name option is checked.	MyFileNameVar
Encoding Type	Select the type of encoding required to process the contents of the file from the drop-down list. **Note:** File Start supports encoding for text files only.	Cp1252

Table 3-7 Start Properties (Continued)

Field Name	Description	Example
Poll for All Files?	Select Yes to poll all files. Select No to poll only the number of files you will specify in Files/Poll to Process.	Yes or No
Files/Poll to Process	Enter the number of files to be polled. Enter this information only if Poll for All Files is set to No.	10
Row Processing?	Select Yes or No. **Note:** You can select Yes only if the File Type is `Text`.	Yes or No
Header Row Count	Enter the number of header rows to be processed. Enter this information only if Row Processing is set to Yes.	2
Footer Row Count	Enter the number of footer rows to be processed. Enter this information only if Row Processing is set to Yes.	2
Body Row Count	Enter the number of rows to be processed in the body of the file. Enter this information only if Row Processing is set to Yes.	10
Process files in order of arrival	Select the check box if the files are to be processed in the order of arrival (creation). Leaving it unchecked will process the files in random.	

4. From the Start Organization list, select the relevant organization, and click OK. The Start Properties dialog box closes.

   On the Workflow Design window, the Start Node indicates the action setting, as shown in the following figure.

Figure 3-13 Start Node with File Action

Resuming a Workflow upon Detecting a File

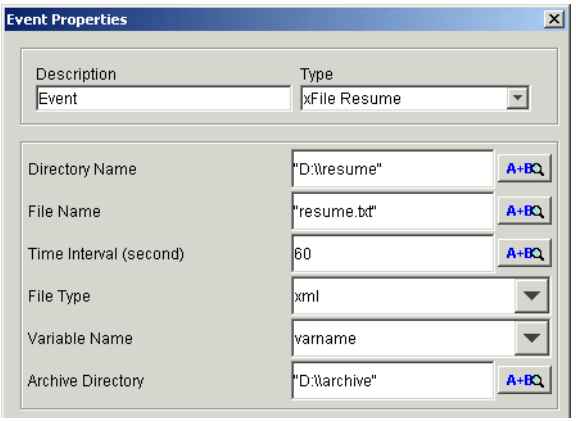
You can set an Event node so that when a workflow instance reaches that node, the workflow checks for the existence of a specified file. If the file exists, the workflow reads the contents of the file and moves on to the next Event node. If the file does not exist, the workflow pauses and polls for the specified file until the file is created. After the file is detected, it is stored in the archive directory.

Note: You must define a workflow variable before defining the workflow's Event properties because the file content is bound to a workflow variable.

To define the properties of the Event node, do the following:

1. In the Workflow Design window, right-click the Event node, and select Properties. The Event Properties dialog box is displayed.
2. From the Event Type drop-down list, select xFile Resume. The fields pertaining to the xFile Event Properties appear, as shown in the following figure.

Figure 3-14 Event Properties Dialog Box



3. Set the following properties:

Table 3-8 Defining Event Properties

Field Name	Description	Example
Directory Name	Enter the path and name of the directory or click A+B to select an expression. For details, see Workflow Expressions.	"D:\\Start\\workflow\\"
File Name	Enter the name of the file or click A+B to select an expression.	"resume.txt"
Time Interval (second)	Enter the polling interval in seconds (integer value) or click A+B to select an expression.	"120"
File Type	Select the type of file from the drop-down list.	Text XML Binary
Variable Name	Select the name of the variable to hold the content, from the drop-down list.	varname
Archive Directory	Enter the path of the archive directory or click A+B to select an expression. Entry in this field is optional.	"C:\\archive"

4. Click OK. The Event Properties dialog box closes.

   On the Workflow Design window, the Event Node indicates the action setting, as shown in the following figure.

Figure 3-15 Event Node with File Action

CHAPTER

4 Configuring the xFile Plug-in for a Migrated Domain

This section describes how to update your database schema and configure the xFile Plug-in for a single server domain and a cluster domain. It contains the following topics:

- Updating the Business Process Management Database Table
- Migrating to a Single Server Domain
- Migrating to a Clustered Domain

Updating the Business Process Management Database Table

The xFile Plug-in uses a new database table called `FILEPOLLX`. To update the Business Process Management (BPM) database with this table, edit the following file by appending the contents of `fp_schema.sql`:

```
BEA_HOME\weblogic700\integration\dbscripts\<database_type>\
migrate\BPM_70-70SP2.sql
```

The contents of `fp_schema.sql` will depend on your database.

Migrating to a Single Server Domain

To configure the xFile Plug-in for a single server domain, copy the `.jar` file to your WebLogic installation directory and edit the domain's `config.xml` file as shown below.

1. To deploy the `fileplugin-ejb.jar` as one component of WebLogic Integration application, add the following:

```
<EJBComponent Name="WLI-BPM xFile Plug-in"
Targets="<Customer_localhost>" URI="xfileplugin-ejb.jar"/>
```

2. To add the JMS queue for the xFile Plug-in, add the following:

```
<JMSQueue JNDIName="com.bea.wli.bpm.xFilePluginQueue"
Name="WLI_BPM_FP" Template="WLI_JMSTemplate"/>
```

Migrating to a Clustered Domain

To configure the xFile Plug-in for a single server domain, copy the `.jar` file to your WebLogic installation directory and edit the domain's `config.xml` file as shown below.

Note: This example shows a cluster system (`mycluster`) with one manager server (`myserver`) and two managed servers (`c1`, `c2`).

1. To deploy `fileplugin-ejb.jar` on cluster server, add the following:

```
<EJBComponent Name="WLI BPM xFile Plug-in"
Targets="mycluster" URI="xfileplugin-ejb.jar"/>
```

2. To configure the JMS Queue for the xFile Plug-in, add the following:

```
<JMSDistributedQueue Name="WLI_BPM_FP"

   JNDIName="com.bea.wli.bpm.FilePluginQueue"
   Targets="mycluster">

<JMSDistributedQueueMember Name="WLI_BPM_FP-c1"

   JMSQueue="WLI_BPM_FP-c1" Weight="1"/>
```

```
<JMSDistributedQueueMember Name="WLI_BPM_FP-c2"

   JMSQueue="WLI_BPM_FP-c2" Weight="1"/>

   <JMSTemplate Name="WLI_BPM_FP"/>

</JMSDistributedQueue>
```

3. On the CLUSTER NODE C1 JMS SERVER, add the following:

```
<JMSQueue Name="WLI_BPM_FP-c1"

   JNDIName="com.bea.wli.bpm.xFilePluginQueue-c1"
   StoreEnabled="true"
   Template="WLI_JMSTemplate-c1"/>
```

4. On the CLUSTER NODE C2 JMS SERVER, add the following:

```
<JMSQueue Name="WLI_BPM_FP-c2"

   JNDIName="com.bea.wli.bpm.xFilePluginQueue-c2"
   StoreEnabled="true"
   Template="WLI_JMSTemplate-c2"/>
```

5 xFile Plug-in Example

This example illustrates how to use the xFile Plug-in to transfer a file to a remote server, using FTP. It contains the following topics:

- Setting Up the Workflow
- Executing the Workflow

The first part of the example tells you how to set up a workflow in the WebLogic Integration Studio, which when executed, will transfer a file to a remote server. The second part of the example tells you how to use the BEA WebLogic Integration Worklist to execute the workflow, resulting in transferring the file.

Note: This example uses the WebLogic Integration Samples Domain. For more information, see the "Configuring and Starting the Samples Domain" section of "Getting Started" in *Starting, Stopping, and Customizing BEA WebLogic Integration* at http://edocs.bea.com/wli/docs70/config/getstart.htm.

Setting Up the Workflow

To set up a workflow in the WebLogic Integration Studio to transfer a file using FTP, do the following:

1. Start the WebLogic Integration Server.
2. To open the WebLogic Integration Studio, do one of the following:
   - On Windows, for WebLogic Integration 7.0, select Start→Programs→BEA WebLogic Platform 7.0→WebLogic Integration 7.0→Studio.

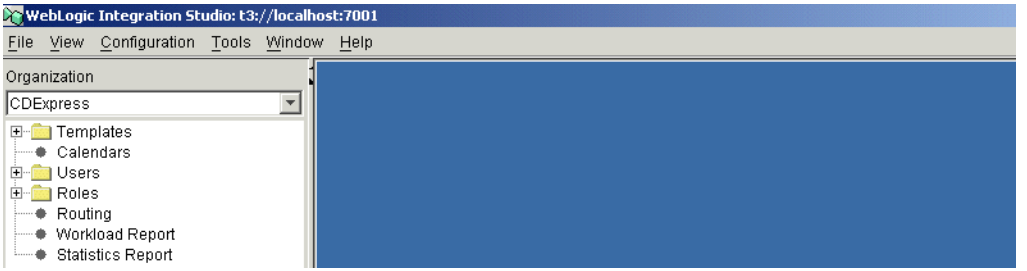
- On Windows, for WebLogic Integration 2.1, select Start→Programs→BEA WebLogic E-Business Platform→Weblogic Integration 2.1→Studio.
- On UNIX, for WebLogic Integration 7.0, run

  BEA_HOME/weblogic700/integration/bin/studio.sh
- On UNIX, for WebLogic Integration 2.1, run
  BEA_HOME/wlintegration2.1/bin/studio.sh

The Logon to WebLogic Integration dialog box is displayed.

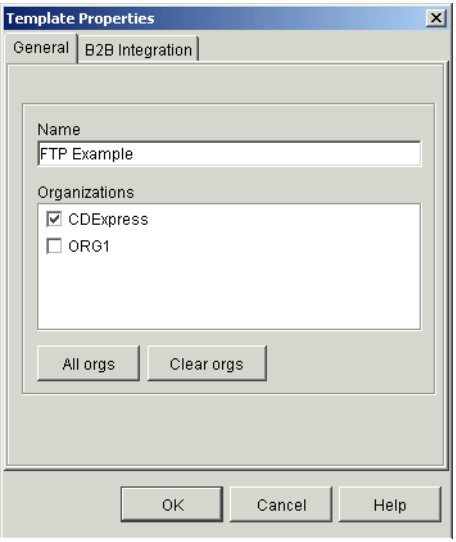
3. Enter the User Name, Password, and Server URL, and click OK. You are connected to the WebLogic Server and WebLogic Integration Studio is displayed.

Figure 5-1 WebLogic Integration Studio



4. In the left pane of WebLogic Integration Studio, from the Organizaton drop-down list, select CDExpress.
5. In the left pane, right-click the Templates folder and select Create Template. The Template Properties dialog box is displayed, as shown in the following figure.

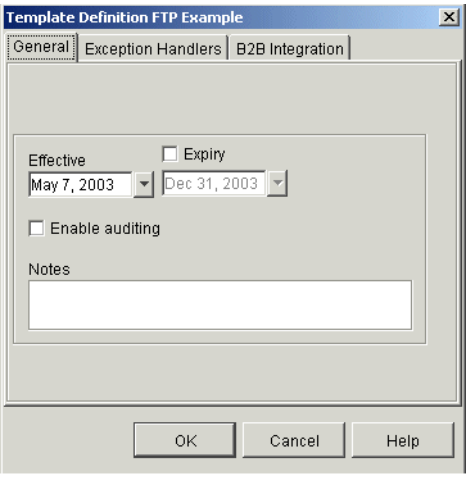
Figure 5-2 Template Properties Dialog Box



6. On the General tab, in the Name field, enter a name for the template (example - FTP Example) and click OK. The Template Properties dialog box closes, and the new template (FTP Example) is displayed in the Templates folder, on the left pane of the WebLogic Integration Studio.

7. Double-click the Template folder, right-click the newly created Template (FTP Example), and select Create Template Definition. The Template Definition FTP Example dialog box is displayed, as shown in the following figure.

Figure 5-3 Template Definition FTP Example Dialog Box



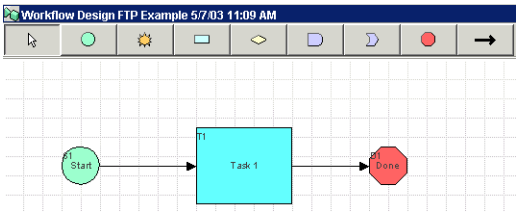
8. Do one of the following:

   - If you want to specify the expiry date for the workflow, select the Expiry check box, and from the drop-down calendar, select the desired date. Click OK.
   - If you want to retain the default expiry date, click OK.

   The Template Definition is created inside the newly created Template (FTP Example) folder, displaying the creation date and time.

   The Workflow Design window is displayed in the right pane.

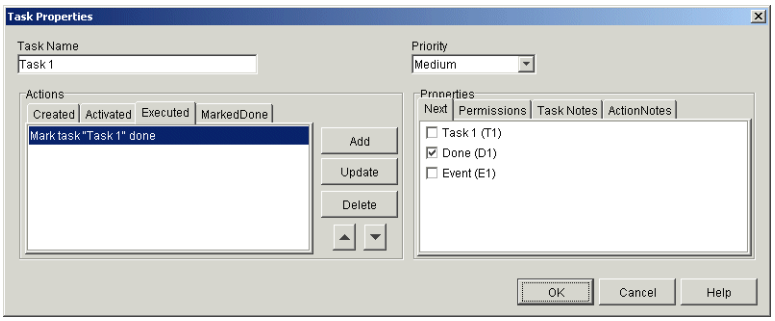
Figure 5-4 Workflow Design – FTP Example Window



9. Right-click Task 1 and select Properties. The Task Properties dialog box is displayed.

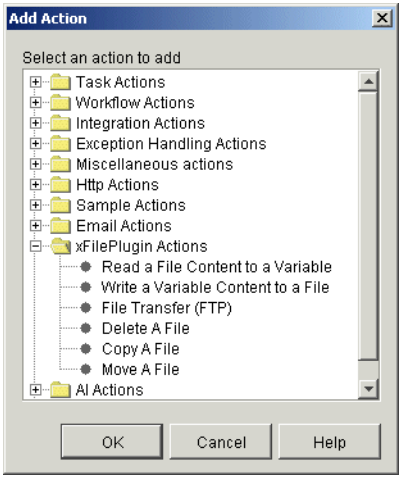
10. Select the Executed tab, as shown in the following figure.

Figure 5-5 Task Properties Dialog Box



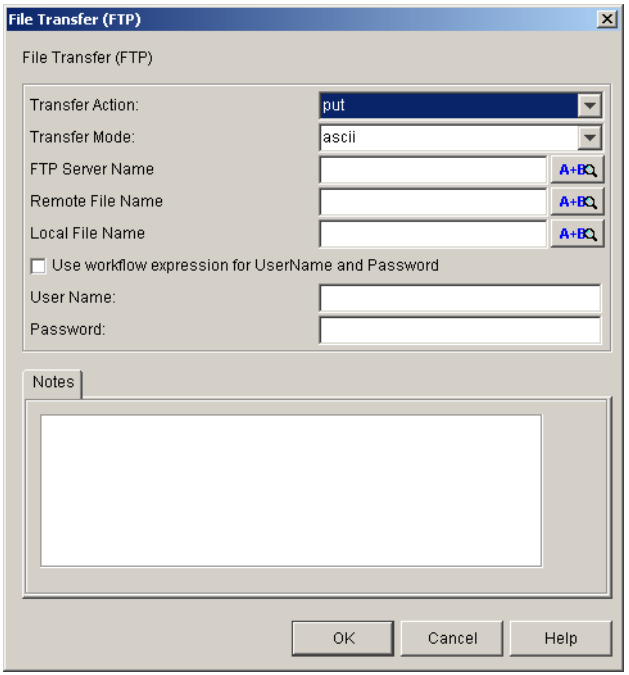
11. Click Add. The Add Action dialog box is displayed.

Figure 5-6 Add Action Dialog Box



12. Double-click the xFilePlug-in Actions folder, select File Transfer (FTP) and click OK. The File Transfer (FTP) dialog box is displayed.

Figure 5-7 File Transfer (FTP) Dialog Box



13. Set the following properties:

Table 5-1 Transferring a File Properties

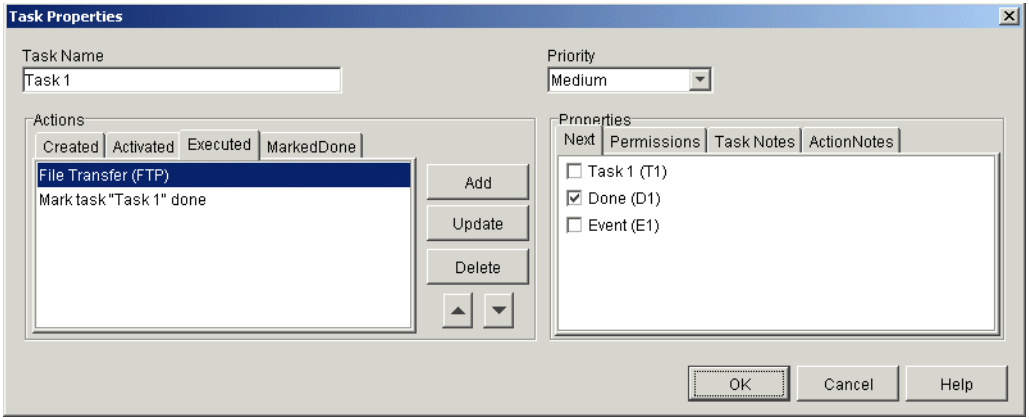
Field Name	Description	Example
Transfer Action	Select the transfer action from the drop-down list.	put
Transfer Mode	Select the mode of transfer from the drop-down list.	`ASCII` `Binary`
FTP Server Name	Enter the name of the FTP Server and the port number. The format is server:port.	`172.16.16.168:21`
Remote File Name	Enter the path and name of the file on the remote machine.	"`C:\\receive\\gettest.txt`"
Local File Name	Enter the path and name of the file on the local machine.	"`D:\\put\\test.txt`"

Table 5-1 Transferring a File Properties (Continued)

Field Name	Description	Example
Use workflow expression for User Name and Password (check box)	Check to specify an expression for the User Name and Password. When checked, the expression builder is added to the User Name and Password fields: **Note:** This option is available in the xFile Plug-in packaged with WebLogic Integration 7.0 SP5; it is not included in the original 7.1 release of the BEA WebLogic Integration Plug-in for xFile.	Unchecked
User Name	Enter the name of the user (needed to log on to the FTP server).	susan
Password	Enter the password (needed to log on to the FTP server).	

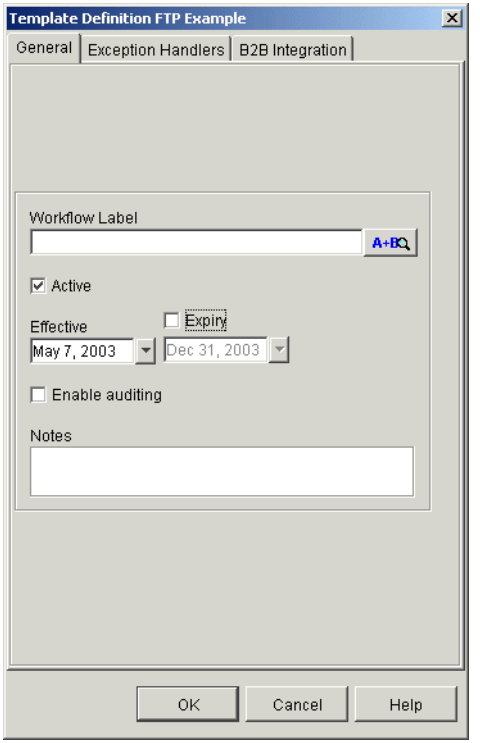
14. Click OK.The File Transfer (FTP) dialog box closes, and the Task Properties dialog box is displayed.
15. To make the displayed action the first to be executed, select the action, and click the Up arrow. The action moves to the top position, as shown in the following figure.

Figure 5-8 Task Properties Dialog Box with File Transfer Action



16. Click OK. The Task Properties dialog box closes.
17. In the left pane of WebLogic Integration Studio, right-click the Template Definition and select Properties. The Template Definition dialog box is displayed.

Figure 5-9 Template Definition FTP Example Dialog Box with Active Check Box



18. Select the Active check box, and click OK. The Template Definition dialog box closes.

Caution: To make changes to the template, do the following:

a. Open the Template Definition dialog box, deselect the Active check box and click OK. The Template Definition dialog box closes.

b. Open the Template Definition dialog box again, make changes to the template, select the Active check box, and click OK. The Template Definition dialog box closes.

This procedure is necessary to ensure that there is only one workflow process per active template at any given time.

19. In the left pane of WebLogic Integration Studio, right-click the Template Definition and select Save.

Note: An asterisk before the definition name indicates that the changes to that folder have not been saved.

20. In your file system, create a text file corresponding to the Local File Name you entered in the File Transfer (FTP) dialog box in step13.

For more information about using WebLogic Integration Studio, see *Learning to Use BEA WebLogic Integration* at http://edocs.bea.com/wli/docs70/bpmtutor/index.htm.

Executing the Workflow

In this part of the example, the WebLogic Integration Worklist executes the workflow, and transfers the text file to the remote file system. This part of the example provides information on the following:

- Executing on WebLogic Integration 7.0
- Executing on WebLogic Integration 2.1

For more information about the WebLogic Integration Worklist, see*Using the WebLogic Integration JSP Worklist* at http://edocs.bea.com/wli/docs70/jspwlist/index.htm.

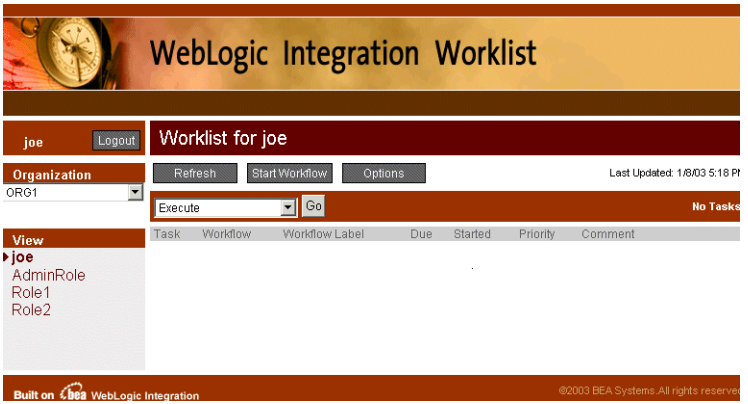
Executing on WebLogic Integration 7.0

1. To start the WebLogic Integration Worklist, do one of the following:
   - On a Windows system, select Start→Programs→BEA WebLogic Platform 7.0→WebLogic Integration 7.0→Worklist.
   - On a UNIX system, open a browser and enter the following URL:

     `http://<localhost>:<port_number>/worklist`

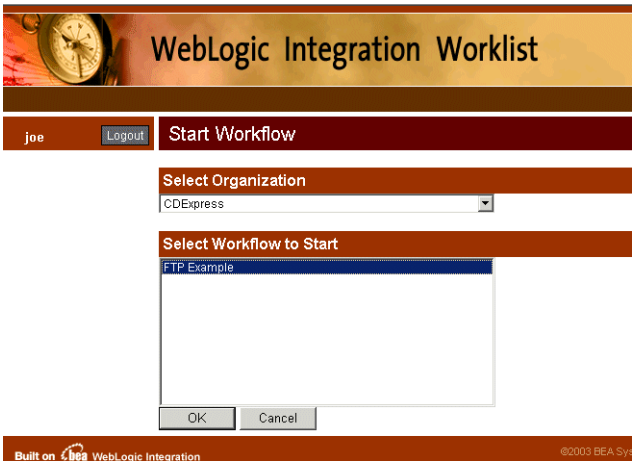
   The WebLogic Integration Worklist Login window is displayed.
2. Enter the User Name and Password, and click OK. The WebLogic Integration Worklist window is displayed.

Figure 5-10 WebLogic Integration Worklist Window



3. From the Organization drop down list, select CDExpress, and click Start Workflow. The WebLogic Integration Worklist - Start Workflow window is displayed.

Figure 5-11 WebLogic Integration Worklist - Start Workflow Window



4. In the Select Workflow to Start list, select FTP Example, and click OK. The WebLogic Integration Worklist window is displayed, displaying the task, as shown in the following figure.

Figure 5-12 WebLogic Integration Worklist Window with Task

5. At the end of the task row, click Execute. The file is transferred to its target directory and the task disappears from the worklist.

6. Check your target directory to see the transferred file.

Executing on WebLogic Integration 2.1

1. To start the WebLogic Integration Worklist, do one of the following:

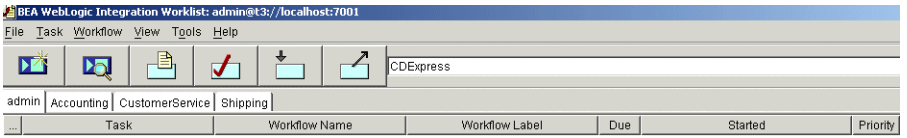
   - On a Windows system, select Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Worklist.
   - On a UNIX system, run

     ```
     BEA_HOME/wlintegration2.1/bin/worklist
     ```

     The Logon to WebLogic Integration dialog box is displayed.

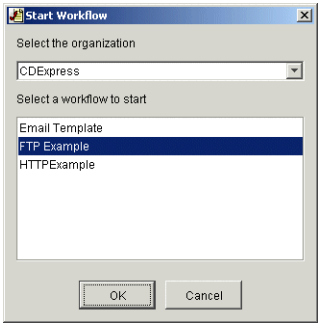
2. Enter the User Name, Password, and Server URL, and click OK. The WebLogic Integration Worklist window is displayed.

Figure 5-13 WebLogic Integration Worklist Window



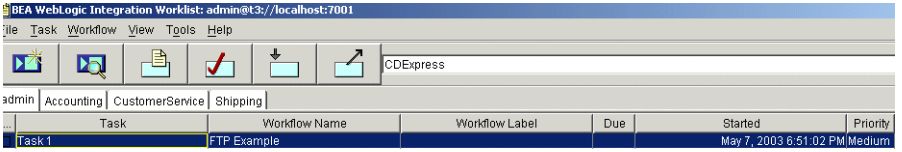
3. From the Organization drop down list, select CDExpress.
4. Choose Workflow→Start a Workflow. The Start Workflow dialog box is displayed.

Figure 5-14 Start Workflow Dialog Box



5. In the Select a Workflow to Start list, select FTP Example, and click OK. The WebLogic Integration Worklist window is displayed, displaying the task, as shown in the following figure.

Figure 5-15 WebLogic Integration Worklist Window with Task



6. Right-click the task and select Execute. The file is transferred to its target directory and the task disappears from the worklist.
7. Check your target directory to see the transferred file.

Index