



BEA WebLogic Integration™

Using the File Plug-In

Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Using the File Plug-In

Part Number	Date	Software Version
N/A	February 2003	7.0 SP2

Contents

About This Document

What You Need to Know	v
e-docs Web Site	v
How to Print the Document	vi
Related Information	vi
Contact Us!	vi
Documentation Conventions	vii

1. Using the File Plug-In

About the File Plug-In	1-1
What the File Plug-In Does	1-2
Using the File Plug-In or Adapter for File?	1-2
What You Need to Know	1-3
File Plug-In and the Plug-in Framework	1-3
Using the File Plug-In	1-4
Overview of File Plug-In Actions	1-4
Workflow Expressions	1-8
Reading Files	1-9
Template Definitions	1-9
Write File	1-10
Delete File	1-11
Resume Workflow on File Detection	1-12
Start Workflow on Detection of File Creation	1-14
Get and Put Files Using FTP	1-16
Configuring File Plug-In for a Migrated Domain	1-18
Updating the BPM Database Table	1-18
Migrating to a Single Server Domain	1-19

Migrating to a Clustered Domain..... 1-19

2. An Example of Using the File Plug-In

Setting Up the Workflow..... 2-1

Activating the Workflow..... 2-8

Index

About This Document

This document explains how to use the File Plug-In.

The document is organized as follows:

- Chapter 1, “Using the File Plug-In,” provides information for using the File Plug-In.
- Chapter 2, “An Example of Using the File Plug-In,” provides a step-by-step example for using the File Plug-In to FTP 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.

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://e-docs.beasys.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 WebLogic Integration documents contain information that is relevant to using this product.

- *Learning to Use BPM with WebLogic Integration*
- *Using the WebLogic Integration Studio*

Contact Us!

Your feedback on the 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 which release of the WebLogic Integration documentation you are using.

If you have any questions about this version of WebLogic Integration Worklist, or if you have problems installing and running the product, contact BEA Customer Support through BEA WebSUPPORT at www.beasys.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
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	Indicates emphasis or book titles.

Convention	Item
monospace text	<p>Indicates code samples, commands and their options, data structures and their members, data types, directories, and filenames and their extensions. Monospace text also indicates text that you must enter from the keyboard.</p> <p><i>Examples:</i></p> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
<i>monospace italic text</i>	<p>Identifies variables in code.</p> <p><i>Example:</i></p> <pre>String expr</pre>
UPPERCASE TEXT	<p>Indicates device names, environment variables, and logical operators.</p> <p><i>Examples:</i></p> <pre>LPT1 SIGNON OR</pre>
{ }	<p>Indicates a set of choices in a syntax line. The braces themselves should never be typed.</p>
[]	<p>Indicates optional items in a syntax line. The brackets themselves should never be typed.</p> <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
	<p>Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.</p>

Convention	Item
...	Indicates one of the following in a command line: <ul style="list-style-type: none">■ 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. <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
.	Indicates the omission of items from a code example or from a syntax line.
.	The vertical ellipsis itself should never be typed.
.	



1 Using the File Plug-In

The File Plug-In extends the functionality of BEA WebLogic Integration Business Process Management (BPM). It provides additional functionality to the WebLogic Integration BPM Studio. Specifically, the File Plug-In provides basic actions for working with files and transferring files between local and remote file systems. Actions are the basic unit of work performed during business process management functions; they define workflow and task behavior.

This section includes the following information:

- About the File Plug-In
- Using the File Plug-In
- Configuring File Plug-In for a Migrated Domain

About the File Plug-In

This section includes information on the following topics:

- What the File Plug-In Does
- Using the File Plug-In or Adapter for File?
- What You Need to Know
- File Plug-In and the Plug-in Framework

What the File Plug-In Does

The File Plug-In has the following functionality:

- Read files and assign the content to workflow variables.
- Write the content of a workflow variable to a file.
- Delete files.
- Start or resume a workflow based on the creation of a file.
- Transfer files to and from a remote file system via FTP.

When using the File Plug-In, you should keep the following in mind:

- The File Plug-In does not have a file locking service. You should prevent multiple threads from modifying the same file at the same time.
- The File Plug-In actions are not transactional and do not have a built-in mechanism to automatically roll back transactional file operations. To perform file operation rollbacks, use compensational actions with exception handlers. Exception handlers have commit and rollback processing paths. For more information about exception handlers, see “[Handling Workflow Exceptions](#)” in *Using the WebLogic Integration Studio*.
- The Data Integration Plug-In provides the `BinaryData` type and file transformation capabilities used in the File Plug-In for read and write operations. The Data Integration Plug-in is automatically loaded with WebLogic Integration server.

Using the File Plug-In or Adapter for File?

In general, use the File Plug-In for basic file actions, such as reading, writing, and deleting files or transferring files between local and remote systems. If you require more capability, use the WebLogic Adapter for File. The WebLogic Adapter for File performs additional file operations and can also read and write Excel spreadsheet contents and Comma Separated Variable (CSV) formats. For more information about File System Adaptor, see the [BEA WebLogic Adapter for File](#).

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 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.

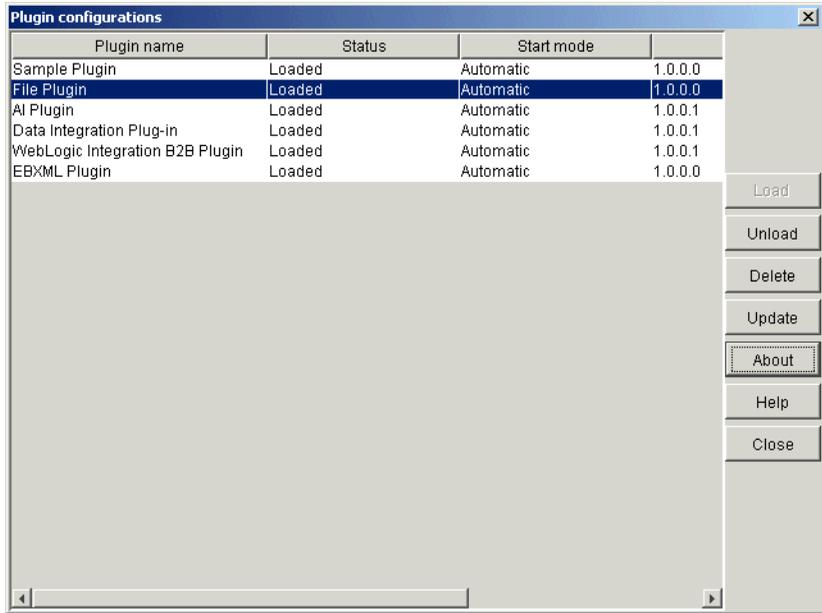
If you do not have the required knowledge of workflows or the WebLogic Integration Studio, see the following documents:

- *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>.

File Plug-In and the Plug-in Framework

Like other plug-ins, the File Plug-In extends BPM functionality. It adheres to the BPM Plug-In specification. For more information, see *Programming BPM Plug-Ins for WebLogic Integration* at <http://e-docs.bea.com/wli/docs70/devplug/index.htm>. The Plug-in Configurations window displays the plug-ins and their status, as shown in Figure 1-1.

Figure 1-1 Plug-ins Configurations Window



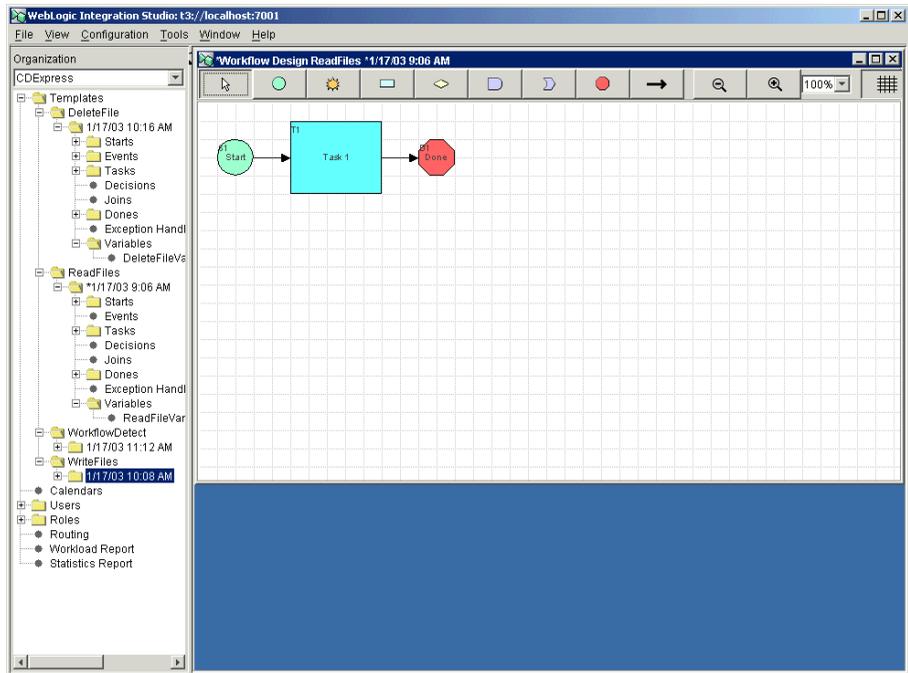
Using the File Plug-In

The following sections provide information about using the File Plug-In. These sections do *not* contain step-by-step instructions for setting up File Plug-In actions and assume that you are knowledgeable about designing WebLogic Integration workflows. For a step-by-step example, see Chapter 2, “An Example of Using the File Plug-In.”

Overview of File Plug-In Actions

This section provides general information for using the File Plug-In. It shows the Studio features needed for setting up File Plug-In actions. Figure 1-2 shows the Studio with a Workflow Design window open.

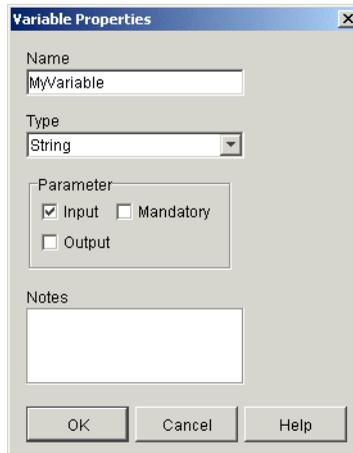
Figure 1-2 WebLogic Integration Studio



The Workflow Design window allows you construct workflows and access the properties that define workflows.

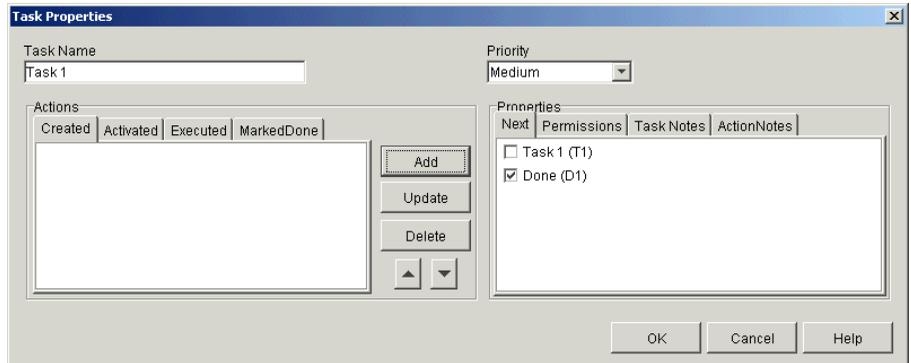
Most File Plug-In actions require a variable to hold file content. Use the Variable Properties window to define the variables used by the workflow actions, as shown in Figure 1-3.

Figure 1-3 Variable Properties



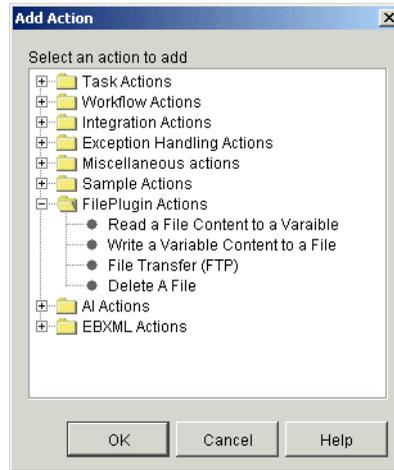
In the Workflow Design window, right-click a task node to open the Task Properties windows, where you can define workflow properties, as shown in Figure 1-4.

Figure 1-4 Task Properties



Click the Add button to open the Add Actions window. In the Add Action window, you can access many File Plug-In actions, as shown in Figure 1-5.

Figure 1-5 Add Action Window



The actions available in the File Plug-In windows and other Studio windows are discussed in the sections that follow.

Workflow Expressions

Wherever you see the Expression button  next to a field in a Studio window, the field requires that an entry formulated in the workflow expression language. In File Plug-In actions (with the exception of starting a workflow upon file creation), the following properties may require expressions:

- File Name
- Directory Name
- Time Interval—Must use an integer.
- Archive Directory
- FTP Server Name
- Remote File Name
- Local File Name

Values for these properties can be obtained from the Expression Builder and XPath Wizard, which returns 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"`

Note: 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:

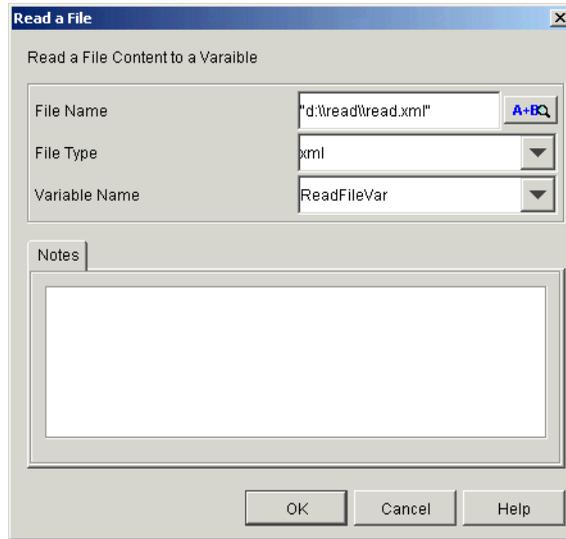
- DOS, such as "d:\\read\\read.xml"
- UNIX, such as "/home/workflow/input.txt"

For more information about expressions, see “[Using Workflow Expressions](#)” in *Using the WebLogic Integration Studio*.

Reading Files

Use the File Plug-In to read a file and assign the content to a workflow variable. Figure 1-6 shows the window for defining the reading files action.

Figure 1-6 Read a File Window



Template Definitions

Set the following properties to read a file:

Note: Before defining the workflow’s task properties, you must define a workflow variable; the file content is bounded to a workflow variable.

- Node: Any node that allows a workflow action, such as a Task node.
- Add Action window—FilePlugin Actions—Read a File Content to a Variable.
- Read a File window (Figure 1-6):
 - File Name: Entry must be formulated in the workflow expression language and include the full path name (see “Workflow Expressions” on page 1-8).

1 Using the File Plug-In

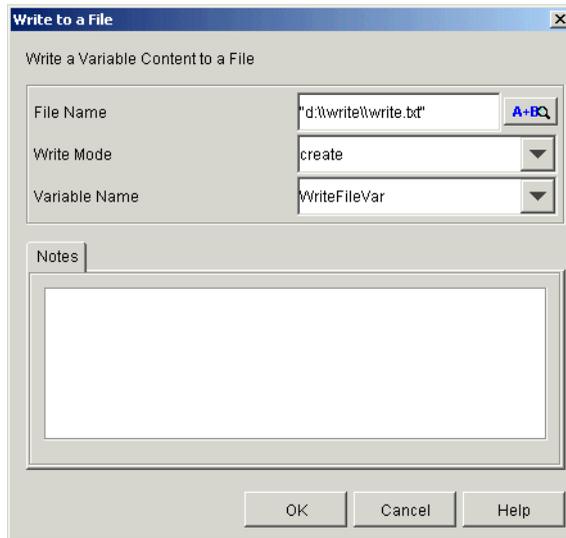
- File Type: XML, text, or binary.
- Variable Name: Variable holding the content.

Caution: Reading the same file by multiple workflow instances is allowed. You should prevent multiple threads from modifying the same file at the same time.

Write File

Use the File Plug-In to define a task that can write a variable's contents to a specified file. This action has the ability to create a new file, append content to an existing file, or overwrite an existing file. Figure 1-7 shows the window for defining write file actions.

Figure 1-7 Write a File Window



Set the following properties to write a file:

Note: Before defining the workflow's task properties, you must define a workflow variable; the file content is bounded to a workflow variable.

- Node: Any node that allows a workflow action, such as a Task node.

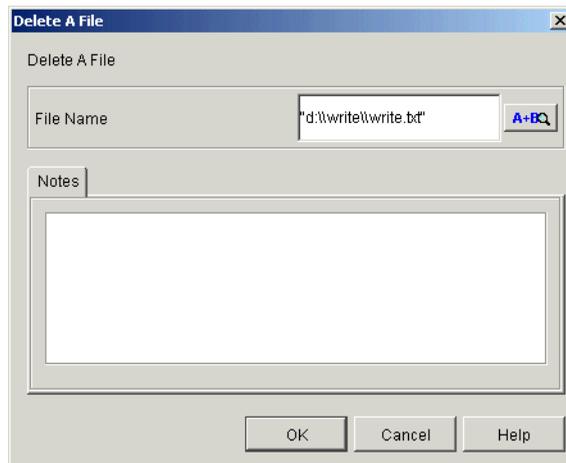
- Add Action window—FilePlugin Actions—Write a Variable Content to a File.
- Write a File window (Figure 1-7):
 - File Name: Entry must be formulated in the workflow expression language and include the full path name (see “Workflow Expressions” on page 1-8).
The available file types are text, XML, and binary.
 - Write Mode: Create, append, or overwrite.
 - Variable Name: Variable holding the content.

Caution: Multiple threads cannot update the same file. You should prevent multiple threads from modifying the same file at the same time.

Delete File

Use the File Plug-In to define workflows that can delete files. Figure 1-8 shows the action window for deleting a file.

Figure 1-8 Delete a File



Set the following properties to delete a file:

- Node: Any node that allows a workflow action, such as a Task node.

- FilePlugin Actions—Delete a File.
- Delete a File window (Figure 1-8):

File Name: Entry must be formulated in the workflow expression language and include the full path name (see “Workflow Expressions” on page 1-8).

Resume Workflow on File Detection

Use the File Plug-In to define 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 content of the file and then moves on. If the file does not exist, the workflow waits for the file to be created. Figure 1-13 shows the window for defining the event node.

Note: Only one file can only resume one workflow instance.

Figure 1-9 Event Properties Window for File Resume

The screenshot shows the 'Event Properties' dialog box for a 'File Resume' event. The 'Description' field contains 'Event' and the 'Type' dropdown is set to 'File Resume'. Below are several configuration fields:

- Directory Name: "d:\resume" (with an A+BQ button)
- File Name: "resume.xml" (with an A+BQ button)
- Time Interval (second): 60 (with an A+BQ button)
- File Type: xml (dropdown menu)
- Variable Name: varName (dropdown menu)
- Archive Directory: "d:\archive" (with an A+BQ button)

At the bottom, there are tabs for 'Variables', 'Actions', 'Next', and 'Notes'. The 'Variables' tab is active, showing a table with columns 'Variable' and 'Expression'. To the right of the table are buttons for 'Add', 'Update', and 'Delete'. At the very bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

Set the following properties for File Resume:

Note: Before defining the workflow's task properties, you must define a workflow variable; the file content is bounded to a workflow variable.

- Node: Event—Event Properties—Type drop-list—File Resume.
- Event Properties window (Figure 1-9):
 - Directory Name: Entry must be formulated in the workflow expression language and include the full path name.
 - File Name: Entry must be formulated in the workflow expression language and include the full path name. No wild card matching allowed.
 - Time Interval: Entry must be formulated in the workflow expression language. Seconds must be an integer value.
 - File Type: Text, XML, or binary.

1 Using the File Plug-In

- Archive Directory: Entry must be formulated in the workflow expression language and include the full path name.

Note: For more information about workflow expressions, see “Workflow Expressions” on page 1-8.

After the properties are set, the Event Node indicates the action setting, as shown in Figure 1-10.

Figure 1-10 Event Node with File Action



Start Workflow on Detection of File Creation

Use the File Plug-in to start a workflow when a file is created. Figure 1-11 shows the window for defining the Start node.

Note: One file can start only one workflow instance.

Figure 1-11 Start Properties Window for File Start

Start Properties

Description
Start

Timed
 Manual
 Called
 Event
 File Start

Directory Name: "d:\start"
 File Name Mask: "*whats.xml*"

Time Interval (second): 100

File Type: xml

Variable Name: WorkflowStartVar

Archive Directory: "c:\archive"

Start Organization: CDEExpress

Use workflow expression

Variables | Actions | Next | Notes

Variable	Expression

Add
Update
Delete

OK Cancel Help

Set the following properties for starting a workflow from the creation of a file:

Note: Before defining the workflow's task properties, you must define a workflow variable; the file content is bounded to a workflow variable.

- Node: Start—Event radio button—Event drop-list—File Start.

Note: After selecting the Event radio button in the Start Properties window, the Event drop-list appears. When you select File Start from the drop-list, the Start Properties window changes to show the File Start properties.

- Start Properties window (Figure 1-11):
 - Directory Name: Full path name. For example, “d:\\start\\workflow.xml”.
 - File Name Mask: Name of file. No wild card matching allowed.
 - Time Interval: Set the polling interval in integer seconds.
 - File Type: Text, XML, or binary.
 - Variable Name: Variable holding the content.
 - Archive Directory: Full path name. For example, “d:\\jan03\\archive”.

After the properties are set, the Start Node indicates the action setting, as shown in Figure 1-12.

Figure 1-12 Start Node with File Action



Get and Put Files Using FTP

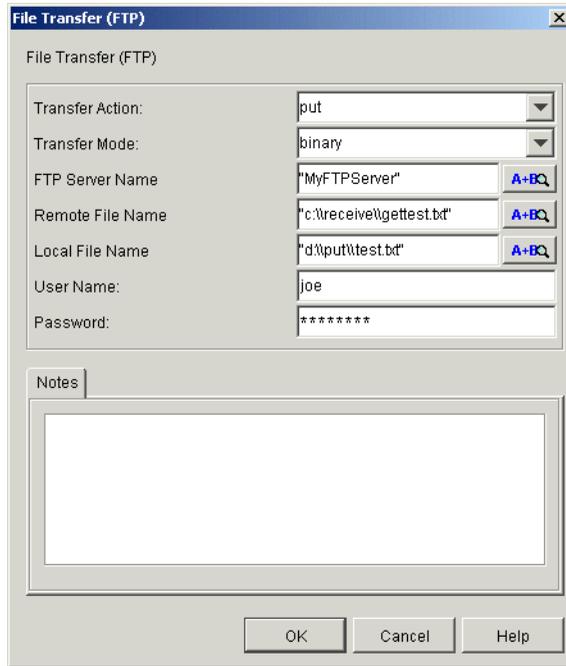
Use the File Plug-In to move files between a local system and a remote system.

Use *Get Files* to retrieve a file from a remote system for additional processing, such as starting a workflow, triggering an event, or reading information into a workflow.

Use *Put Files* to move a file from a local system to a remote system for archiving or triggering an application on the remote system.

Figure 1-13 shows the window for defining the Get and Put actions.

Figure 1-13 File Transfer (FTP) Window



Set the following properties for transferring files between systems:

Note: Before defining the transfer action, you must define an action to read the file, as described in “Reading Files” on page 1-9. As required by the Reading Files action, you must also define a workflow variable; the file content is bounded to a workflow variable.

- Node: Any node that allows a workflow action, such as a Task node.
- Add Action window—FilePlugin Actions—File Transfer (FTP).
- File Transfer window (Figure 1-13):
 - Transfer Action: Put or get.
 - Transfer Mode: ASCII or binary.
 - FTP Server Name: Entry must be formulated in the workflow expression language and include the full path name.

- Remote File Name: Entry must be formulated in the workflow expression language and include the full path name.
- Local File Name: Entry must be formulated in the workflow expression language and include the full path name.
- User Name: The name needed to log on to your FTP server.
- Password: The password needed to log on to your FTP server.

Caution: Get overwrites a file if the same file name exists in local system and Put overwrites a file if the same file name exists in remote system.

Note: For more information about workflow expressions, see “Workflow Expressions” on page 1-8.

Configuring File Plug-In for a Migrated Domain

This section tells you how to update your database schema and configure the File Plug-In for a single server domain and a cluster domain.

Updating the BPM Database Table

The File Plug-in uses a new database table called FILEPOLL. To update the BPM database with this table, run the following script:

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

In the preceding line *BEA_HOME* represents the WebLogic Platform home.

Migrating to a Single Server Domain

To configure the File Plug-In for a single server domain, edit the domain's `config.xml` file as follows:

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

```
<EJBComponent Name="WLI-BPM File Plug-in"
  Targets="<Customer_Server_Name>" URI="fileplugin-ejb.jar"/>
```

2. To add the JMS queue for the File Plug-In, add the following:

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

Migrating to a Clustered Domain

To configure File Plug-In for a clustered domain, edit the cluster's `config.xml` file as follows:

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

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

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

2. To configure the JMS Queue for File 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"/>
```

1 *Using the File Plug-In*

```
</JMSDistributedQueue>
```

3. On the cluster node c1 JMS server, add the following:

```
<JMSQueue Name="WLI_BPM_FP-c1"  
  JNDIName="com.bea.wli.bpm.FilePluginQueue-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.FilePluginQueue-c2"  
  StoreEnabled="true"  
  Template="WLI_JMSTemplate-c2"/>
```

2 An Example of Using the File Plug-In

This section provides an example of how to use the File Plug-In to transfer a file to a remote server via FTP. The first part of the example tells you how to set up a workflow in the WebLogic Integration Studio to FTP a file when the workflow is executed. The second part of the example tells you how to use the JSP Worklist to start the workflow, resulting in the transfer of a text file.

Note: This example uses the WebLogic Integration Samples Domain. For more information, see “[Configuring and Starting the Samples Domain](#)” in *Starting, Stopping, and Customizing BEA WebLogic Integration*.

Setting Up the Workflow

To set up a workflow in Studio to FTP a file, complete the following steps:

Note: For more information about using WebLogic Integration Studio, see *Learning to Use BEA WebLogic Integration* at <http://e-docs.bea.com/wli/docs70/bpmtutor/index.htm>.

1. Start the WebLogic Integration server.
 - On Windows, choose Start→Programs→BEA WebLogic Platform 7.0→WebLogic→Integration Examples→Start Server.

2 An Example of Using the File Plug-In

- On UNIX, run
`BEA_HOME/weblogic700/samples/integration/config/samples/startWebLogic.sh.`

In the preceding line, `BEA_HOME` represents the WebLogic Platform home directory.

2. After the server is running, open the WebLogic Integration Studio:

- On Windows, choose Start—BEA WebLogic Platform 7.0—WebLogic Integration 7.0—Studio.

- On UNIX, run:

```
BEA_HOME/weblogic700/integration/bin/studio.sh
```

3. Log on to the WebLogic Integration Studio with the following parameters: (See Figure 2-1.)

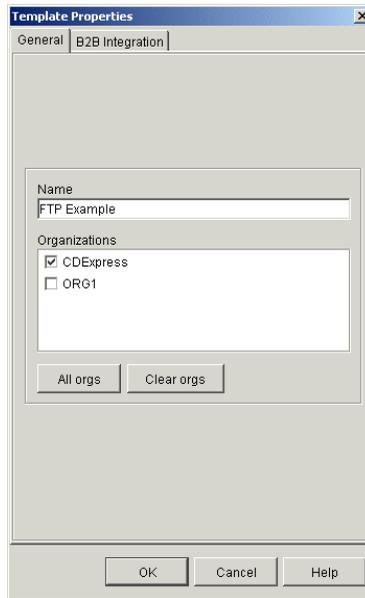
- User Name: joe
- Password: password
- Server URL: t3://localhost:7001

Figure 2-1 Logon to WebLogic Integration Studio



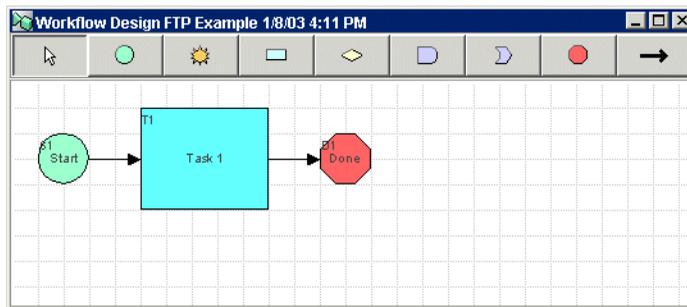
4. After WebLogic Integration Studio opens, select CDEExpress from the Organization drop-down list.
5. Right-click the Templates node and select Create Template. The Template Properties window opens, as shown in Figure 2-2.

Figure 2-2 Template Properties Window



6. On the General tab, in the Name field, enter “FTP Example” and then click OK.
7. In the Organization tree, right-click the FTP Example Template node, and then select Create Template Definition.
8. In the Template Definition FTP Example window, click OK. The Workflow Design window opens in the right pane, as shown in Figure 2-3.

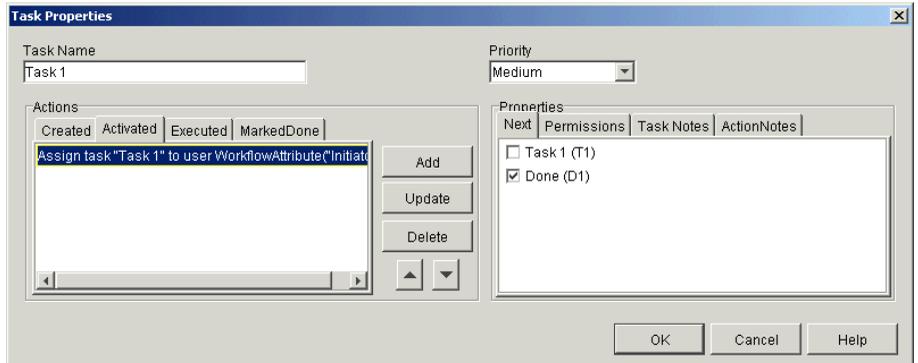
Figure 2-3 Workflow Design – FTP Example



2 An Example of Using the File Plug-In

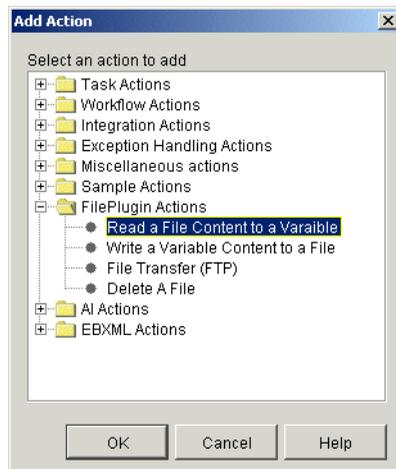
9. Right-click Task 1 and select Properties. The Task Properties window opens, as shown in Figure 2-4.

Figure 2-4 Task Properties Window



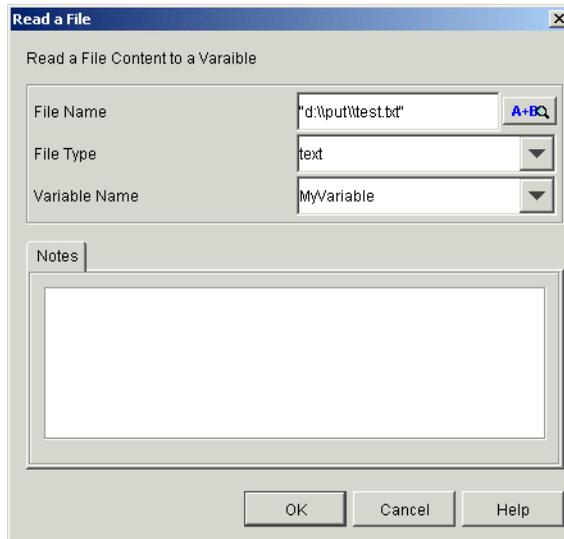
10. Make sure the Activated tab is selected. (This is the run-time state.)
11. Click Add. The Add Action window opens, as shown in Figure 2-5.

Figure 2-5 Add Action Window



12. Open the FilePlugin Actions folder, select Read a File Content to a Variable, and then click OK. The Read a File window opens, as shown in Figure 2-6.

Figure 2-6 Read a File Window



13. Set the Read a File actions as follows:

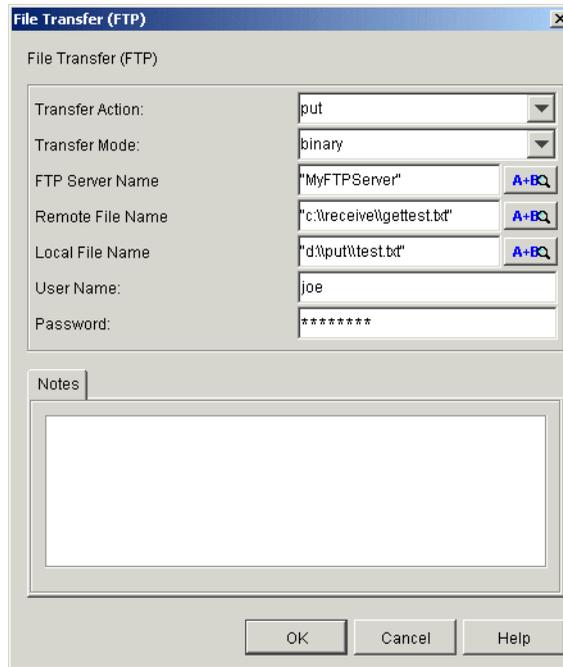
- File Name: "d:\\put\\test.txt" (enter full path including quotes)
- File Type: text
- Variable Name: My Variable

Note: Use two backslashes for Window systems.

14. In the Task Properties window, select the Executed tab, and then click Add. The Add Action dialog box opens.

15. Open the FilePlugin Actions folder, select the File Transfer (FTP) node, and then click OK. The File Transfer window opens, as shown in Figure 2-7.

Figure 2-7 File Transfer Window

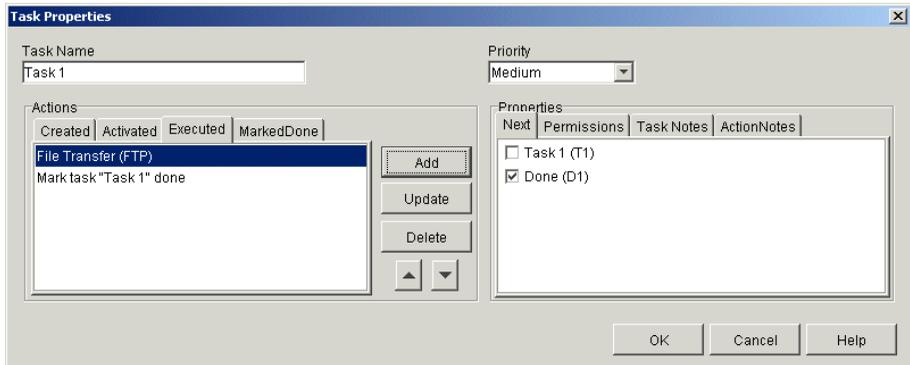


16. Set the File Transfer actions as follows:

- Transfer Action: put.
- Transfer Mode: binary.
- FTP Server Name: "Name of your FTP server" including quotes.
- Remote File Name: "c:\\receive\\gettest.txt" including quotes.
- Local File Name "d:\\put\\test.txt" including quotes.
- User Name: The name needed to log on to your FTP server.
- Password: The password needed to log on to your FTP server.

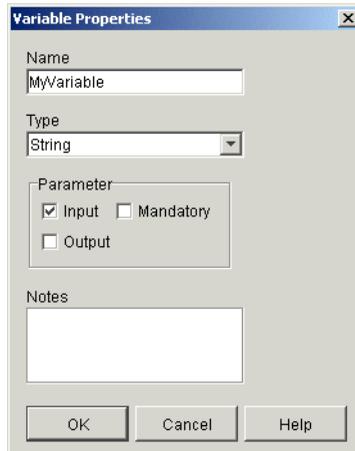
17. In the Task Properties window, select File Transfer (FTP), click the Up arrow, and then click OK. See Figure 2-8.

Figure 2-8 Task Properties – Move File Transfer Task



18. In the Organization tree, right-click the Variables Node, and then select Create Variable. The Variable Properties window opens, as shown in Figure 2-9.

Figure 2-9 Variable Properties



19. Set the variable properties as follows:

- Name: My Variable
- Type: String
- Parameter: Input

20. Right-click the Template Definition Folder and select Properties.

21. Select the Active check box and click OK.

22. Right-click the Template folder and select Save.

Note: An asterisk by the folder indicates that the changes have not been saved.

23. In your file system, create a text file corresponding to the Local File Name you created in the File Transfer window.

Activating the Workflow

In this part of the example, the JSP Worklist executes the workflow, resulting in the transfer of the text file to the remote file system.

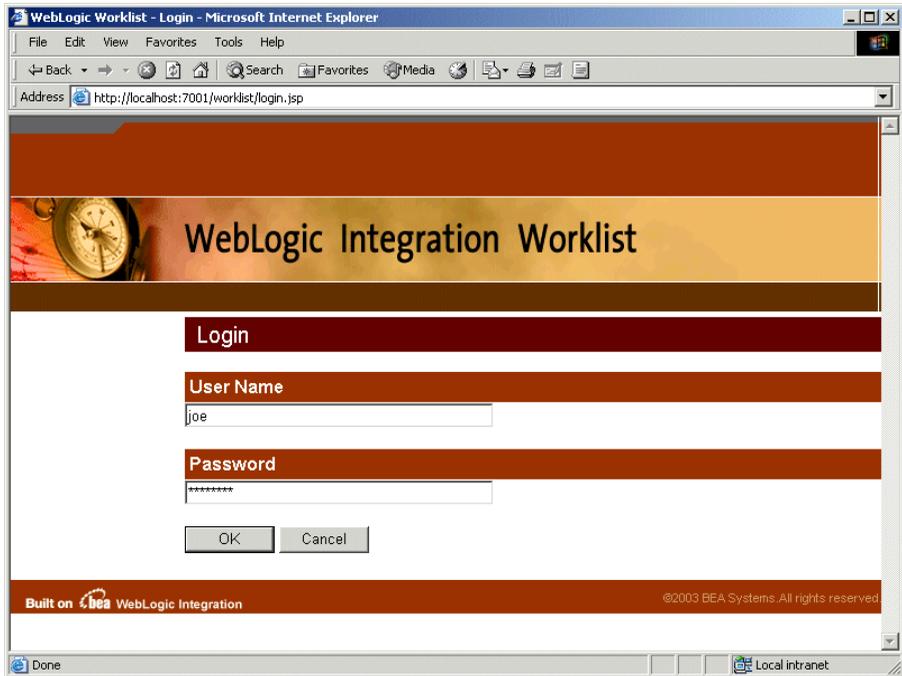
For more information about the JSP Worklist, see [Using the WebLogic Integration JSP Worklist](#).

1. To start the JSP Worklist, do one of the following:
 - On a Windows system, choose 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:7001/worklist`

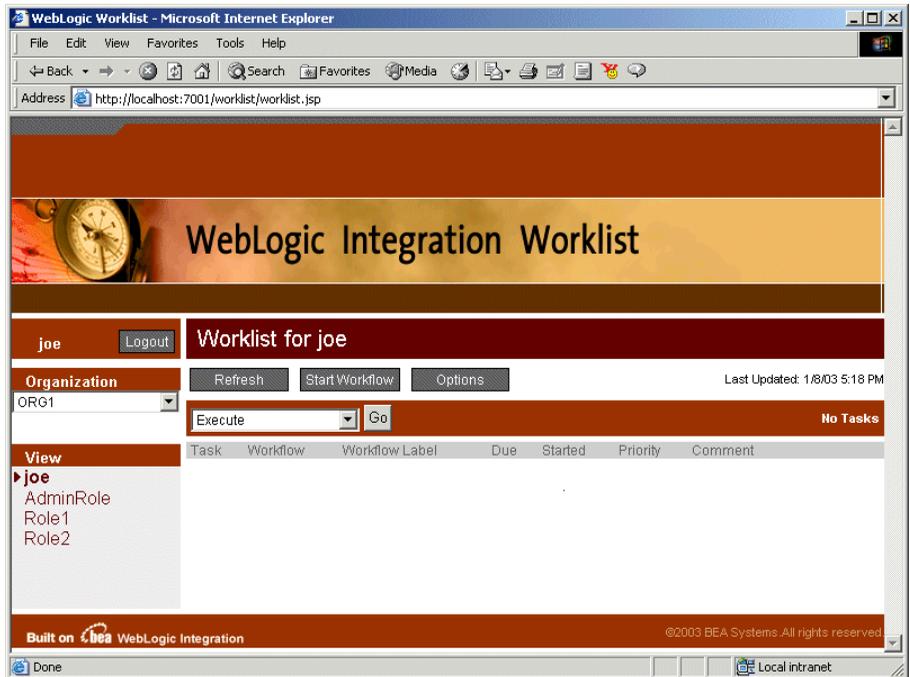
The Login page displays, as shown in Figure 2-10.

Figure 2-10 Login to WebLogic Integration JSP Worklist



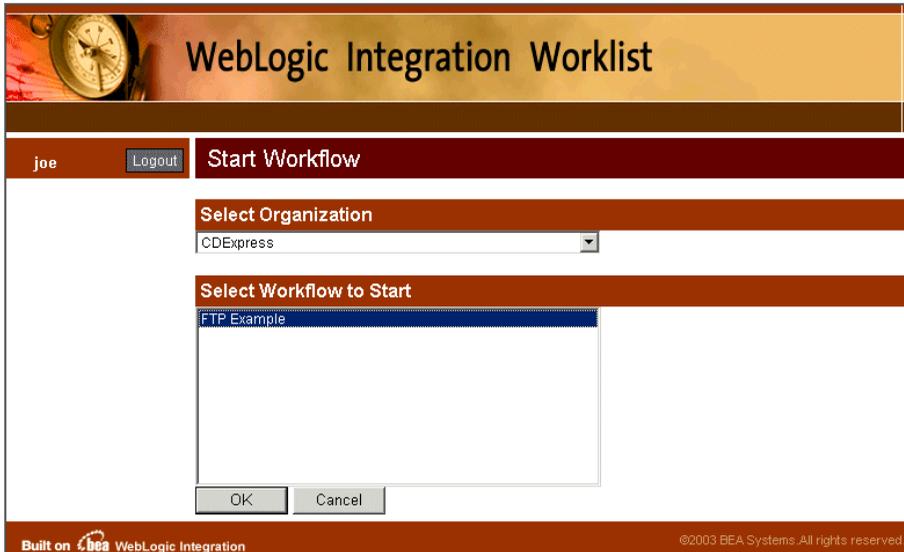
2. Enter `joe` in the User Name field.
3. Enter `password` in the Password field.
4. Click the OK button. The JSP Worklist displays, as shown in Figure 2-10.

Figure 2-11 JSP Worklist Main Page



5. Select CDExpress from the Organization drop-list.
6. Click Start Workflow. The Start Workflow page opens, as shown in Figure 2-12.

Figure 2-12 Start Workflow Page



The screenshot shows a web application interface for starting a workflow. At the top, there is a header with a clock icon and the text "WebLogic Integration Worklist". Below this is a navigation bar with the user name "joe", a "Logout" button, and the page title "Start Workflow". The main content area is divided into two sections: "Select Organization" with a dropdown menu showing "CDEExpress", and "Select Workflow to Start" with a list box containing "FTP Example". At the bottom of the list box are "OK" and "Cancel" buttons. The footer includes the BEA logo and text "Built on WebLogic Integration" and "©2003 BEA Systems. All rights reserved."

7. In Select Workflow to Start list, select FTP Example, and then click OK. The Worklist opens showing the task, as shown in Figure 2-13.

Figure 2-13 Worklist with New Task

The screenshot displays the 'WebLogic Integration Worklist' interface. At the top, there is a header with a clock icon and the title 'WebLogic Integration Worklist'. Below this, a navigation bar shows the user 'joe' with a 'Logout' button and the text 'Worklist for joe'. A secondary bar contains 'Organization' (set to 'CDExpress'), 'Refresh', 'Start Workflow', and 'Options' buttons, along with a 'Last Updated' timestamp of '1/8/03 6:29 PM'. A central bar features a dropdown menu set to 'Execute' and a 'Go' button, with the text 'Task 1-1 of 1 | First | Previous | Next | Last'. Below this is a table with columns: Task, Workflow, Workflow Label, Due, Started, Priority, and Comment. The table contains one row: 'Task 1', 'FTP Example', an empty cell, '1/8/03 6:29 PM', '1-Medium', and an empty cell. An 'Execute' button is located to the right of this row. A left sidebar shows a 'View' section with 'joe' and 'Accounting' listed. The footer includes 'Built on bea WebLogic Integration' and '©2003 BEA Systems. All rights reserved'.

8. Click Execute. The file is transferred to its target directory and the task disappears from the worklist.
9. Check your target directory to see the transferred file.

Index

A

- actions, definition of 1-1
- activating a workflow 2-8
- add actions window 1-6
- append content to file 1-10

C

- customer support contact information vii

D

- Data Integration Plug-in 1-2
- database migrating 1-18
- deleting file 1-11
- documentation conventions vii

E

- e-docs web site v
- example of using File Plug-In 2-1
- exception handlers 1-2
- expression button, about 1-8

F

- file
 - append 1-10
 - delete 1-11
 - get 1-16
 - overwrite 1-10
 - put 1-16
 - transfer 2-1

- write 1-10
- FTP 1-16
- functionality of File Plug-In, list of 1-2

G

- general information 1-4
- get file 1-16

H

- how to migrate File Plug-In 1-18

M

- migrating File Plug-In 1-18
- migrating to a clustered domain 1-19
- migrating to a single server domain 1-19
- migrating to database 1-18

O

- overview 1-4
- overwrite file 1-10

P

- plug-in configurations window 1-3
- prerequisites 1-3
- printing product documentation vi
- put files 1-16

R

- related information vi

resume workflow 1-12

S

start workflow 1-14
support, technical vii

T

transferring file example 2-1

V

variables 1-5

W

WebLogic Adapter for File 1-2
WebLogic Integration Samples Domain 2-1
what you need to know v
workflow
 activating 2-8
 resume 1-12
 start 1-14
workflow design window 1-6
workflow expressions 1-8
write file 1-10