BEA WebLogic Java Adapter for Mainframe

Workflow Processing Guide

BEA WebLogic Java Adapter for Mainframe 4.2
Document Edition 4.2
July 2001

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA WebLogic Java Adapter for Mainframe Workflow Processing Guide

Document Edition	Part Number	Date	Software Version
4.2	N/A	July 2001	BEA WebLogic Java Adapter for Mainframe 4.2

Contents

About This Document

1. Workflow Processing with Java Adapter for Mainframe

2. Using Mainframe Applications with Workflows

3. Example of JAM Integrated with the WebLogic Process Integrator

About This Document

The BEA WebLogic Java Adapter for Mainframe product (hereafter referred to as JAM) is a gateway connectivity application that enables client/server transactions between Java applications and OS/390 CICS or IMS programs.

This document provides the following topics on using JAM software with WebLogic Process Integrator workflows:

- "Workflow Processing with Java Adapter for Mainframe" describes the BEA products that work together to allow workflows to access mainframe applications and allow mainframe applications to initiate workflows.
- "Using Mainframe Applications with Workflows" provides the tasks required to use mainframe applications with workflows in WebLogic Process Integrator.
- "Example of JAM Integrated with the WebLogic Process Integrator" gives the user an example of how a workflow uses JAM to access information in a mainframe application.

What You Need to Know

This document is intended for system administrators, application programmers, and business analysts who will use the BEA WebLogic Java Adapter for Mainframe application with WebLogic Process Integrator for workflow processing.

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

A PDF version of this document is available on the JAM documentation Home page on the e-docs Web site (and also on the installation 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 JAM 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 publications are available for JAM 4.2:

- *BEA WebLogic Java Adapter for Mainframe Release Notes*
- *BEA WebLogic Java Adapter for Mainframe Introduction*
- *BEA WebLogic Java Adapter for Mainframe Installation Guide*
- *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*
- *BEA WebLogic Java Adapter for Mainframe Programming Guide*
- *BEA WebLogic Java Adapter for Mainframe Scenarios Guide*

- *BEA WebLogic Java Adapter for Mainframe Workflow Processing Guide*
- *BEA WebLogic Java Adapter for Mainframe Reference Guide*

Contact Us

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

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Java Adapter for Mainframe 4.2 release.

If you have any questions about this version of JAM, or if you have problems installing and running JAM, 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 that 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
blue text	Indicates a hypertext link in PDF or HTML
italics	Indicates emphasis or book titles or variables.
`“string with quotes”`	Indicates a string entry that requires quote marks.
UPPERCASE TEXT	Indicates generic file names, device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	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` **`xa_commit`** `( )`
`{ }`	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:
	`buildclient [-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:
	`buildclient [-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 Workflow Processing with Java Adapter for Mainframe

Many customers need an advanced e-business infrastructure that incorporates process rules, workflow, and application components. BEA designed WebLogic Java Adapter for Mainframe (JAM) to seamlessly plug in to BEA WebLogic Process Integrator, a powerful process integration and workflow engine. This software combination allows you to create business processes, process rules, and workflows that access mainframe applications.

Workflow Processing with WebLogic Process Integrator

WebLogic Process Integrator provides an intuitive flowchart model for defining business processes that work across the enterprise and in support of business-to-business interactions with minimal human interaction. The powerful workflow engine executes those processes, automatically invoking the resources required at each step. Online process monitoring provides real-time data about how the processes are working and any necessary changes can be made dynamically.

Fully J2EE compliant, WebLogic Process Integrator provides an intuitive graphical user interface that enables businesses to design, monitor, and modify workflow and e-business processes without scripting or programming. Its run-time engine automates and manages the activities of applications and such technologies as fax, WAP-enabled telephones, and other devices, with exception-based human intervention.

WebLogic Process Integrator uses XML as its standard data format. To access data from a legacy system in WebLogic Process Integrator, the data must be converted to XML first. WebLogic Process Integrator uses WebLogic XML/Non-XML Translator to translate data from binary to XML or from XML to binary.

Data Translation with WebLogic XML/Non-XML Translator

XML is quickly becoming the standard for exchanging information between applications and is invaluable in integrating disparate applications. WebLogic XML/Non-XML Translator provides for an exchange of information between applications by supporting data translations between binary formats from legacy systems and XML. WebLogic XML/Non-XML Translator normalizes legacy data into XML so it may be directly consumed by XML applications such as WebLogic Process Integrator.

Accessing Mainframe Applications Using Java Adapter for Mainframe

WebLogic Java Adapter for Mainframe seamlessly plugs in to WebLogic Process Integrator, allowing business analysts to include mainframe applications in business processes. Within the WebLogic Process Integrator framework, JAM uses WebLogic XML/Non-XML translator to translate mainframe data to and from XML. JAM can be used with WebLogic Process Integrator in the following two ways:

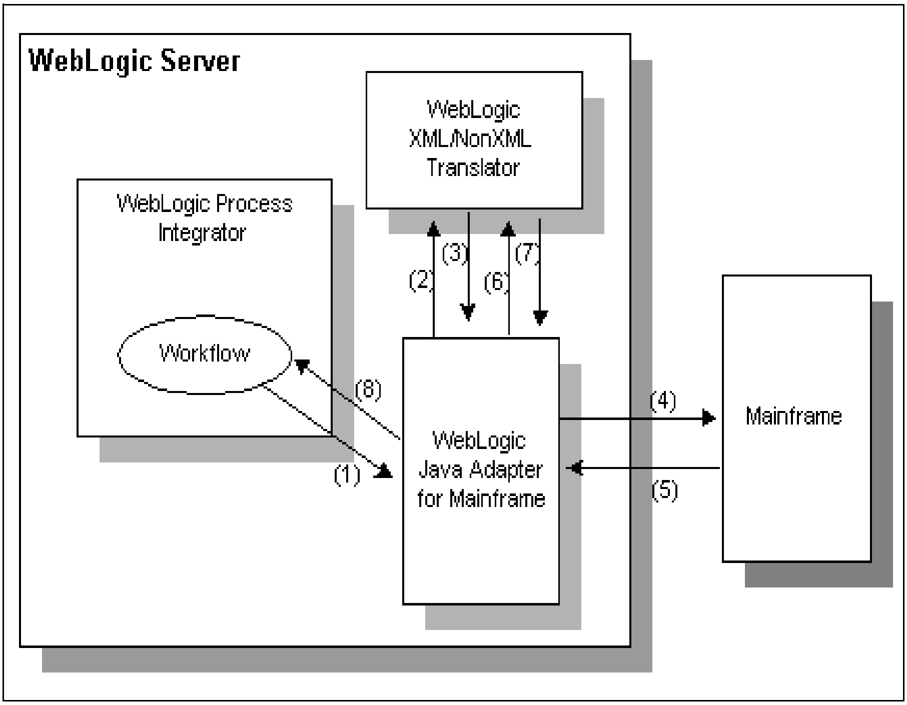
- WebLogic Process Integrator workflows can invoke applications on the mainframe. Data from the workflow is passed to the invoked application and any resulting data is returned to the workflow.

- Mainframe applications initiate a workflow in WebLogic Process Integrator using an XML event interface.

Understanding WebLogic Process Integrator Interaction with the Mainframe

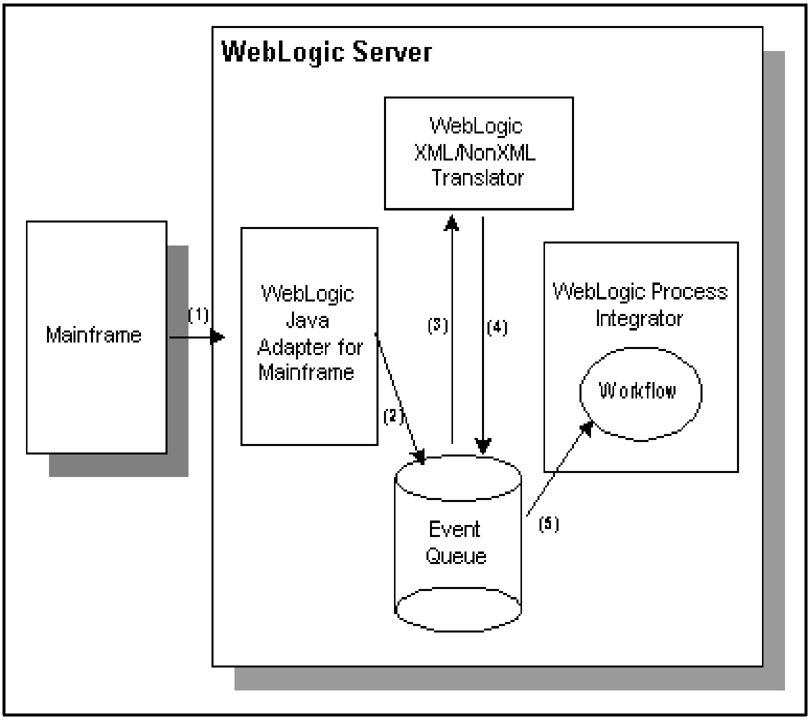
Figure 1-1 illustrates the flow of data when a WebLogic Process Integrator workflow accesses mainframe applications through JAM. When WebLogic Process Integrator sends data to the mainframe application, the JAM gateway converts the XML data received from WebLogic Process Integrator to the binary format expected by the mainframe application. The JAM gateway invokes WebLogic XML/Non-XML Translator to perform this conversion. It then invokes the mainframe program and receives a response. This response is sent to WebLogic XML/Non-XML Translator to convert the data from mainframe format to XML. Then the converted response is sent to WebLogic Process Integrator.

Figure 1-1 Workflows Accessing Mainframe Applications


WebLogic Server
WebLogic XML/NonXML Translator
WebLogic Process Integrator
Workflow
WebLogic Java Adapter for Mainframe
Mainframe
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)

JAM also allows a workflow to be initiated by an event triggered by a mainframe application. When JAM receives a request from the mainframe, it looks within its configuration to find the destination application within WebLogic Server. If the specified destination is WebLogic Process Integrator, then JAM puts the mainframe data on a WebLogic Process Integrator queue. WebLogic XML/Non-XML Translator pulls the data off the Event Queue and recognizes that translation is required. The translation is performed and the data is returned to the queue. WebLogic Process Integrator workflow that uses this XML is then initiated. This flow of data is illustrated in Figure 1-2.

Figure 1-2 Mainframe Application Initiating a Workflow


WebLogic Server
WebLogic XML/NonXML Translator
Mainframe
(1)
WebLogic Java Adapter for Mainframe
(2)
(3)
(4)
WebLogic Process Integrator
Workflow
Event Queue
(5)

CHAPTER

2 Using Mainframe Applications with Workflows

BEA WebLogic Java Adapter for Mainframe (JAM) works with BEA WebLogic Process Integrator to access information or applications on the mainframe. This interaction occurs in two different ways:

- WebLogic Process Integrator workflows invoke mainframe programs
- Mainframe programs initiate WebLogic Process Integrator workflows

Action List

To use JAM with WebLogic Process Integrator, you must complete the following tasks.

	Your action...	**Refer to...**
1	Verify prerequisite tasks have been completed.	"Prerequisites"
2	Verify the setup for the JAM Plug-in for WebLogic Process Integrator.	"Task 1: Verifying the JAM Plug-in Setup"

	Your action...	Refer to...
3	Set up data translation.	"Task 2: Setting Up Data Translation"
4	Set up workflow with JAM for mainframe services.	"Setting Up the Workflow with JAM for Mainframe Services"

Prerequisites

Before creating WebLogic Process Integrator workflows with JAM, make sure the following tasks have been completed:

	Your action...	Refer to...
1	Install and configure WebLogic Server.	*BEA WebLogic Server Installation Guide*
2	Install and configure WebLogic Process Integrator.	*BEA WebLogic Process Integrator Installation Guide*
3	Install and configure WebLogic XML/Non-XML Translator.	*BEA WebLogic XML/Non-XML Translator Installation Guide*
4	Install and configure WebLogic Java Adapter for Mainframe.	*BEA WebLogic Java Adapter for Mainframe Installation Guide*
5	Read Release Notes for each product so that you are aware of any special considerations.	BEA WebLogic product documentation
6	Understand the way these products work together.	"Understanding WebLogic Process Integrator Interaction with the Mainframe" in this guide

Preparing to Use JAM in Workflow Processing

To use mainframe applications during workflow processing with WebLogic Process Integrator, you must prepare your system. The following tasks must be completed:

- Verifying the JAM Plug-in setup

  This task is required only once.

- Setting up data translation.

  This task must be completed each time a new COBOL copybook is used.

Task 1: Verifying the JAM Plug-in Setup

The JAM Plug-in is a component that allows the JAM gateway to interact with WebLogic Process Integrator. It runs with WebLogic Process Integrator in the WebLogic Server environment. The JAM Plug-in should have been set up during the JAM installation process.

1. If the JAM Plug-in was not properly set up, you may need to add the entries in Listing 2-1 to the WebLogic Process Integrator application definition to the WebLogic Server configuration file (`config.xml`).

Warning: Copy the `config.xml` file and store it in an alternate location as a backup in case the `config.xml` file you are working with is corrupted during setup.

Listing 2-1 JAM Plug-in Setup

```
<EJBComponent Name="jampi.jar" Targets="myserver"
       URI="jampi.jar"/>

<WebAppComponent Name="com.bea.jam.JamPlugin"
       Targets="myserver" URI="jampi.war"/>
```

The WebLogic Server configuration file (config.xml) is located at:

```
<bea_home> <install.dir>/config/<domain.name>
```

In this address:

`install.dir`
: is where WebLogic Process Integrator is installed

`domain.name`
: is the domain from which WebLogic Process Integrator is started

Listing 2-2 shows an example of this address:

Listing 2-2 Example of config.xml Address

```
C:/BEA/WLPI/config/wlpidomainexample
```

2. Review your server configuration. If the server name is not `myserver`, the `Targets` entry needs to be modified with the name of your server.
3. Verify that the WebLogic Process Integrator application configuration section of the WebLogic Server `config.xml` file contains the entries shown in Listing 2-3. If not, edit the `config.xml` file with the appropriate entries.

Listing 2-3 WebLogic Process Integrator Application Configuration Section of the WebLogic Server config.xml

```
<Application Deployed="true" Name="WLPI Application" Path="lib">

    <EJBComponent Name="wlpi-mdb-ejb.jar" Targets="myserver"
    URI="wlpi-mdb-ejb.jar"/>

    <EJBComponent Name="repository-ejb.jar"
    Targets="myserver"URI="repository-ejb.jar"/>

    <EJBComponent Name="wlpi-ejb.jar" Targets="myserver"
     URI="wlpi-ejb.jar"DeploymentOrder=0/>

    <EJBComponent Name="wlxtpi.jar" Targets="myserver" URI="wlxtpi.jar"/>

    <WebAppComponent Name="com.bea.wlxt.WLXTPlugin"
    Targets="myserver"URI="wlxtpi.war"/>
```

```
   <EJBComponent Name="jampi.jar" Targets="myserver" URI="jampi.jar"/>

   <WebAppComponent Name="com.bea.jam.JamPlugin" Targets="myserver"
   URI="jampi.war"/>

</Application>
```

Task 2: Setting Up Data Translation

To ensure that workflows operate properly, you must set up your system to enable the different software applications to interact. In order for WebLogic XML/Non-XML Translator to convert XML data to the binary format expected by your mainframe program, a description of the binary format must be created and stored in your WebLogic Process Integrator repository. WebLogic XML/Non-XML Translator Format Builder can be used to create this description. Format Builder imports the COBOL copybook for your mainframe application, and automatically creates a description that details the layout of the COBOL copybook. Using this description, WebLogic XML/Non-XML Translator converts instances of this layout to and from XML.

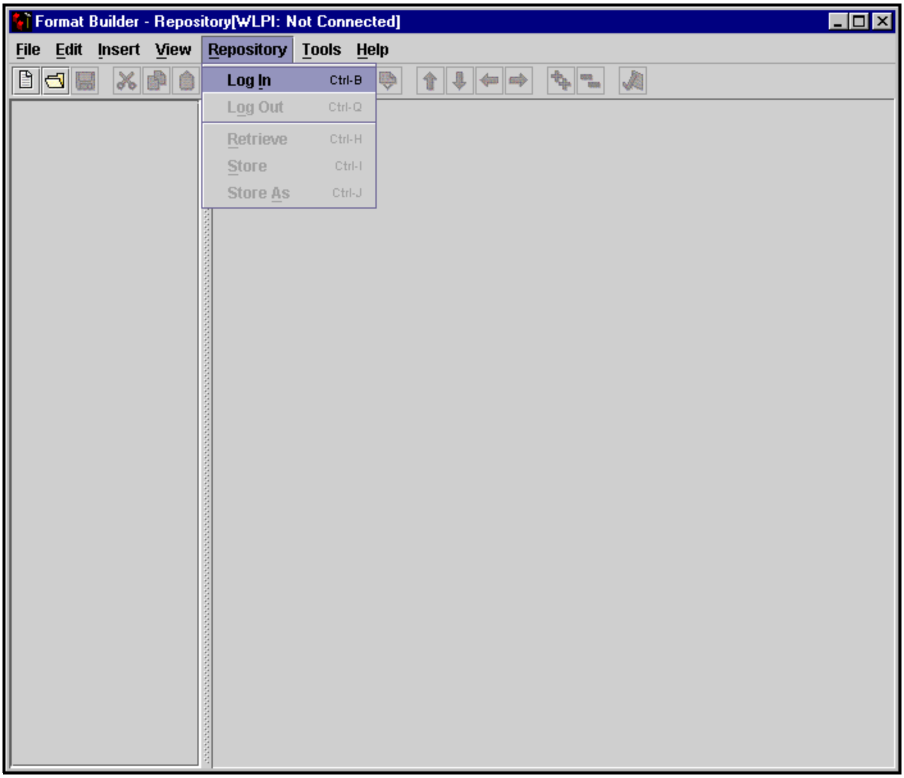
To set up your system for the required data translation, complete the following steps:

1. Obtain the COBOL copybook(s) or C header file(s) used by your mainframe application. For more information about obtaining the COBOL copybook, refer to the *BEA Java Adapter for Mainframe Programming Guid*e.

2. Import the Copybooks(s) or C header file(s) into the WebLogic Process Integrator repository using the following steps:

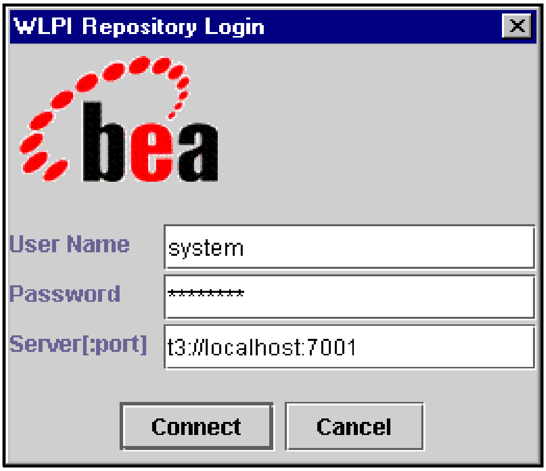
   a. Launch the WebLogic XML/Non-XML Translator Format Builder and select **Log In** from the Repository menu, as shown in Figure 2-1.

Figure 2-1 WebLogic XML/Non-XML Translator



b. From the WLPI Repository Login dialog box, log in as shown in Figure 2-2. Enter your user name and password. If your server is different than the default shown in Figure 2-2, enter your server address.

Figure 2-2 WLPI Repository Login



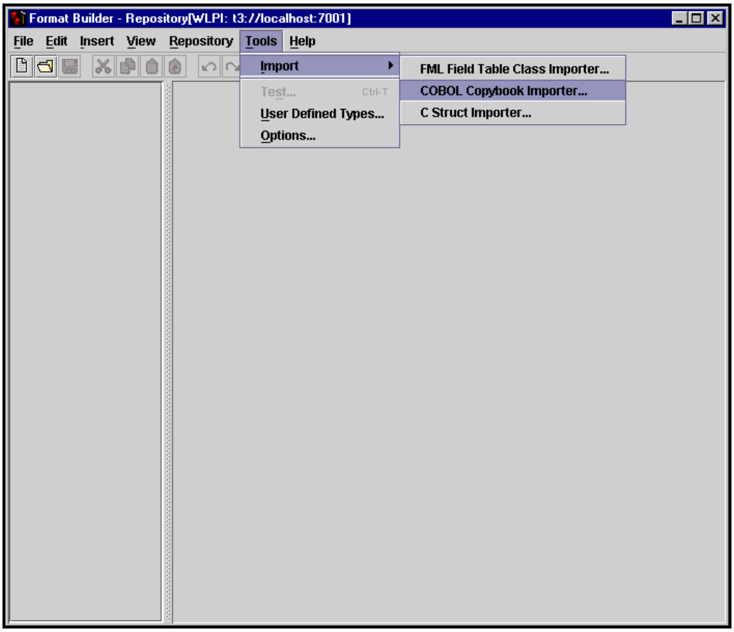
User Name: system

Password: *********

Server[:port]: t3://localhost:7001

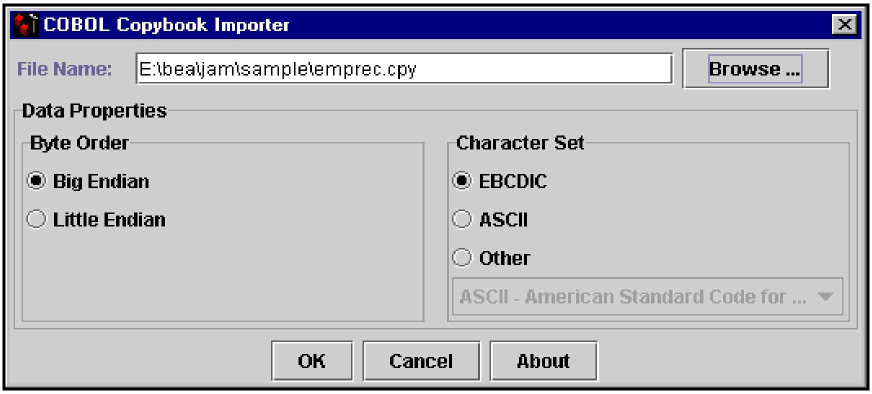
c. From the Tools menu, select **Import** and **COBOL Copybook Importer**, as shown in Figure 2-3. If you are working with a C Struct, select **C Struct Importer**.

Figure 2-3 WebLogic XML/Non-XML Translator Tools Menu


Format Builder - Repository[WLPI: t3://localhost:7001]
File
Edit
Insert
View
Repository
Tools
Help
Import
Test...
Ctrl-T
User Defined Types...
Options...
FML Field Table Class Importer...
COBOL Copybook Importer...
C Struct Importer...

d. Import the copybook files into WebLogic XML/Non-XML Translator by entering the copybook file name in the **File Name** field. If you are unsure of the location or name of the copybook file, use the **Browse** feature to locate the correct file. See Figure 2-4. The `emprec.cpy` is a sample copybook available in the `samples.jar` file.

Figure 2-4 WebLogic XML/Non-XML Translator COBOL Copybook Importer Dialog Box



File Name: E:\bea\jam\sample\emprec.cpy

Byte Order

● Big Endian

○ Little Endian

Character Set

● EBCDIC

○ ASCII

○ Other

ASCII - American Standard Code for ...

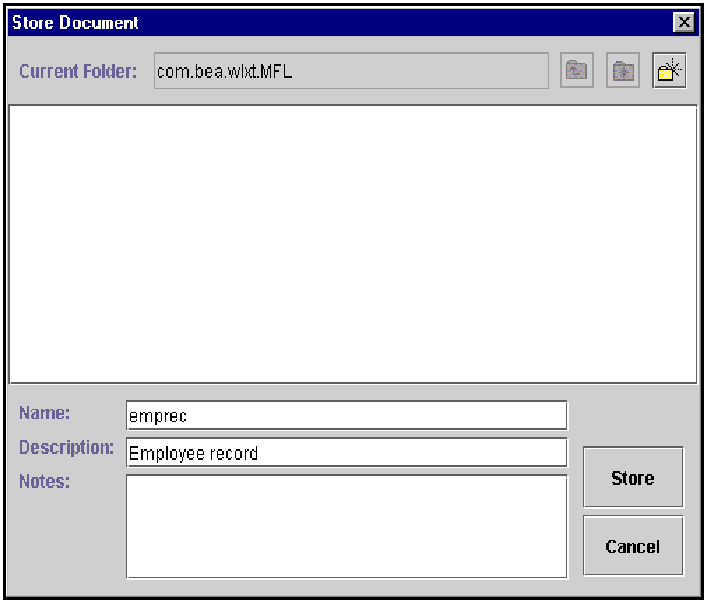
e. The COBOL Copybook Importer dialog box displays the default settings, as shown in Figure 2-4. To accept these default settings, click **OK**. If the default settings are not appropriate for your use, refer to the “Building Format Definitions” section of the *BEA WebLogic XML/Non-XML Translator User Guide* for more information about other options.

f. After your copybook has been imported, select **Store As** from the Repository menu to open the **Store Document** dialog box.

g. Enter your name for this document in the **Name** field, as shown in Figure 2-5. This name is the `SCHEMA` name you will use to modify the gateway configuration file. Enter appropriate information in the **Description** and **Notes** fields, as needed.

h. Click **Store** in the **Store Document** dialog box to save the document in the WebLogic XML/Non-XML Translator Repository.

Figure 2-5 WebLogic XML/Non-XML Translator Store Document Dialog Box



Current Folder: com.bea.wlxt.MFL

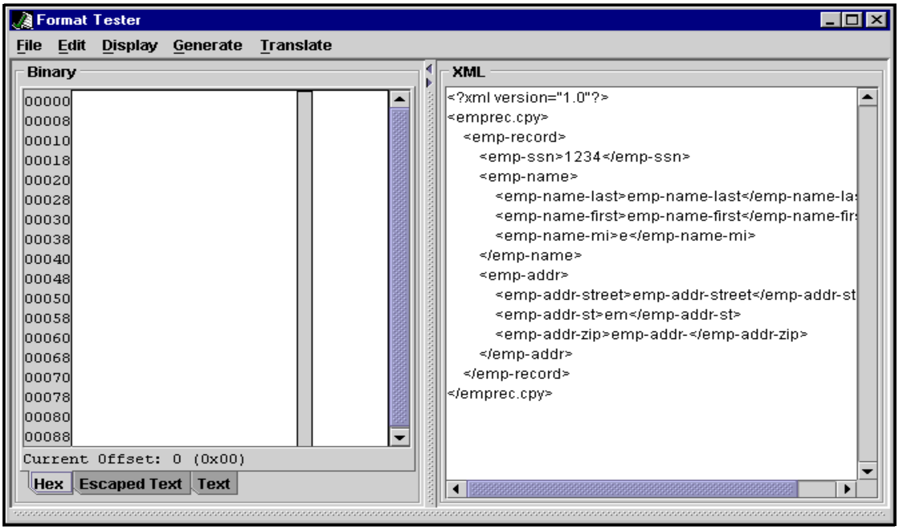
Name: emprec

Description: Employee record

Notes:

i. From the Tools menu, select **Test** to open the **Format Tester** dialog box as shown in Figure 2-6. From the **Generate** menu, select **XML** to display a sample of the XML you will use during the creation of your workflow.

Figure 2-6 Sample XML



```
00000
00008
00010
00018
00020
00028
00030
00038
00040
00048
00050
00058
00060
00068
00070
00078
00080
00088
Current Offset: 0 (0x00)
```

```
<?xml version="1.0"?>
<emprec.cpy>
  <emp-record>
    <emp-ssn>1234</emp-ssn>
    <emp-name>
      <emp-name-last>emp-name-last</emp-name-la
      <emp-name-first>emp-name-first</emp-name-fir
      <emp-name-mi>e</emp-name-mi>
    </emp-name>
    <emp-addr>
      <emp-addr-street>emp-addr-street</emp-addr-st
      <emp-addr-st>em</emp-addr-st>
      <emp-addr-zip>emp-addr-</emp-addr-zip>
    </emp-addr>
  </emp-record>
</emprec.cpy>
```

Setting Up the Workflow with JAM for Mainframe Services

Determine how your workflow will use mainframe services:

- Request mainframe services from WebLogic Process Integrator
- Initiate workflows in WebLogic Process Integrator from the mainframe

Requesting Mainframe Services from the Workflow

To request mainframe services from WebLogic Process Integrator, you must:

- Modify the JAM gateway configuration file (`jcrmgw.cfg`)
- Add appropriate mainframe actions to tasks contained in the WebLogic Process Integrator workflow template definition

Modify the Gateway Configuration File

The JAM gateway configuration file must have a `JC_REMOTE_SERVICE` entry for each of the services to be accessed using JAM. This definition must specify the schema that was created in the previous step.

The JAM gateway configuration file is located at:

```
<bea_home> <install.dir>/config/<domain.name>/jcrmgw.cfg
```

In this address:

`install.dir`
is where WebLogic Process Integrator is installed

`domain.name`
is the domain from which WebLogic Process Integrator was started

To access mainframe applications from a WebLogic Process Integrator workflow, specify each remote service in the JC_REMOTE_SERVICES section of the JAM gateway (`jcrmgw.cfg`) file. For specific information about configuring the JAM gateway, refer to the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*.

- If the translation schemas for the request and response are the same, you can specify them with a single schema entry for the service as follows:

  ```
  SCHEMA=myemployeerecord
  ```

- If the translation schemas for the request and reply are different, specify each in the following way:

  ```
  INPUTSCHEMA=myemployeerecordin
  ```

  `INPUTSCHEMA` specifies the translation schema for the request sent to the mainframe.

  ```
  OUTPUTSCHEMA=myemployeerecordout
  ```

  `OUTPUTSCHEMA` specifies the translation schema for the response received from the mainframe.

Listing 2-4 is an example of the `JC_REMOTE_SERVICES` section of the `jcrmgw.cfg` file.

Listing 2-4 `JC_REMOTE_SERVICES` Example

```
#----------------------------------------------------------------
*JC_REMOTE_SERVICES
sampleRead    RDOM="CICS4"
              RNAME="DPLDEMOR"
              TRANTIME=10000
              SCHEMA=sample.emprec
sampleUpdate  RDOM="CICS4"
              RNAME="DPLDEMOU"
              TRANTIME=10000
              SCHEMA=sample.emprec
sampleCreate  RDOM="CICS4"
              RNAME="DPLDEMOC"
              TRANTIME=10000
              SCHEMA=sample.emprec
```

For specific information about configuring the JAM gateway, refer to the *WebLogic Java Adapter for Mainframe Configuration and Administration Guide*.

Using Mainframe Actions in Workflows

You can design and edit Mainframe Actions used by WebLogic Process Integrator workflows in the following ways:

- Add mainframe actions
- Update mainframe actions
- Delete mainframe actions
- Change the sequence of mainframe actions

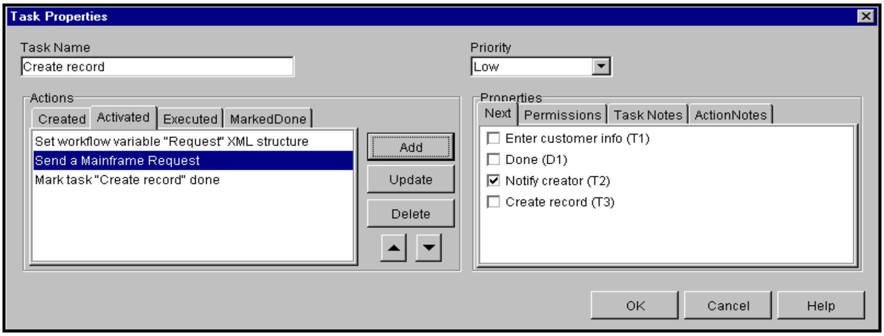
Adding Mainframe Actions

Mainframe Actions can be added during design time from the WebLogic Process Integrator Studio Console. The following steps allow you to add a mainframe action:

1. Open an existing or new workflow template definition to which you want to add a mainframe action.

2. From the workflow template definition, double-click the **Task** node to which you want to add a mainframe action. The Task Properties dialog box displays as shown in Figure 2-7.

Figure 2-7 Task Properties



Task Name: Create record

Priority: Low

Set workflow variable "Request" XML structure
Send a Mainframe Request
Mark task "Create record" done

□ Enter customer info (T1)
□ Done (D1)
☑ Notify creator (T2)
□ Create record (T3)

3. In the Task Properties dialog box shown in Figure 2-7, select the appropriate **Actions** tab to designate when the mainframe action will be executed.

   Determine when the action is executed by selecting the appropriate tab defined in Table 2-1.

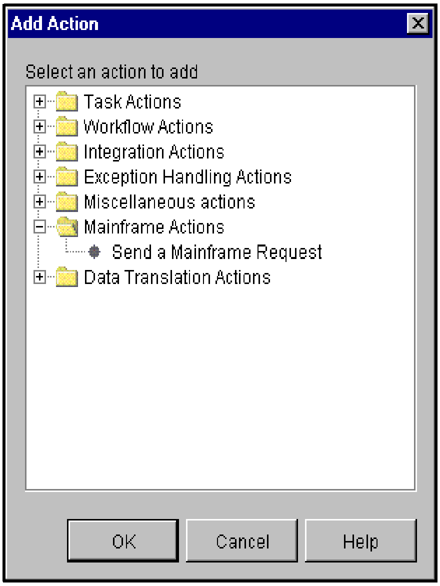
Table 2-1 WebLogic Process Integrator Actions Definitions

Tab	Description
Created	The workflow is started and all tasks in the workflow are created.
Activated	The workflow diagram indicates a task transition has occurred, usually as a result of another task being marked as done or a decision being evaluated. The task is available to be performed.
Executed	An event has occurred that causes this task to be executed. This usually occurs when a user selects a task in the worklist and executes it, often by double-clicking the task, or when an **Execute Task** action is performed for that task.
MarkedDone	The task has been completed. This event usually occurs in response to the execution of a **Mark Task as Done** action.

The *BEA WebLogic Process Integrator Studio User Guide* provides information about the execution of workflows.

4. Click **Add** to display the Add Action dialog box shown in Figure 2-8.

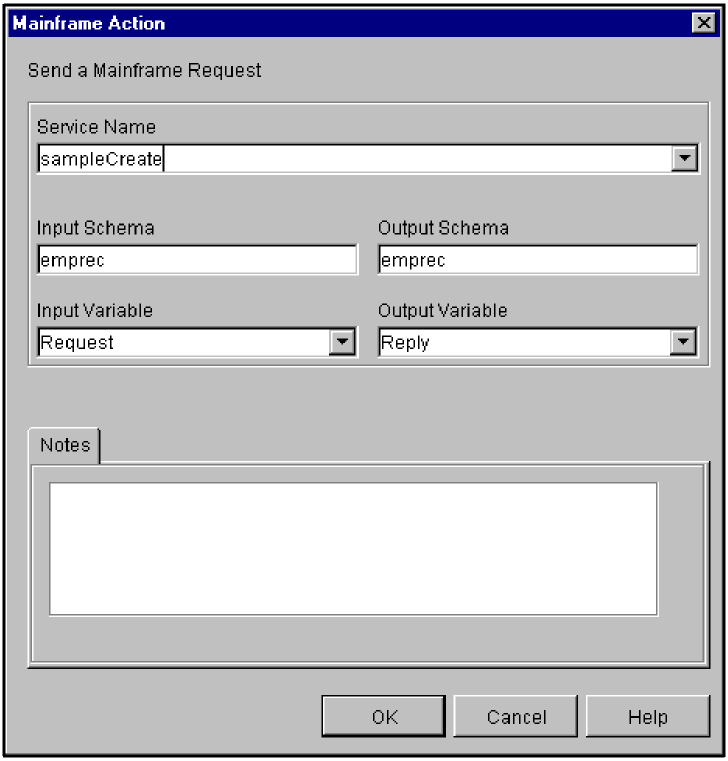
Figure 2-8 Adding Mainframe Actions



5. Double-click the Mainframe Actions folder to expand it. Select **Send a mainframe request**, and click **OK** to display the Mainframe Action dialog box.

6. Enter data in the fields as shown in Figure 2-9. A description of the input fields for this dialog box is shown in Table 2-2.

Figure 2-9 Mainframe Actions Dialog Box



Mainframe Action

Send a Mainframe Request

Service Name: sampleCreate

Input Schema: emprec

Output Schema: emprec

Input Variable: Request

Output Variable: Reply

Table 2-2 lists the fields used for adding and updating mainframe actions when using WebLogic Java Adapter for Mainframe (JAM) with a WebLogic Process Integrator workflow.

Table 2-2 Mainframe Action Definitions

Field	Description
Mainframe Action Definition	
Service Name	Select the name of a remote service from the drop-down list or enter the name of a new service. This drop-down list displays all the remote services defined in the JAM jcrmgw.cfg file. If you enter the name of a new service, you must be sure to add the service to the JC_REMOTE_SERVICES section of the `jcrmgw.cfg` file before you can run the workflow.
Input Schema	Displays the input schema name that corresponds to the Service Name you selected. The schema specified by this field must be created using Format Builder prior to executing this workflow. You may also manually enter a new schema name, but you must define the schema with Format Builder before you can run the workflow. Schemas must be in the repository.
Output Schema	Displays the output schema name that corresponds to the Service Name you selected. The schema specified by this field must be created using Format Builder prior to executing this workflow. You may also manually enter a new schema name, but you must define the schema with Format Builder before you can run the workflow. Schemas must be in the repository.
Input Variable	Select an input variable from the drop-down list or manually enter a new variable. This variable contains the XML document that is converted to binary format using the Input Schema and then sent to JAM. The variable must be of type "String" or "XML." If you enter a variable name that does not currently exist, you will be asked if you wish to create it.

Field	Description
Output Variable	Select an output variable from the drop-down list or manually enter a new variable. This variable is the destination for response data from the mainframe that has been received by the JAM gateway and translated into XML using the output schema. The variable must be of type "String" or "XML." If you enter a variable name that does not currently exist, you will be asked if you wish to create it.
Notes	Enter your own custom notes about the mainframe action.
Mainframe Action Buttons	
OK	Saves your mainframe action definition information and closes the Mainframe Action dialog. If any fields are blank, an error message displays and you will be returned to the Mainframe Action dialog. If either variable name specifies a variable that does not exist, a message displays allowing you to add the variable.
Cancel	Closes the Mainframe Action dialog without saving any mainframe action definition information that you entered.

7. Click **OK**. The action is added to the Task Properties dialog box and will be implemented when the workflow is executed.

Updating Mainframe Actions

Mainframe actions can be updated during design time or runtime from the WebLogic Process Integrator Studio Console. The following steps allow you to update a mainframe action:

1. Open an existing workflow template definition that you want to update.
2. From the workflow template definition, double-click the **Task** node you want to update. The Task Properties dialog box displays.
3. In the Task Properties dialog box shown in Figure 2-7, select the appropriate **Actions** tab to designate when the mainframe action is executed. The Actions are defined in Table 2-1.
4. Select the **Mainframe Action** to be updated from the **Actions** tab.

5. Click **Update** to display the Update Action dialog box.
6. Modify the fields in the dialog box as needed. Refer to Table 2-1 for the definitions of the **Mainframe Action Fields**.
7. Click **OK** to save the changes.

Deleting a Mainframe Action

To delete a mainframe action:

1. Open the existing WebLogic Process Integrator workflow template definition.
2. From the workflow diagram, double-click the **Task** node for which you want to delete a mainframe action. The Task Properties dialog box displays.
3. Select the desired action in the **Actions** section of the Task Properties dialog box, and click the **Delete** button.
4. A confirmation dialog box displays to prevent accidental deletions. Click **Yes** to confirm the deletion or **No** to cancel the deletion.

   When an action is deleted, all references to the action are removed. The action is removed from all instances of the workflow, including those currently running.

Changing the Sequence of Mainframe Actions

The sequence of mainframe actions can be changed during design time or run time.

To change the sequence of actions:

1. From your workflow diagram, double-click the **Task** node for which you want to change the sequence of a mainframe action. The Task Properties dialog box displays.
2. Click on an action in the **Actions** list.
3. Press the up or down arrow button to move the action’s position up or down in the list.
4. Click on **OK** to save the change to the workflow template definition.

Initiating Workflows from the Mainframe

By working with JAM and WebLogic XML/Non-XML Translator, WebLogic Process Integrator workflows can be initiated from mainframe applications. To initiate workflows from the mainframe, you must:

- Modify the gateway configuration file
- Create a WebLogic Process Integrator workflow with an XML event-driven start node

Example of Initiating a Workflow from the Mainframe

In the following example, a new employee record is created on the mainframe. It is sent to the workflow process to be added the employee database.

The screen data is validated and moved into the employee record copybook. This copybook corresponds to the copybook used to generate the MFL in the WebLogic XML/Non-XML Translator Distributed Program Link (DPL).

The COBOL CICS program in Listing 2-5 illustrates how a workflow may be initiated. The CICS link sends the employee record to the workflow. The `SYSID` directs the `LINK` to the correct remote environment, and the `PROGRAM` name corresponds to the `RNAME` in the `jcrmgw.cfg` `JC_LOCAL_SERVICES` entry.

Listing 2-5 Example of Initiating a Workflow From the Mainframe

```
DATA DIVISION.
 WORKING-STORAGE SECTION.
 01  WS-VARIABLES.
     COPY EMPREC.
 01  WS-CONSTANTS.
     05  EMP-REQUEST                  PIC  X(8)
                                          VALUE 'DPL1SVR'.
     05  REMOTE-SYSID                 PIC  X(4)
                                          VALUE 'BEA1'.
 .
 .
 .
 PROCEDURE DIVISION.
 MAINLINE SECTION.
   PERFORM VALIDATE-SCREEN-RECORD
   MOVE SCREEN-SSN             TO EMP-SSN
   MOVE SCREEN-LAST-NAME       TO EMP-NAME-LAST
   MOVE SCREEN-FIRST-NAME      TO EMP-NAME-FIRST
   MOVE SCREEN-MIDDLE-INIT     TO EMP-NAME-MI
   MOVE SCREEN-STREET          TO EMP-ADDR-STREET
   MOVE SCREEN-STATE           TO EMP-ADDR-STATE
   MOVE SCREEN-ZIP             TO EMP-ADDR-ZIP
   EXEC CICS LINK
      PROGRAM(EMP-REQUEST) SYSID(REMOTE-SYSID)
        COMMAREA(EMP-RECORD) LENGTH(LENGTH OF EMP-RECORD)
          RESP(RESP-CODE)
   END-EXEC
   PERFORM CHECK-RESPONSE
   PERFORM RETURN-TO-CICS
 MAINLINE-EXIT.
   EXIT.
```

Modify the Gateway Configuration File

The JAM gateway configuration file must have a JC_LOCAL_SERVICE entry for each of the services in your configuration. This definition must specify the schema that was created when you set up data translation.

To enable the mainframe to initiate a workflow in WebLogic Process Integrator specify each local service in the JC_LOCAL_SERVICES section of the jcrmgw.cfg file. Each JC_LOCAL_SERVICE that is to work within WebLogic Process Integrator must specify JAMToWLPIHome as its name . This name specifies the Home Interface of the EJB delivered with the JAM plug-in. Within the JAMToWLPIHome service, specify the translation schema to be applied to each service using the schema keyword in the following way:

```
SCHEMA = emprec
```

Listing 2-6 is an example of the JC_LOCAL_SERVICES section of the jcrmgw.cfg file.

Listing 2-6 JC_LOCAL_SERVICES Example

```
#--------------------------------------------------------------
*JC_LOCAL_SERVICES
JAMToWLPIHome RNAME='DPL1SVR'
SCHEMA=emprec
```

For specific information about configuring the JAM gateway, refer to the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*.

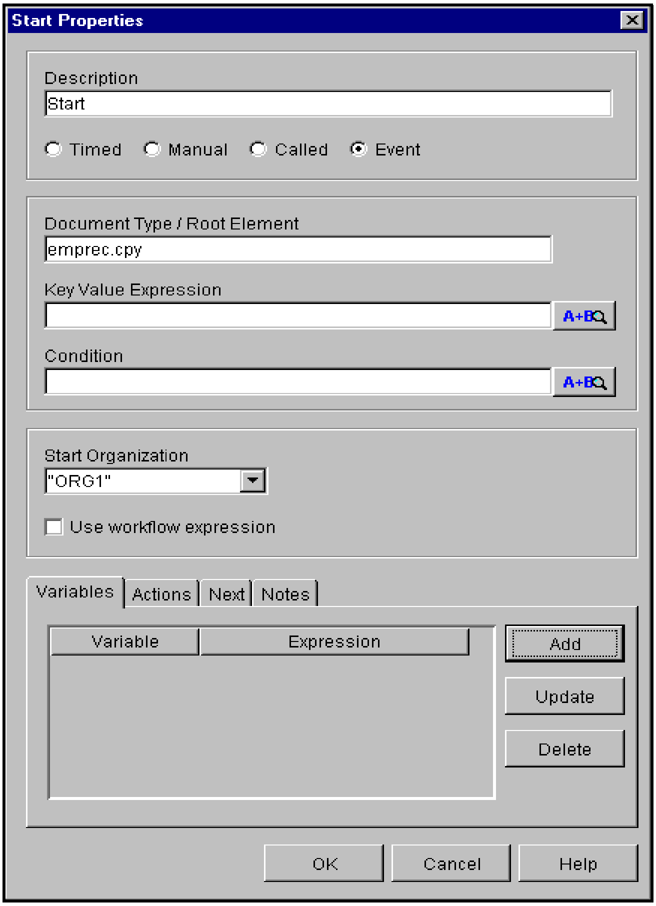
Edit the Workflow Start Node

For a mainframe event or action to initiate a workflow, it must be included in the Start Node of the WebLogic Process Integrator workflow. To edit the workflow Start Node to initiate the workflow, complete the following steps.

1. Double click on the **Start Node** you wish to edit.

2. In the Start Properties dialog box, shown in Figure 2-10, select **Event** as the start method for the workflow.

Figure 2-10 Start Node



Start Properties

Description: Start

○ Timed ○ Manual ○ Called ● Event

Document Type / Root Element: emprec.cpy

Key Value Expression:

Condition:

Start Organization: "ORG1"

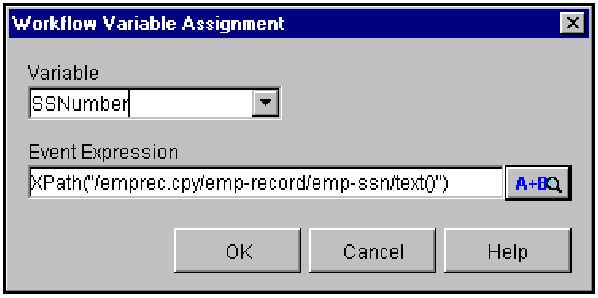
□ Use workflow expression

3. In the Document/Type field, enter the root element name of the XML document. To obtain this name, view the XML document structure in the WebLogic XML/Non-XML Translator Format Builder.

4. To define the variable, select the variable tab and click **Add**.

5. In the Workflow Variable Assignment dialog box, shown in Figure 2-11, enter the name you want to assign to the variable. You may want to use an XPath function to select fields from the XML. For more information about XPath functions, refer to the *BEA WebLogic Process Integrator Studio Guide*.

Figure 2-11 Workflow Variable Assignment Dialog Box



Variable: SSNumber

Event Expression: XPath("/emprec.cpy/emp-record/emp-ssn/text()")

The workflow will be started automatically when data is received from the mainframe.

3 Example of JAM Integrated with the WebLogic Process Integrator

The BEA WebLogic Java Adapter for Mainframe (JAM) software includes three sample workflows designed to illustrate the way JAM works with WebLogic Process Integrator and WebLogic XML/Non-XML Translator. This section describes these workflow templates and gives you step-by-step instructions for running the example.

Action List

To use this example of JAM working with WebLogic Process Integrator, complete the following tasks.

	Your action...	Refer to...
1	Set up the mainframe.	"Step 1: Setting up the Mainframe"
2	Configure and run WebLogic Process Integrator.	"Step 2: Configure and Run WebLogic Process Integrator using WebLogic Server"

	Your action...	Refer to...
3	Create the workflow templates and import the template definitions.	"Step 3: Create the Templates and Import the Template Definitions"
4	Open and activate the workflow template.	"Step 4: Open and Activate the Template"
5	Execute the workflow.	"Step 5: Executing the Workflow"

Prerequisites

Before you can run the JAM/WebLogic Process Integrator scenarios, make sure the following tasks have been completed:

	Your action...	Refer to...
1	Install and configure WebLogic Server for your system.	*BEA WebLogic Server Installation Guide*
2	Install and configure WebLogic Process Integrator for your system.	*BEA WebLogic Process Integrator Installation Guide*
3	Install and configure WebLogic XML/Non-XML Translator for your system.	*BEA WebLogic XML/Non-XML Installation Guide*
4	Install and configure JAM for your system.	*BEA WebLogic Java Adapter for Mainframe Installation Guide and BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*
5	Read the Release Notes for each product to become familiar with any special requirements.	BEA WebLogic product documentation

Note: The instructions presented in this section assume that you have a good working knowledge of BEA WebLogic Process Integrator, BEA WebLogic Server, and BEA XML/Non-XML Translator. You should have successfully installed WebLogic Process Integrator and run a sample workflow prior to running these samples.

Running the WebLogic Process Integrator/JAM Example

This example simulates storage and retrieval of employee records in a mainframe database initiated by workflow processing. The data is entered from a worklist form and translated to XML. The data is then translated to COBOL copybook format and stored in the data base. The employee record can then be retrieved from the database into the workflow. Examples of COBOL mainframe applications (`dpldemoc.cbl`, `dpldemor.cbl`, `dpldemou.cbl`, and `dpldemod.cbl`) can be extracted from the `sample.jar` file in the `sample` directory located in the `<JAM Installation>\examples` directory.

Files for the WebLogic Process Integrator/JAM Example

Table 3-1 describes the files used in the examples. These files are located in the `samples.jar` in the JAM installation directory.

Table 3-1 List of WebLogic Process Integrator/JAM Sample Files

Directory	File	Description
`sample/wlpi`	`jcrmgw2.cfg`	Configuration file for the JAM Gateway
`sample/wlpi`	`create.xml`	Workflow that creates an employee record
`sample/wlpi`	`read.xml`	Workflow that reads an employee record
`sample/wlpi`	`readString.xml`	Workflow that reads as employee record using string data type

Table 3-1 List of WebLogic Process Integrator/JAM Sample Files

Directory	File	Description
`sample/wlpi`	`emprec.cpy`	Copybook for the employee record
`sample`	`dpldemoc.cbl`, `dpldemor.cbl`, `dpldemou.cbl`, and `dpldemod.cbl`	Mainframe files

How to Run the WebLogic Process Integrator/JAM Example

Follow the steps below to run the WebLogic Process Integrator/JAM example. For specific instructions on performing the tasks in WebLogic Process Integrator, WebLogic XML/Non-XML Translator, and WebLogic Server, please refer to the documentation that accompanies those applications.

Step 1: Setting up the Mainframe

- Set up the mainframe programs. COBOL copybooks such as `dpldemoc.cbl`, `dpldemor.cbl`, `dpldemou.cbl`, and `dpldemod.cbl` are examples of the types of programs that you might use.

Step 2: Configure and Run WebLogic Process Integrator using WebLogic Server

1. Move the `jcrmgw.cfg` file from the sample directory to:
   `<BEA.Home>/<install.dir>/config/<domain.name>`
2. Edit the sample `jcrmgw.cfg` file, providing the address and port to work with the Communications Resource Manager (CRM).
3. Start the WebLogic Process Integrator Server.
4. Run WebLogic Process Integrator Studio.

5. Import the emprec.cpy copybook using WebLogic XML/Non-XML Translator Format Builder and store it in the repository as described in the "Using Mainframe Applications with Workflows" section of this guide.

Step 3: Create the Templates and Import the Template Definitions

This example provides three workflow templates for you to work with. The templates are listed and described in Table 3-2

Table 3-2 Provided Workflow Templates

Workflow Template	File Name	Description
`JAM create`	`create.xml`	Creates an employee record
`JAM read`	`read.xml`	Reads an employee record
`JAM readString`	`readString.xml`	Reads and employee record using string data type

To create the template and import the template definition for `JAM create`:

1. Create a new template called `JAM create`.
2. Select the template from the tree view in the left pane and click the right mouse button.
3. Choose **Import Template Definition**.
4. Select the corresponding definition file: `create.xml`.
5. Click **OK** to close the message box.
6. Repeat steps 1 through 5 for:
   - `JAM read` from the definition file `read.xml`
   - `JAM readString` from the definition file `readString.xml`

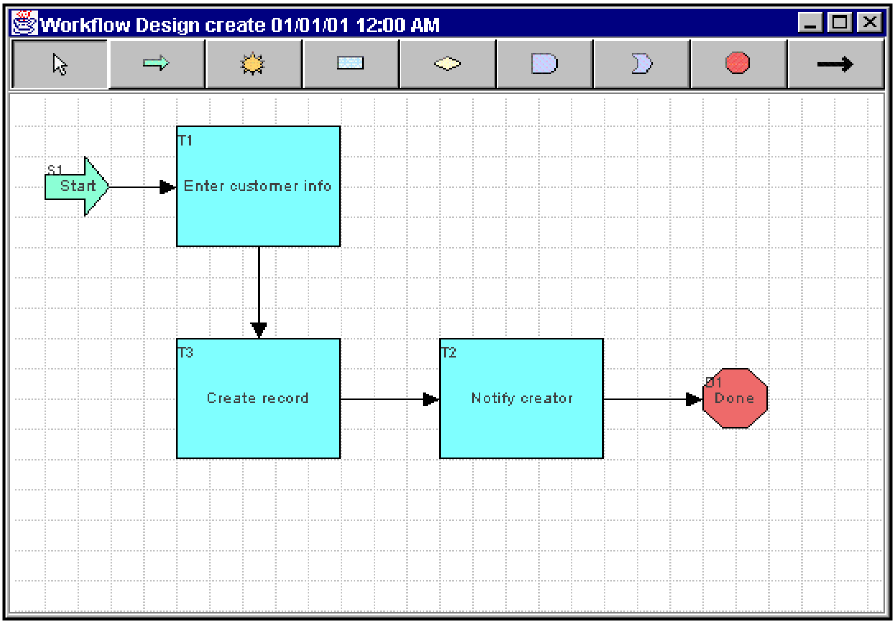
Step 4: Open and Activate the Template

To enable each workflow template, open and activate the template using the following steps:

1. Select the `JAM create` template definition imported in the previous step from the tree view and click the right mouse button.

2. Choose **Open**. The workflow created for this sample application displays, as shown in Figure 3-1.

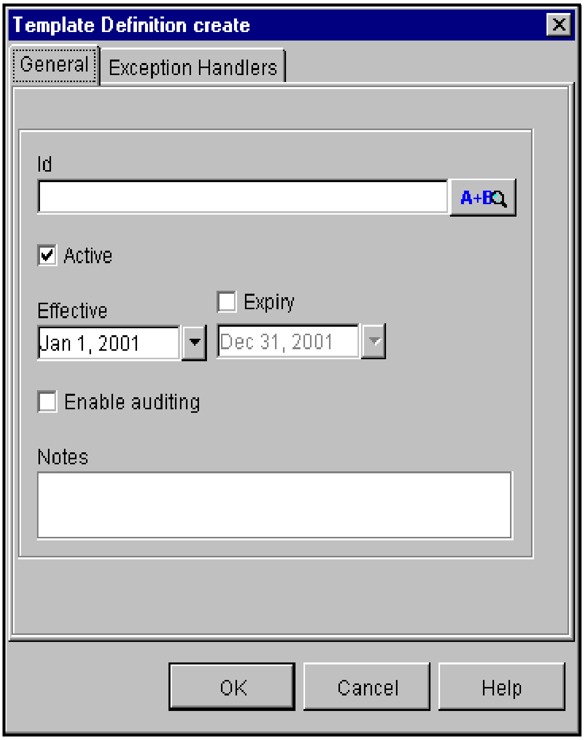
Figure 3-1 Workflow for JAM create



3. Select the JAM create template definition again from the tree view and click the right mouse button.

4. Choose **Properties**. The Template Definition properties dialog box displays, as shown in Figure 3-2.

Figure 3-2 Template Definition `create`



☑ Active

Effective: Jan 1, 2001

□ Expiry: Dec 31, 2001

□ Enable auditing

5. Click **Active** to activate the template and click **OK**.

6. Select the `JAM create` template definition a third time from the tree view and click the right mouse button again.

7. Choose **Save** to save the template definition with the changes you made.

8. Repeat Steps 1 through 7 for `JAM read` and `JAM readString`.

Step 5: Executing the Workflow

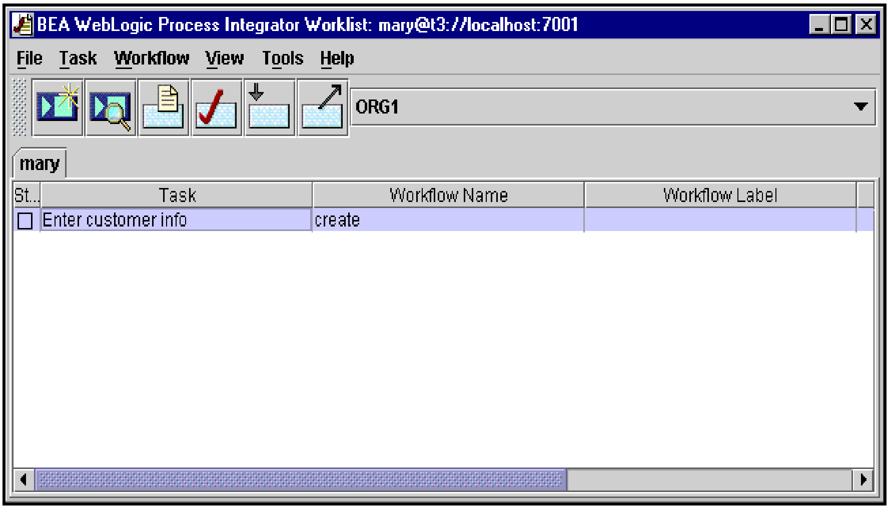
This section provides step-by-step instructions for executing workflows to create an employee record and to read an employee record.

Execute the JAM create Workflow

The `JAM create` workflow creates an employee record. To start the workflow from the Weblogic Process Integrator Worklist, complete the following steps:

1. Start WebLogic Process Integrator Worklist and choose **Workflow→Start a Workflow**.
2. Select `JAM create`. The template opens and the first task is displayed, as shown in Figure 3-3.

Figure 3-3 `JAM create` Worklist

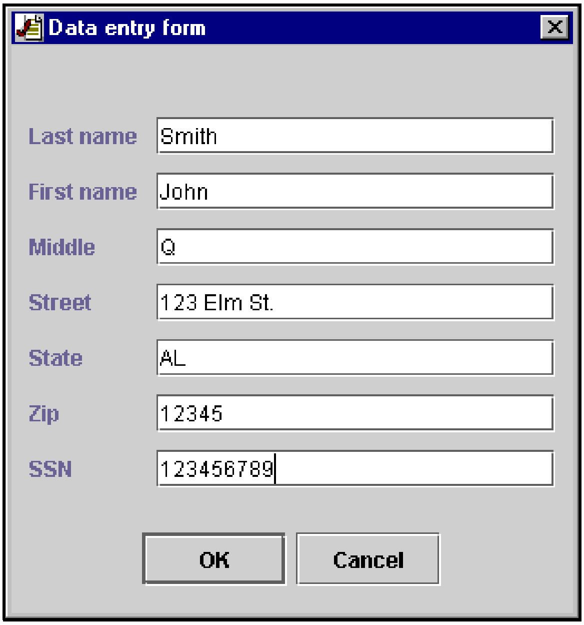

BEA WebLogic Process Integrator Worklist: mary@t3://localhost:7001
File
Task
Workflow
View
Tools
Help
ORG1
mary

St...	Task	Workflow Name	Workflow Label
□	Enter customer info	create	

3. Select the **Enter Customer Info** task and click the right mouse button.

4. Select **Execute**. The **Enter Customer Info** dialog displays, as shown in Figure 3-4.

Figure 3-4 Data Entry Form



Last name Smith

First name John

Middle Q

Street 123 Elm St.

State AL

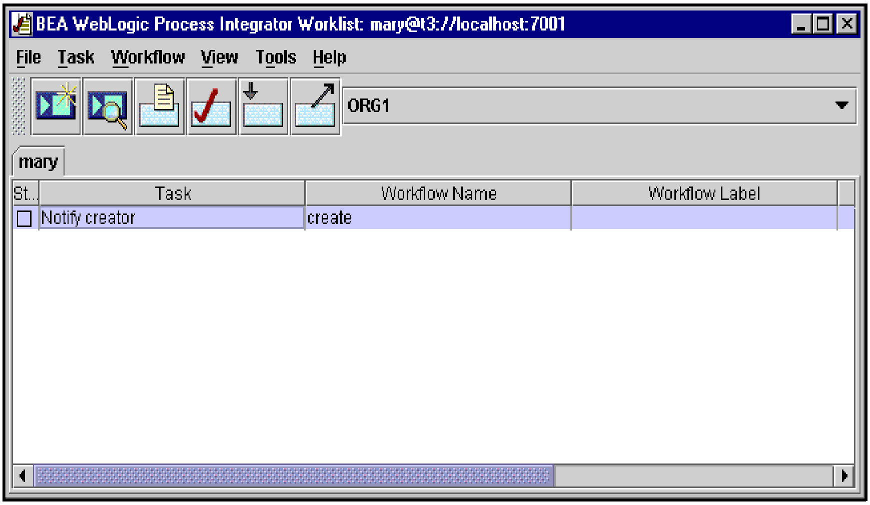
Zip 12345

SSN 123456789

5. Enter the employee information and click **OK**. The task is started and the workflow runs.

6. A new task, **Notify Creator** will appear on the worklist, as shown in Figure 3-5. Select the **Notify Creator** task.

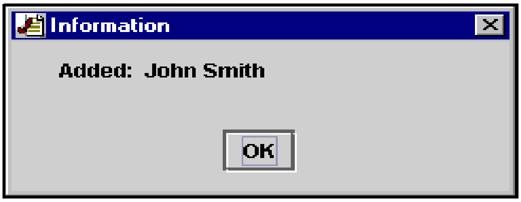
Figure 3-5 Notify Creator Task


BEA WebLogic Process Integrator Worklist: mary@t3://localhost:7001
File
Task
Workflow
View
Tools
Help
ORG1
mary

St...	Task	Workflow Name	Workflow Label
□	Notify creator	create	

7. Select **Execute**. When the **Information** dialog appears, as shown in Figure 3-6, click **OK** to close the dialog box. The workflow is now complete.

Figure 3-6 Information Dialog window



Added: John Smith

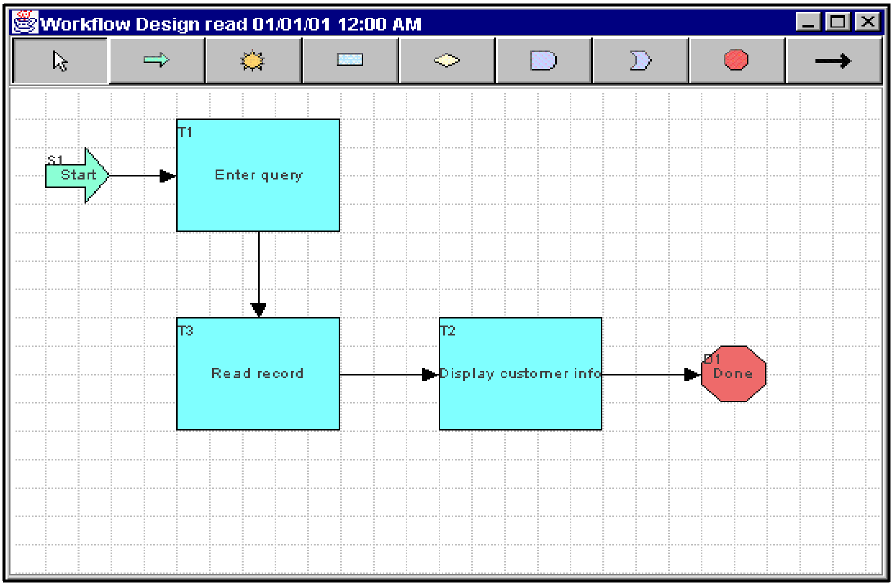
Execute the JAM read and JAM readString Workflow

The `JAM read` and `JAM readString` workflows read and display an employee record for the mainframe database. The `JAM readString` workflow is functionally the same as the `JAM read` workflow. The difference is that the `JAM readString` workflow uses String variables internally to contain the XML used for data translation.

To execute these workflows from the WebLogic Process Integrator Worklist, complete the following steps:

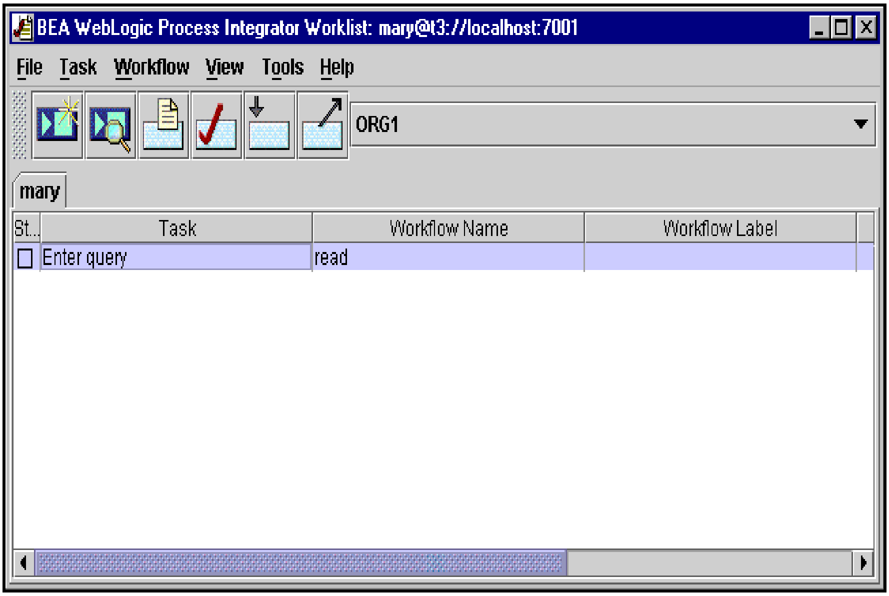
1. Start WebLogic Process Integrator Worklist and choose **Workflow→Start a Workflow**.
2. Select `JAM read` or `JAM readString`. The template opens, as shown in Figure 3-7 and the first task is displayed.

Figure 3-7 `JAM read` or `JAM readString` Workflow



3. Select the **Enter Query** task, as shown in Figure 3-8, and click the right mouse button.

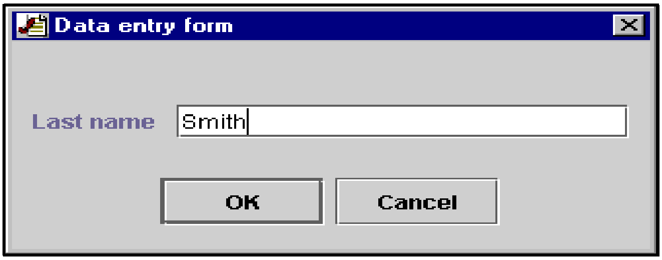
Figure 3-8 `JAM read` or `JAM readString` Worklist


BEA WebLogic Process Integrator Worklist: mary@t3://localhost:7001
File
Task
Workflow
View
Tools
Help
ORG1
mary

St...	Task	Workflow Name	Workflow Label
□	Enter query	read	

4. Select **Execute**. The **Data entry form** dialog box displays, as shown in Figure 3-9.

Figure 3-9 Data entry form

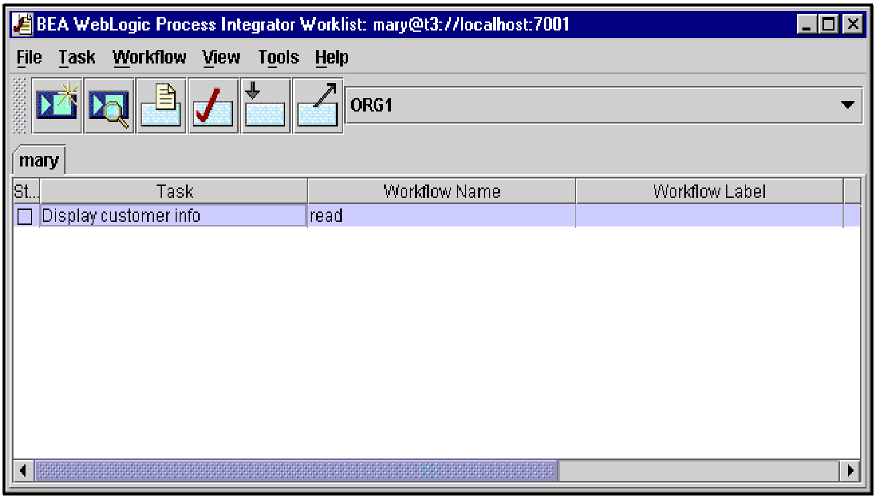


Last name: Smith

5. Enter the last name of the employee to be read. For example, the last name entered when you created an employee record was Smith. Click **OK**.

6. A new task, **Display Customer Information**, will appear on the worklist. Select the **Display Customer Information** task, as shown in Figure 3-10.

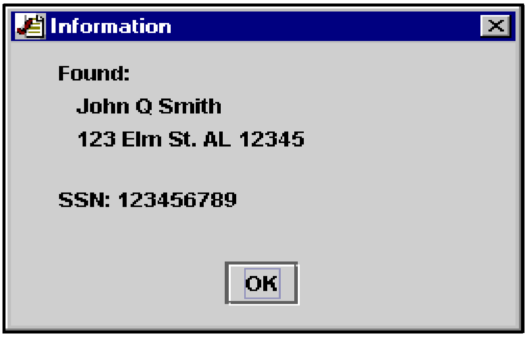
Figure 3-10 Notify Creator Task


BEA WebLogic Process Integrator Worklist: mary@t3://localhost:7001
File
Task
Workflow
View
Tools
Help
ORG1
mary

St...	Task	Workflow Name	Workflow Label
□	Display customer info	read	

7. Select **Execute**. When the Information dialog box appears, as shown in Figure 3-11, click **OK** to close the dialog box. The workflow is now complete.

Figure 3-11 Information Dialog window



Found:
John Q Smith
123 Elm St. AL 12345

SSN: 123456789

Index