BEA WebLogic Java Adapter for Mainframe™

Workflow Processing Guide for WebLogic Integration 2.1

Version 4.2
November 2001

Copyright

Restricted Rights Legend

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 WebLogic Java Adapter for Mainframe Workflow Processing Guide

Part Number	Date	Software Version
N/A	November 2001	4.2

Contents

4. Running the WebLogic Integration Examples

CHAPTER

1 Workflow Processing with BEA WebLogic Java Adapter for Mainframe

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

WebLogic JAM can be used with WebLogic Integration in the following ways:

- WebLogic Integration 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 Integration using an XML event interface.

This section provides information on the following subjects:

Introduction to Business Process Management with WebLogic Integration

WebLogic Integration 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 Integration 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 Integration uses XML as its standard data format. To access data from a legacy system in WebLogic Integration, the data must be converted to XML first. WebLogic Integration provides data integration by allowing you to translate data from binary to XML or from XML to binary.

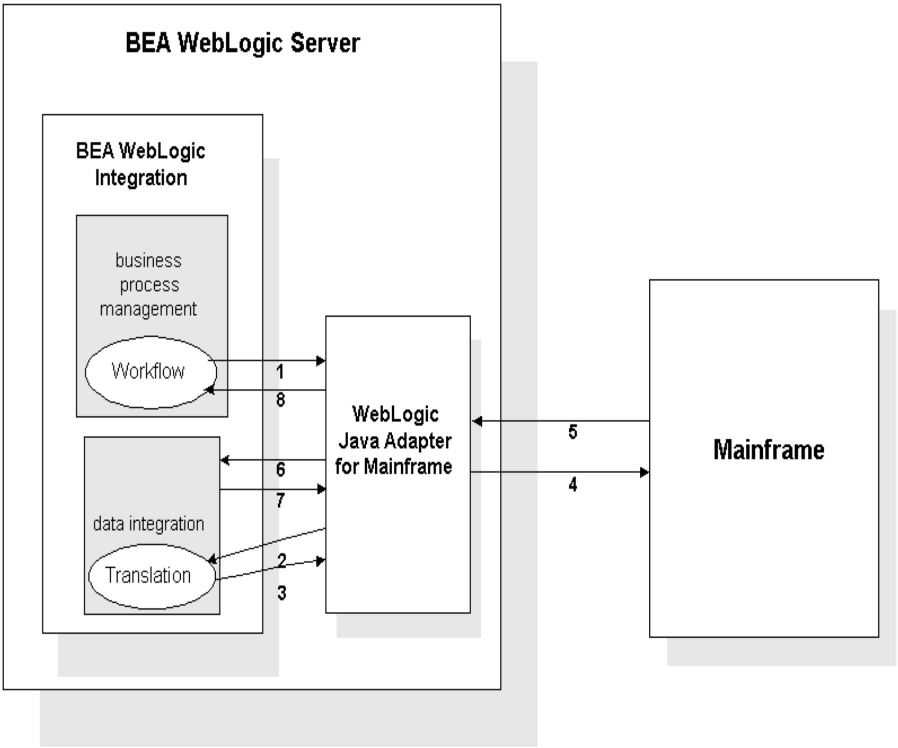
This section provides information on the following subjects:

- Understanding WebLogic Integration Interaction with the Mainframe
- Accessing Mainframe Applications Using WebLogic JAM

Understanding WebLogic Integration Interaction with the Mainframe

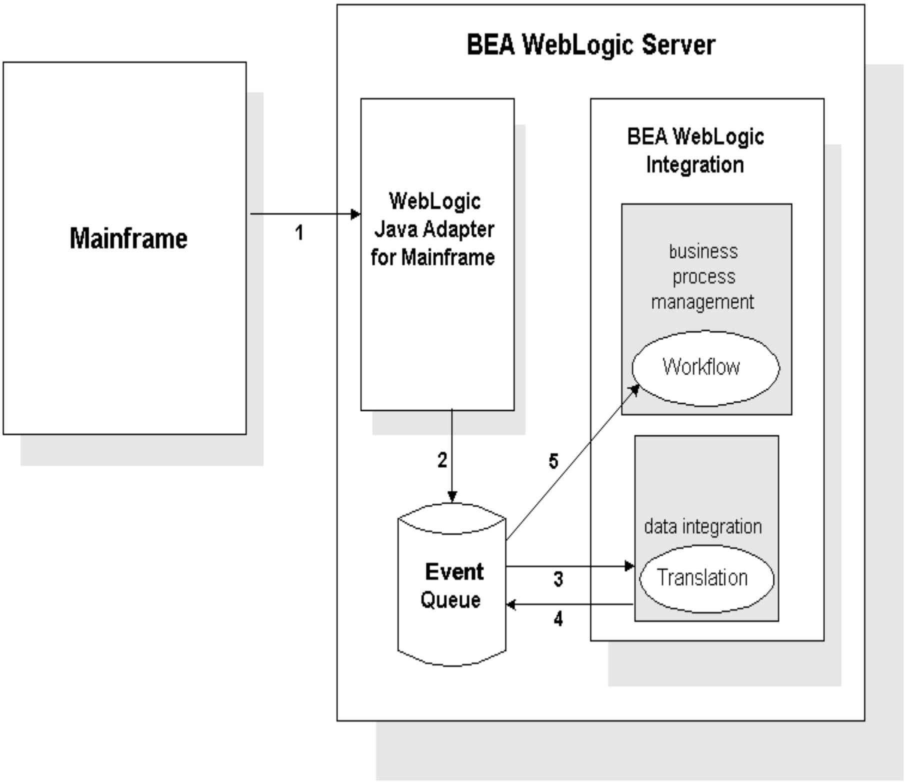
Figure 1-1 illustrates the flow of data when a WebLogic Integration workflow accesses mainframe applications through WebLogic JAM. When WebLogic Integration sends data to the mainframe application, the WebLogic JAM Gateway converts the XML data received from WebLogic Integration to the binary format expected by the mainframe application. The WebLogic JAM Gateway invokes the data integration component of WebLogic Integration to perform this conversion. It then invokes the mainframe program and receives a response. This response is sent to WebLogic Integration to convert the data from binary format to XML.

Figure 1-1 Workflows Accessing Mainframe Applications



WebLogic JAM also allows a workflow to be initiated by an event triggered by a mainframe application. When WebLogic 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 Integration, then WebLogic JAM puts the mainframe data on a WebLogic Integration queue. WebLogic Integration pulls the data off the WebLogic Server event queue and recognizes that translation is required. The translation is performed, and the data is returned to the queue. WebLogic Integration 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



Accessing Mainframe Applications Using WebLogic JAM

WebLogic JAM seamlessly plugs in to WebLogic Integration, allowing business analysts to include mainframe applications in business processes. Within the WebLogic Integration framework, WebLogic JAM uses WebLogic Integration to translate binary data to and from XML.

XML is quickly becoming the standard for exchanging information between applications and is invaluable in integrating disparate applications. WebLogic Integration provides for an exchange of information between applications by supporting data translations between binary formats from legacy systems and XML. WebLogic JAM works with WebLogic Integration’s data translation feature to convert data between mainframe copybook formats and XML format that can be used by the process engine.

CHAPTER

2 Installing WebLogic JAM Plug-In

This section provides information on the following subjects:

- Preparing to Install WebLogic JAM Plug-in
- Installing the WebLogic JAM Plug-in
- Updating Your license.bea File

Preparing to Install WebLogic JAM Plug-in

Before installing the WebLogic JAM Plug-in, you must:

- Download WebLogic JAM from the Web
- Verify Software Requirements

Download WebLogic JAM from the Web

An evaluation copy of WebLogic JAM is available for download from the BEA corporate Web site at http://commerce.bea.com/downloads/weblogic_server.jsp#jam. When you download WebLogic JAM, you get a built-in 30-day evaluation license. Because the WebLogic JAM Plug-in is included in the WebLogic JAM license, it will be updated when you update the WebLogic JAM license.

Verify Software Requirements

The WebLogic JAM Plug-in is dependent on the WebLogic Integration software. Verify that WebLogic Integration has been properly installed and configured before installing WebLogic JAM. For more information on installing and configuring WebLogic Integration, refer to *Installing WebLogic Integration*.

Installing the WebLogic JAM Plug-in

The WebLogic JAM Plug-in is installed as part of WebLogic JAM, provided WebLogic Integration is already installed. Refer to the *BEA WebLogic Java Adapter for Mainframe Installation Guide* for step-by-step installation instructions.

If you do not have WebLogic Integration installed prior to installing WebLogic JAM, the Plug-in will not be installed. In order to install the Plug-in, you must do the following:

1. Uninstall WebLogic JAM (refer to the *BEA WebLogic Java Adapter for Mainframe Installation Guide* for instructions).
2. Install WebLogic Integration (refer to *Installing WebLogic Integration* for step-by-step instructions).
3. Re-install WebLogic JAM. The installation program will find the installation of WebLogic Integration and automatically install the Plug-in.

Updating Your license.bea File

After you have installed WebLogic JAM, you must update the WebLogic JAM `license.bea` file. For more information about updating the `license.bea` file, refer to the *BEA WebLogic Java Adapter for Mainframe Installation Guide*.

CHAPTER

3 Using Mainframe Applications with Workflows

This section provides information on the following subjects:

- Before You Begin
- Using WebLogic JAM with Workflow Processing

Before You Begin

To be able to use mainframe applications with workflows, be sure that you have:

- Installed and configured WebLogic Server.
- Installed and configured WebLogic Integration.
- Installed and configured BEA WebLogic Java Adapter for Mainframe (WebLogic JAM) and the WebLogic JAM Plug-in.
- Reviewed and understand the steps for creating and editing workflows in WebLogic Integration.

Using WebLogic JAM with Workflow Processing

The WebLogic JAM Plug-in is a component that allows BEA WebLogic Integration to interact with mainframe applications using the WebLogic JAM Gateway. It runs with WebLogic Integration in the WebLogic Server environment.

Use the following steps for workflow processing with WebLogic JAM:

- Step 1: Set Up the WebLogic JAM Plug-in Components
- Step 2: Set Data Translation
- Step 3: Set Up the Workflow with WebLogic JAM for Mainframe Services

Step 1: Set Up the WebLogic JAM Plug-in Components

To use WebLogic JAM to access mainframe applications during workflow processing, you must set up the WebLogic JAM Plug-in components by completing the following tasks.

1. Add the entries shown below to the “WLI” application definition in the WebLogic Server configuration file (`config.xml`). See Listing 3-1 for an example of the “WLI” application definition section.

   ```
   <EJBComponent Name="jampi.jar"
      Targets="myserver" URI="jampi.jar"DeploymentOrder="21"/>
   ```

   ```
   <WebAppComponent Name="com.bea.jam.JamPlugin"
      Targets="myserver" URI="jampi.war"DeploymentOrder="22"/>
   ```

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

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

   Where *`install.dir`* is where WebLogic Integration is installed (usually `wlintegration2.1`) and *`domain.name`* is the domain from which WebLogic Integration is started. For example:

   ```
   C:/BEA/wlintegration2.1/config/bpmdomain
   ```

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.

2. Review your server configuration in the `config.xml` file. If the server name is not `myserver`, the `Targets` entry needs to be modified with the name of your server.

   Note: The `config.xml` file can control the order in which the `.jar` files are deployed. The `EJBComponent` and `WebAppComponent` fields include a `DeploymentOrder` attribute, which indicates the order in which a `.jar` file is deployed with respect to the other `.jar` files.

Listing 3-1 shows an example of the WebLogic Integration section of the WebLogic Server configuration file (`config.xml`).

Listing 3-1 WebLogic Integration Configuration Section of the WebLogic Server config.xml

```
<Application Name="WLI" Path="c:\bea\wlintegration2.1/lib">

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

<EJBComponent Name="wlpi-ejb.jar" Targets="myserver"
URI="wlpi-ejb.jar" DeploymentOrder="1"/>

<EJBComponent Name="wlpi-master-ejb.jar" Targets="myserver"
URI="wlpi-master-ejb.jar" DeploymentOrder="2"/>

<EJBComponent Name="wlpi-mdb-ejb.jar" Targets="myserver"
URI="wlpi-mbd-ejb.jar" DeploymentOrder="3"/>

<EJBComponent Name="pobean.jar" Targets+"myserver"
URI="pobean.jar DeploymentOrder="4"/>

<EJBComponent Name="jampi.jar: Targets="myserver"
URI="jampi.jar" DeploumentOrder="5"/>

<WebAppComponent Name="com.bea.jam.JamPlugin"
Targets="myserver" URI="jampi.war" DeploymentOrder="6"/>

</Application>
```

Step 2: Set 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 Integration 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 Integration repository.

The data integration design-time component of WebLogic Integration (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 Integration converts instances of this layout to and from XML.

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

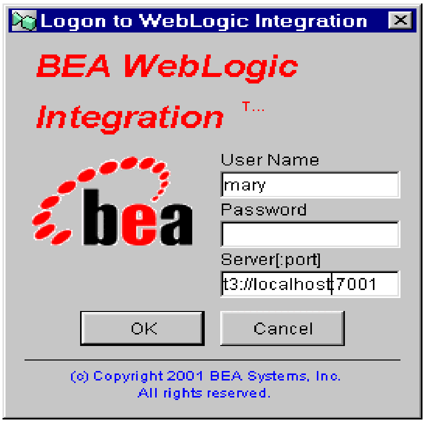
1. Obtain the COBOL copybook(s) or C header file(s) used by your mainframe application. For more information about obtaining the COBOL copybook, refer to the *BEA WebLogic Java Adapter for Mainframe Programming Guid*e.

2. Import the copybooks(s) or C header file(s) into the WebLogic Integration repository using the following steps:

   a. Choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.1→Format Builder to launch the WebLogic Integration Format Builder.

   b. Choose Repository→Login. The WebLogic Integration Repository Login dialog box displays (see Figure 3-1).

Figure 3-1 WebLogic Integration Repository Login



User Name: mary

Password:

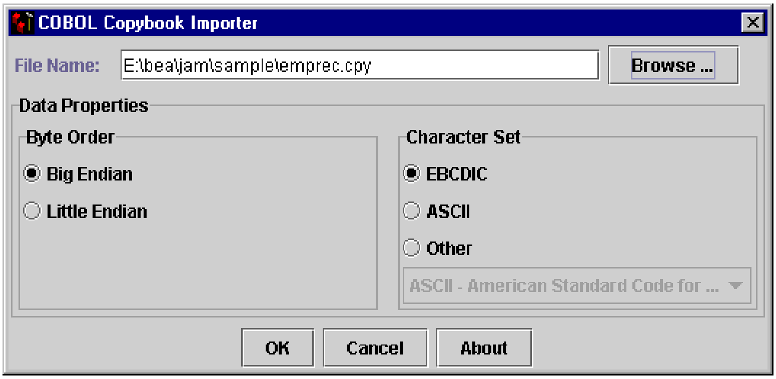
Server[:port]: t3://localhost:7001

(c) Copyright 2001 BEA Systems, Inc.
All rights reserved.

c. Enter your user name and password. If your server is different than the default, enter your server address.

d. Choose Tools→Import→COBOL Copybook Importer (if you are working with a copybook) or C Struct Importer (if you are working with a C Struct). The appropriate dialog box displays (Figure 3-2 shows the COBOL Copybook Importer dialog box. Figure 3-3 shows the C Struct Importer dialog box.)

Figure 3-2 WebLogic Integration COBOL Copybook Importer Dialog Box



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

Data Properties

Byte Order

● Big Endian

○ Little Endian

Character Set

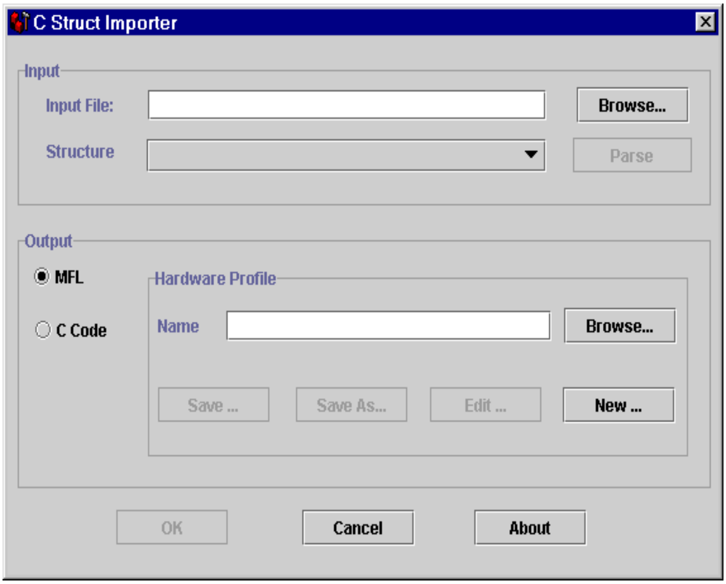
● EBCDIC

○ ASCII

○ Other

ASCII - American Standard Code for ...

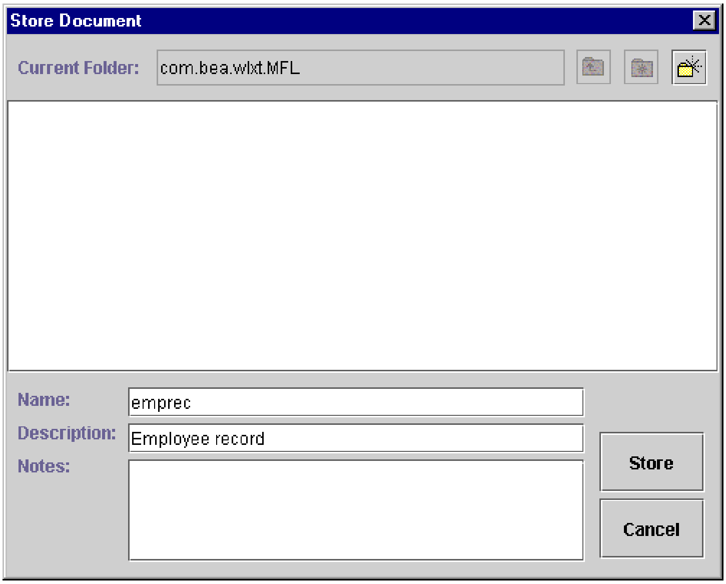
Figure 3-3 WebLogic Integration C Struct Importer Dialog Box.



e. Fill in the fields in the dialog box as appropriate for your environment.

f. After your file has been imported, choose Repository→Store As. The Store Document dialog box displays (see Figure 3-4).

Figure 3-4 WebLogic Integration Store Document Dialog Box



Current Folder: com.bea.wlxt.MFL

Name: emprec

Description: Employee record

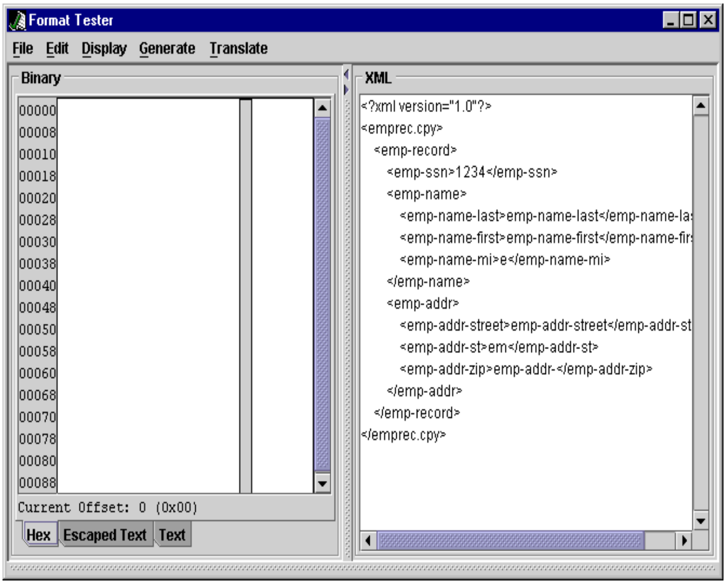
Notes:

g. Enter your name for this document in the **Name** field. 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** to save the document in the WebLogic Integration Repository.

i. Choose Tools→Test. The Format Tester opens (see Figure 3-5).

Figure 3-5 Format Tester



```
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>
```

j. Choose Generate→XML to display a sample of the XML you will use during the creation of your workflow.

Step 3: Set Up the Workflow with WebLogic JAM for Mainframe Services

Determine how your workflow will use mainframe services. You have the following choices:

- Request mainframe services from WebLogic Integration.
- Initiate workflows in WebLogic Integration from the mainframe.

Requesting Mainframe Services from the Workflow

To request mainframe services from WebLogic Integration, you must:

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

Modify the Gateway Configuration File

The WebLogic 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 WebLogic JAM gateway configuration file is located at:

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

In this address:

`install.dir`
is where WebLogic Integration is installed

`domain.name`
is the domain from which WebLogic Integration was started

To access mainframe applications from a WebLogic Integration workflow, specify each remote service in the JC_REMOTE_SERVICES section of the WebLogic JAM gateway (`jcrmgw.cfg`) file. For specific information about configuring the WebLogic 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 3-2 is an example of the `JC_REMOTE_SERVICES` section of the `jcrmgw.cfg` file.

Listing 3-2 `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 Integration workflows in the following ways:

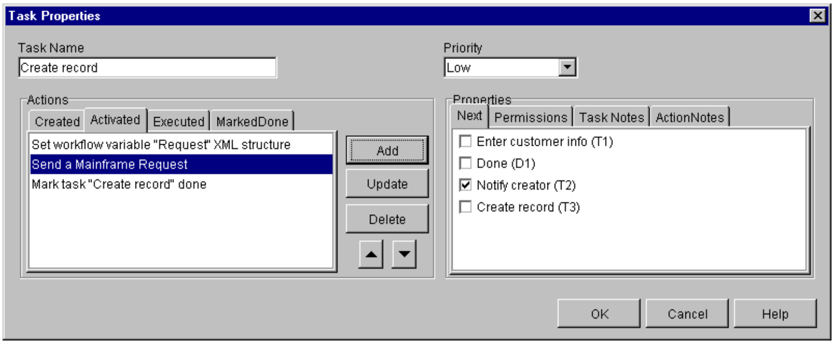
- Adding Mainframe Actions
- Updating Mainframe Actions
- Deleting a Mainframe Action
- Changing the Sequence of Mainframe Actions

Adding Mainframe Actions

Mainframe actions can be added during design time from the WebLogic Integration Studio. 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 3-6.

Figure 3-6 Task Properties



Task Properties

Task Name: Create record

Priority: Low

Actions (Activated):
- Set workflow variable "Request" XML structure
- Send a Mainframe Request
- Mark task "Create record" done

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

3. In the Task Properties dialog box shown in Figure 3-6, 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 3-1.

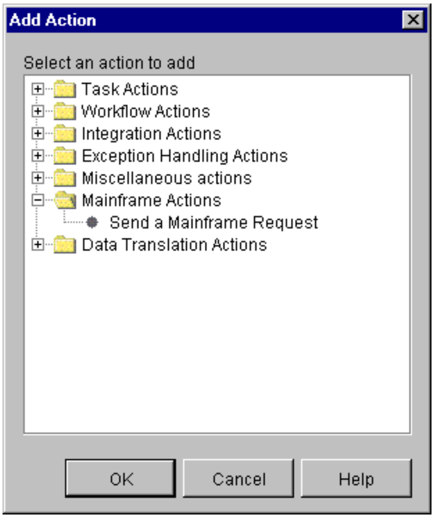
Table 3-1 Business Process Management Action Definitions

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.

See *Using the Studio* in the WebLogic Integration 2.1 documentation for more information about the execution of workflows.

4. Click **Add** to display the Add Action dialog box shown in Figure 3-7.

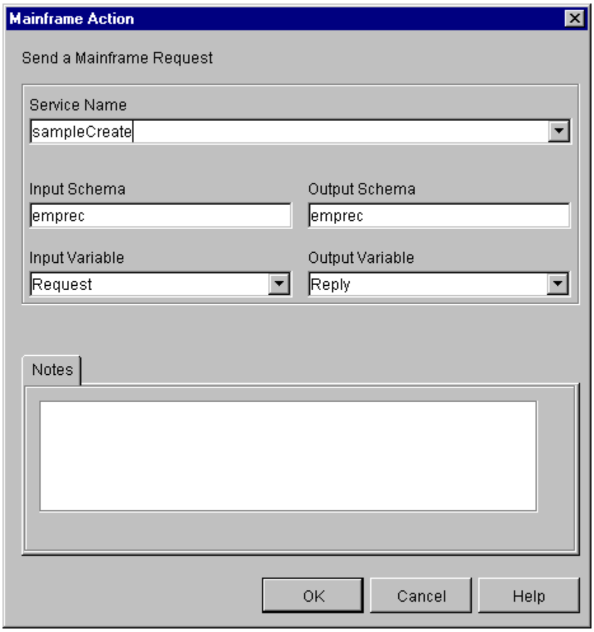
Figure 3-7 Adding Mainframe Actions



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 3-8. A description of the input fields for this dialog box is shown in Table 3-2.

Figure 3-8 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

Notes

Table 3-2 lists the fields used for adding and updating mainframe actions when using WebLogic JAM with a WebLogic Integration workflow.

Table 3-2 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 WebLogic 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.

Field	Description
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 WebLogic 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.
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 WebLogic 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 Integration Studio. 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 3-6, select the appropriate **Actions** tab to designate when the mainframe action is executed. The Actions are defined in Table 3-1.
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 3-1 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 Integration 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 WebLogic JAM and WebLogic Integration, workflows can be initiated from mainframe applications. To initiate workflows from the mainframe, you must:

- Modify the WebLogic JAM Configuration
- Edit the Workflow 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 Step 2 of the Using WebLogic JAM with Workflow Processing section.

The COBOL CICS program in Listing 3-3 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 service name of a WebLogic Integration event entry in the WebLogic JAM configuration.

Listing 3-3 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 WebLogic JAM Configuration

The WebLogic 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 Integration, 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 Integration must specify JAMToWLPIHome as its name. This name specifies the Home Interface of the EJB delivered with the WebLogic 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 3-4 is an example of the JC_LOCAL_SERVICES section of the jcrmgw.cfg file.

Listing 3-4 JC_LOCAL_SERVICES Example

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

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

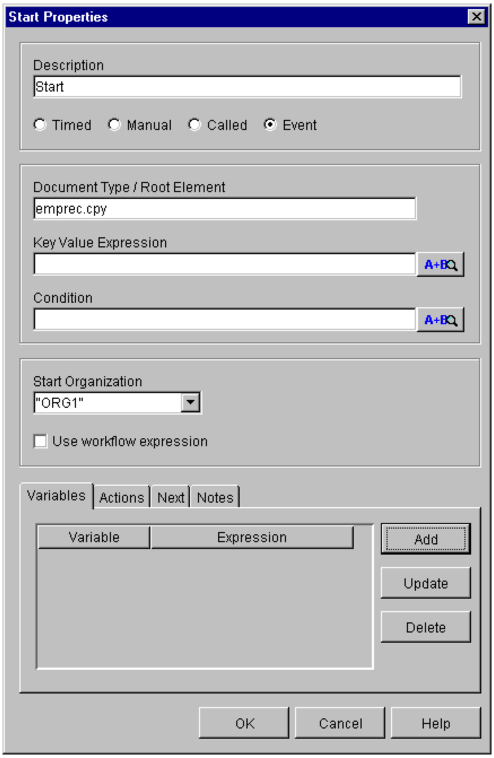
Edit the Workflow Start Node

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

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

2. In the Start Properties dialog box, shown in Figure 3-9, select **Event** as the start method for the workflow.

Figure 3-9 Start Node



Start Properties

Description: Start

○ Timed ○ Manual ○ Called ● Event

Document Type / Root Element: emprec.cpy

Key Value Expression:

Condition:

Start Organization: "ORG1"

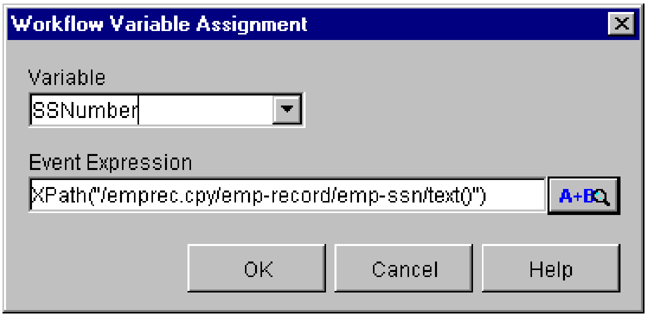
□ Use workflow expression

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 Integration 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 3-10, 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 *Using the Studio* in the WebLogic Integration 2.1 documentation.

Figure 3-10 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.

CHAPTER

4 Running the WebLogic Integration Examples

The BEA WebLogic Java Adapter for Mainframe (WebLogic JAM) Plug-in software includes sample workflows designed to illustrate the way WebLogic JAM works with BEA WebLogic Integration. This section describes these workflow templates and gives you step-by-step instructions for running the example. This section contains information on the following subjects:

- What is Included in the Sample
- Running the BEA WebLogic Integration/WebLogic JAM Example
  - Step 1: Set up the Mainframe
  - Step 2: Configure and Run WebLogic Integration using WebLogic Server
  - Step 3: Create the Templates and Import the Template Definitions
  - Step 4: Open and Activate the Template
  - Step 5: Execute the Workflow

What is Included in the Sample

Table 4-1 describes the files used in the examples. These files are located in the `samples.jar` in the WebLogic JAM installation directory.

Table 4-1 List of WebLogic Integration/WebLogic JAM Sample Files

Directory	File	Description
`sample/wlpi`	`jcrmgw2.cfg`	Configuration file for the WebLogic 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
`sample/wlpi`	`emprec.cpy`	Copybook for the employee record
`sample`	`dpldemoc.cbl`, `dpldemor.cbl`, `dpldemou.cbl`, and `dpldemod.cbl`	Mainframe files

Running the BEA WebLogic Integration/WebLogic 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.

Follow the steps below to run the BEA WebLogic Integration/JAM example. For specific instructions on performing the tasks in WebLogic Integration and WebLogic Server, please refer to the documentation that accompanies those applications.

Step 1: Set 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 Integration 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 Integration Server.
4. Run WebLogic Integration.
5. Import the emprec.cpy copybook using WebLogic Integration and store it in the repository as described in Using Mainframe Applications with Workflows.

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 4-2

Table 4-2 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 `WebLogic 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:
   - `WebLogic JAM read` from the definition file `read.xml`
   - `WebLogic JAM readString` from the definition file `readString.xml`

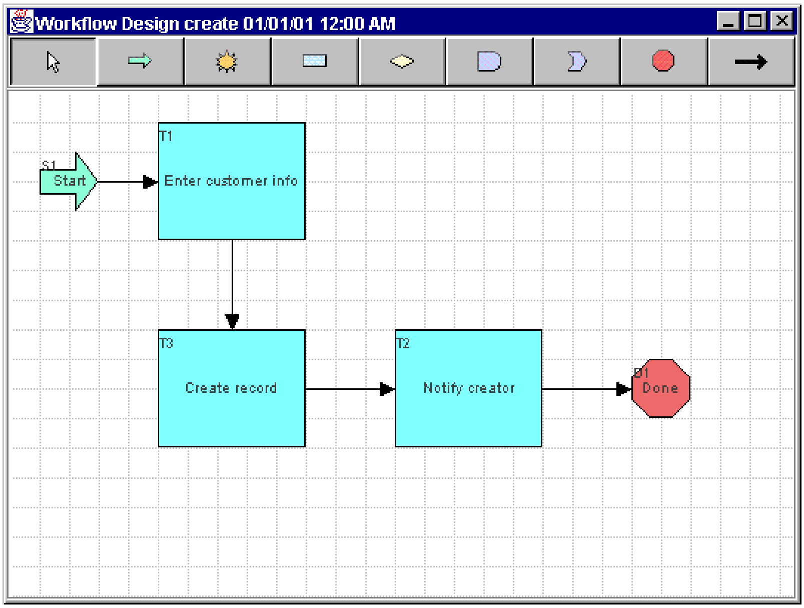
Step 4: Open and Activate the Template

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 4-1.

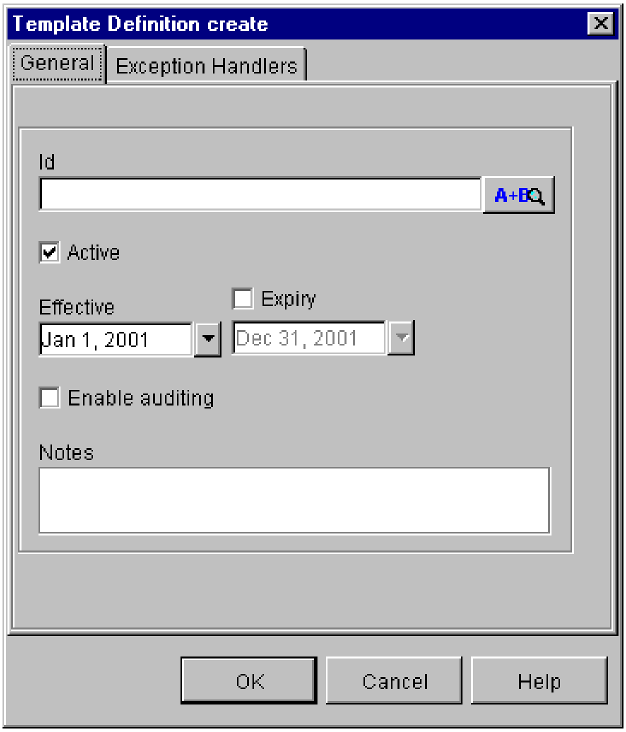
Figure 4-1 Workflow for JAM create



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 4-2.

Figure 4-2 Template Definition `create`



Id

☑ Active

Effective Jan 1, 2001

□ Expiry Dec 31, 2001

□ Enable auditing

Notes

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: Execute the Workflow

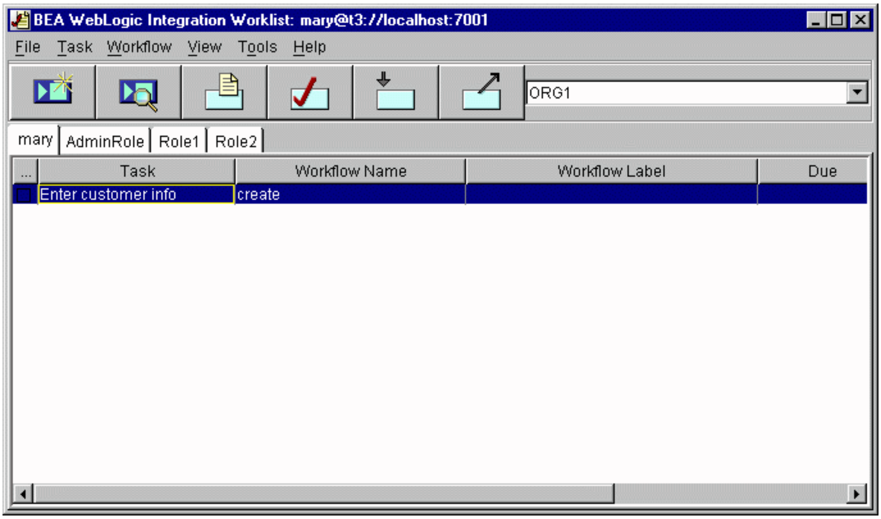
This section provides step-by-step instructions for executing workflows to create an employee record and to read an employee record.

Execute the JAM create Workflow

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

1. Start WebLogic Integration Worklist and choose **Workflow→Start a Workflow**.
2. Select `JAM create`. The template opens and the first task is displayed, as shown in Figure 4-3.

Figure 4-3 JAM create Worklist



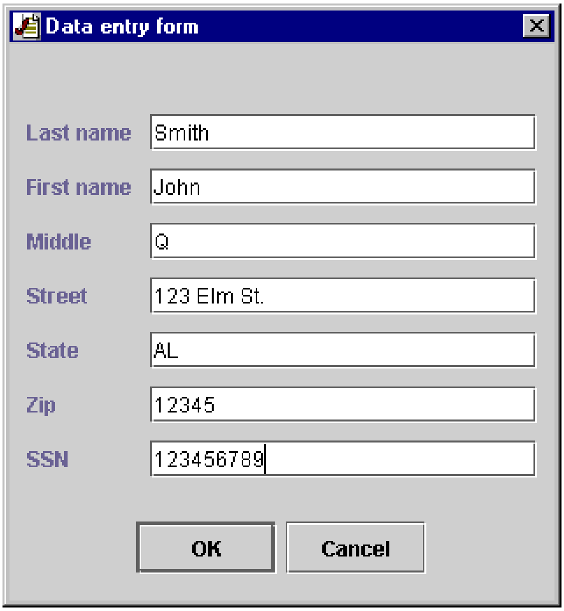
ORG1

...	Task	Workflow Name	Workflow Label	Due
	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 4-4.

Figure 4-4 Data Entry Form

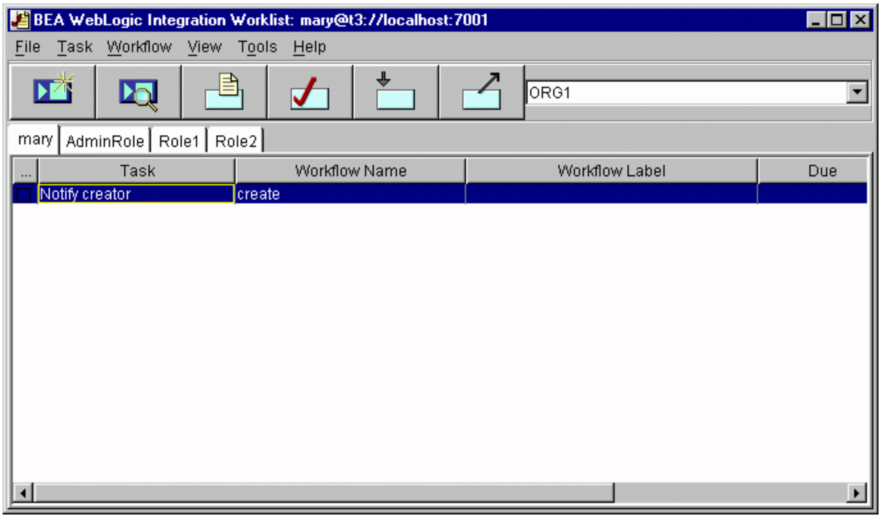


Last name	Smith
First name	John
Middle	Q
Street	123 Elm St.
State	AL
Zip	12345
SSN	123456789

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 4-5. Select the **Notify Creator** task.

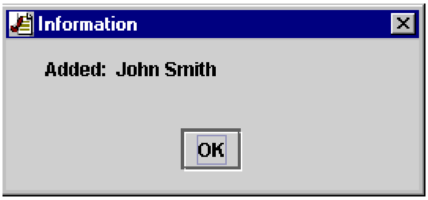
Figure 4-5 Notify Creator Task



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

7. Select **Execute**. When the **Information** dialog appears, as shown in Figure 4-6, click **OK** to close the dialog box. The workflow is now complete.

Figure 4-6 Information Dialog window



Added: John Smith

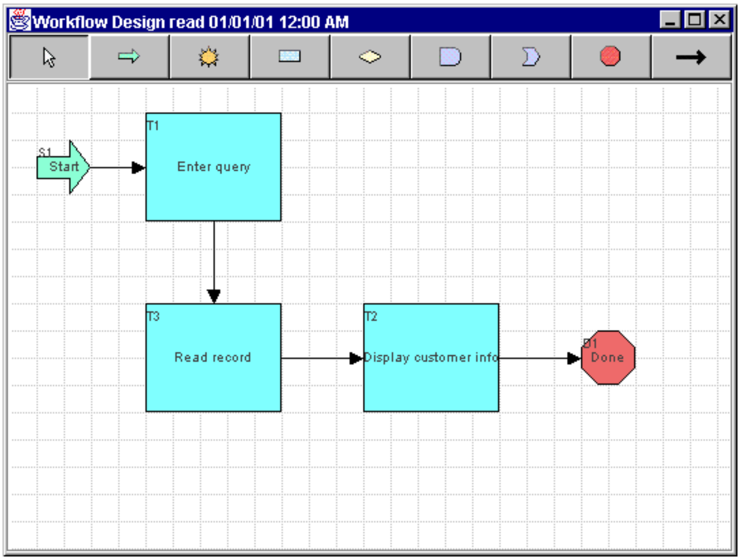
Execute the JAM read and JAM readString Workflow

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 Integration Worklist, complete the following steps:

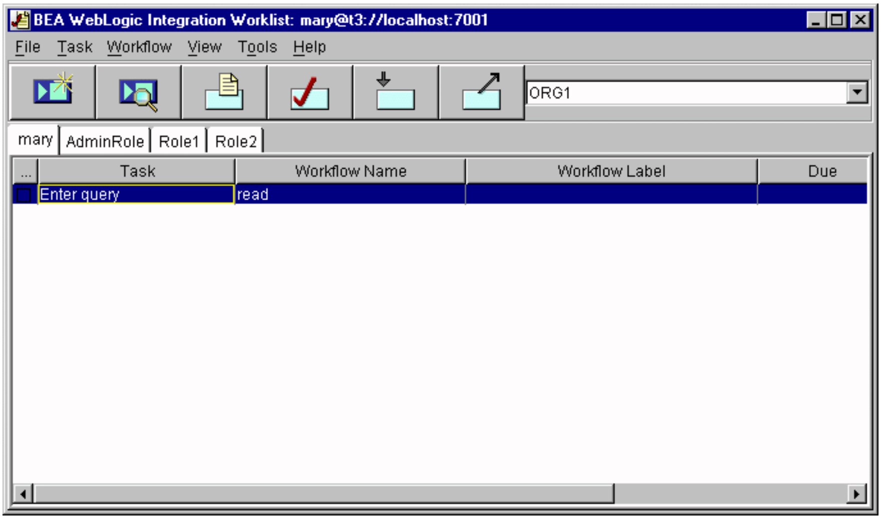
1. Start WebLogic Integration Worklist and choose **Workflow→Start a Workflow**.
2. Select `JAM read` or `JAM readString`. The template opens, as shown in Figure 4-7 and the first task is displayed.

Figure 4-7 `JAM read` or `JAM readString` Workflow



3. Select the **Enter Query** task, as shown in Figure 4-8, and click the right mouse button.

Figure 4-8 `JAM read` or `JAM readString` Worklist

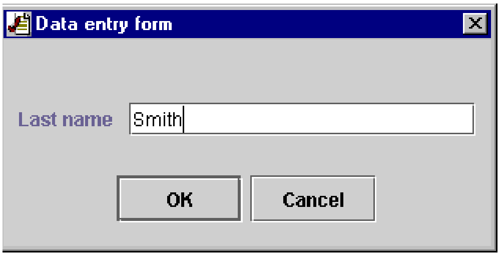


ORG1

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

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

Figure 4-9 Data entry form

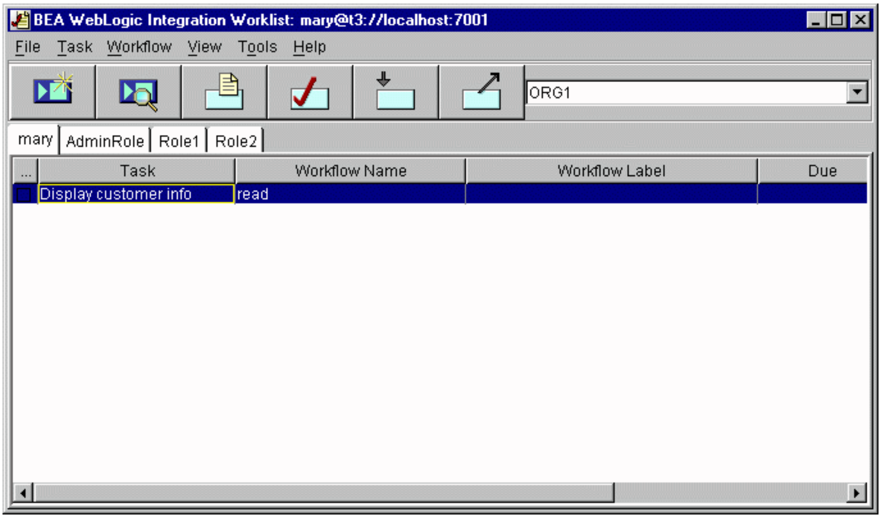


Last name: Smith

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 4-10.

Figure 4-10 Display Customer Information Task

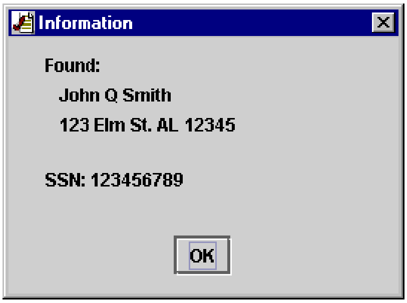


ORG1

...	Task	Workflow Name	Workflow Label	Due
	Display customer info	read		

7. Select **Execute**. When the Information dialog box appears, as shown in Figure 4-11, click **OK** to close the dialog box. The workflow is now complete.

Figure 4-11 Information Dialog window



Found:
John Q Smith
123 Elm St. AL 12345

SSN: 123456789