



# BEA WebLogic® Java Adapter for Mainframe

## Workflow Processing Guide

## Copyright

Copyright © 2002 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

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

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

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

## Trademarks or Service Marks

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

All other trademarks are the property of their respective companies.

### **BEA WebLogic Java Adapter for Mainframe Workflow Processing Guide**

<b>Part Number</b>	<b>Date</b>	<b>Software Version</b>
N/A	January 2002	5.0

---

# Contents

## 1. Workflow Processing with BEA WebLogic Java Adapter for Mainframe

Introduction to Business Process Management with WebLogic Integration ....	1-2
Understanding WebLogic Integration Interaction with the Mainframe.....	1-3
Accessing Mainframe Applications Using WebLogic JAM.....	1-5
Roadmap for Workflow Processing with WebLogic JAM .....	1-5

## 2. Installing the WebLogic JAM Plug-In

Preparing to Install WebLogic JAM Plug-in.....	2-2
Verify Software Requirements.....	2-2
Determine the Appropriate Platform.....	2-2
Download WebLogic JAM Plug-in from the Web .....	2-2
Determine the Installation Method.....	2-3
Installing the WebLogic JAM Plug-in .....	2-3
Using Graphics Mode Installation.....	2-3
Using Console Mode Installation.....	2-9
Installing Over a Previous Version .....	2-10
Installing Without WebLogic JAM.....	2-11
Installing Without WebLogic Server .....	2-11
Installing Without WebLogic Integration .....	2-11
Verifying Your Installation .....	2-12
Uninstalling BEA WebLogic Java Adapter for Mainframe Plug-in .....	2-13
Uninstalling in Graphics Mode .....	2-13
Uninstalling in Console Mode .....	2-15

## 3. Using Mainframe Applications with Workflows

Before You Begin.....	3-1
-----------------------	-----

---

Using WebLogic JAM with Workflow Processing .....	3-2
Step 1: Set Up the WebLogic JAM Plug-in Components .....	3-2
Step 2: Set Up Data Translation .....	3-4
Step 3: Set Up the Workflow to Interact With the Mainframe.....	3-8
Requesting Mainframe Services from a Workflow .....	3-9
Configuring Mainframe Services .....	3-9
Using Mainframe Actions in Workflows .....	3-10
Adding Mainframe Actions.....	3-11
Updating Mainframe Actions.....	3-15
Deleting a Mainframe Action.....	3-15
Changing the Sequence of Mainframe Actions.....	3-16
Initiating Workflows from the Mainframe .....	3-16
Example of Initiating a Workflow from the Mainframe .....	3-16
Modify the WebLogic JAM Configuration.....	3-18
Edit the Workflow Start Node.....	3-19

## 4. Running the WebLogic Integration Examples

What is Included in the Sample.....	4-1
Running the BEA WebLogic Integration/WebLogic JAM Example .....	4-2
Step 1: Set up the Mainframe .....	4-3
Step 2: Configure and Run WebLogic Integration using WebLogic Server...	4-3
Step 3: Create the Templates and Import the Template Definitions .....	4-3
Step 4: Open and Activate the Template.....	4-4
Step 5: Execute the Workflow.....	4-7
Execute the JAM create Workflow .....	4-7
Execute the JAM read and JAM readString Workflow .....	4-9

# 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](#)
  - [Understanding WebLogic Integration Interaction with the Mainframe](#)
  - [Accessing Mainframe Applications Using WebLogic JAM](#)
- [Roadmap for Workflow Processing with WebLogic JAM](#)

## 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.

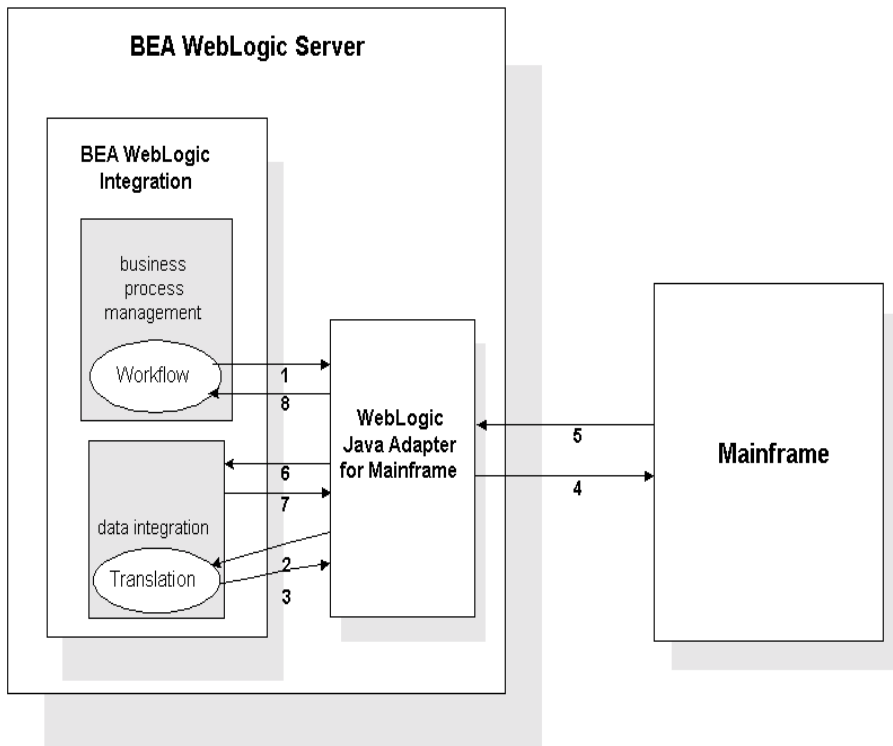
This section provides information on the following subjects:

- [Understanding WebLogic Integration Interaction with the Mainframe](#)
- [Accessing Mainframe Applications Using WebLogic JAM](#)

## Understanding WebLogic Integration Interaction with the Mainframe

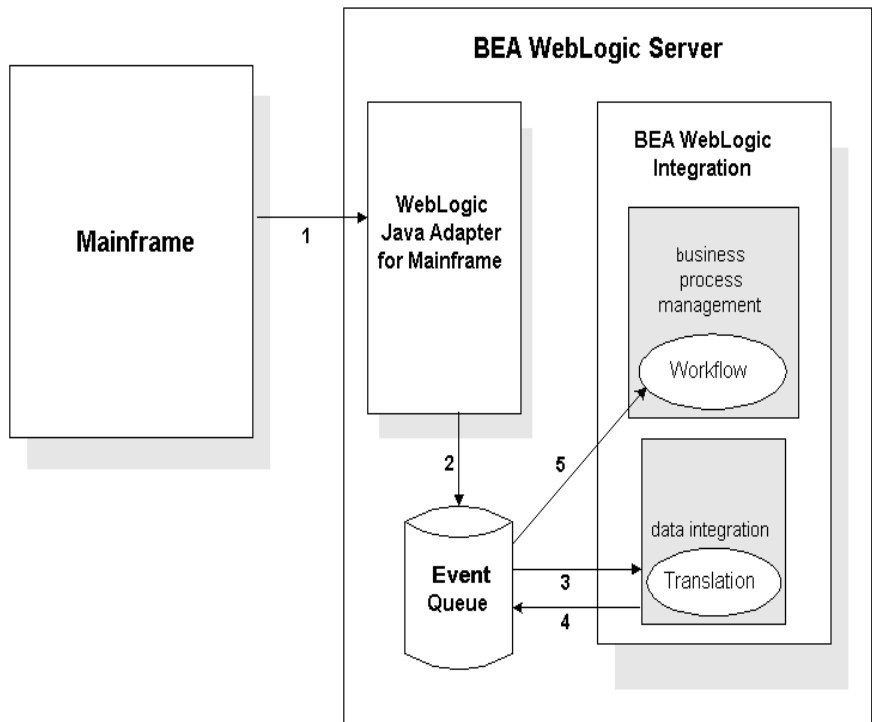
Figure 1-1 illustrates the flow of data when a WebLogic Integration workflow accesses mainframe applications through WebLogic JAM. When WebLogic Integration sends data to the mainframe application, the WebLogic JAM Gateway converts the XML data received from WebLogic Integration to the binary format 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

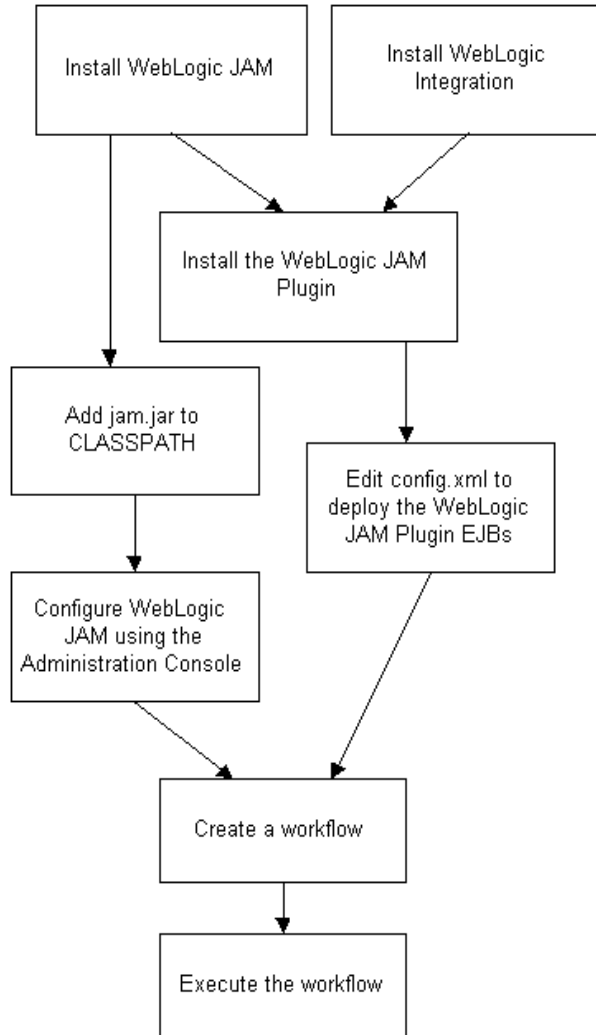
WebLogic JAM seamlessly plugs in to WebLogic Integration, allowing business analysts to include mainframe applications in business processes. WebLogic JAM uses WebLogic Integration to translate binary data to and from XML.

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.

## Roadmap for Workflow Processing with WebLogic JAM

The steps outlined in [Figure 1-3](#) provide you with a high-level guideline to all of the tasks and processes that you must perform to install, configure, and work with WebLogic JAM. Think of these steps as a roadmap to guide you through the process and to point you to the resources available to help you.

**Figure 1-3 Roadmap for Workflow Processing with WebLogic JAM**



1. Install WebLogic Integration (see *Installing BEA WebLogic Integration*) and WebLogic JAM (see the *BEA WebLogic Java Adapter for Mainframe Installation Guide*).
2. Add the `jam.jar` file to your CLASSPATH (see the *BEA WebLogic Java Adapter for Mainframe Installation Guide*).

3. Install the WebLogic JAM Plug-in (see [Installing the WebLogic JAM Plug-In](#)).
4. Edit the `config.xml` file to deploy the WebLogic JAM Plug-in EJBs (see [Using WebLogic JAM with Workflow Processing](#)).
5. Configure WebLogic JAM using the Administration Console (see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*).
6. Create a workflow to interact with a mainframe application (see [Using Mainframe Applications with Workflows](#)).
7. Execute the workflow (see [Using Mainframe Applications with Workflows](#)).



# 2 Installing the WebLogic JAM Plug-In

This section provides information on the following subjects:

- [Preparing to Install WebLogic JAM Plug-in](#)
  - [Verify Software Requirements](#)
  - [Download WebLogic JAM Plug-in from the Web](#)
  - [Determine the Installation Method](#)
- [Installing the WebLogic JAM Plug-in](#)
  - [Using Graphics Mode Installation](#)
  - [Using Console Mode Installation](#)
- [Verifying Your Installation](#)
- [Uninstalling BEA WebLogic Java Adapter for Mainframe Plug-in](#)

# Preparing to Install WebLogic JAM Plug-in

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

- [Verify Software Requirements](#)
- [Download WebLogic JAM Plug-in from the Web](#)
- [Determine the Installation Method](#)

## Verify Software Requirements

The WebLogic JAM Plug-in is dependent on the following software. Verify the installation and configuration of:

- WebLogic JAM
- WebLogic Integration

Verify that these software products have been properly configured by referring to the appropriate product documentation.

## Determine the Appropriate Platform

The WebLogic JAM Plug-in may be installed on a variety of platforms. Refer to the *BEA WebLogic Java Adapter for Mainframe Release Notes* for a complete list of supported platforms.

You must decide the platform on which platform you will install the WebLogic JAM Plug-in before proceeding.

## Download WebLogic JAM Plug-in from the Web

An evaluation copy of WebLogic JAM Plug-in is available for download from the BEA corporate Web site at:

[http://commerce.bea.com/downloads/weblogic\\_server.jsp#jam](http://commerce.bea.com/downloads/weblogic_server.jsp#jam).

Because the WebLogic JAM Plug-in shares the WebLogic JAM license, its license will be updated when you update the WebLogic JAM license.

## Determine the Installation Method

The WebLogic JAM Plug-in can be installed in the following ways:

- In Graphics Mode on Windows or Unix

Graphics mode installation is the graphics-based method of executing the BEA Installation program. It can be run on both Windows and Unix systems.

**Note:** To run Graphics mode installation, the console attached to the machine on which you are installing the software must support a Java-based GUI.

- In Console Mode on Unix

Console-mode installation is the text-based method of executing the BEA Installation program. It can be run only on Unix systems and is intended for Unix systems with non-graphic consoles.

## Installing the WebLogic JAM Plug-in

After performing the tasks described in [Preparing to Install WebLogic JAM Plug-in](#), you can begin the installation procedure. There are two ways to install the WebLogic JAM Plug-in:

- [Using Graphics Mode Installation](#)
- [Using Console Mode Installation](#)

## Using Graphics Mode Installation

To install the WebLogic JAM software in graphics mode, perform the following steps:

## 2 Installing the WebLogic JAM Plug-In

---

1. For Windows platforms, open the `jampi_win.exe` file you downloaded (see [Download WebLogic JAM Plug-in from the Web](#)).

For Unix platforms, do the following:

- a. Define the `DISPLAY` environment variable as follows:

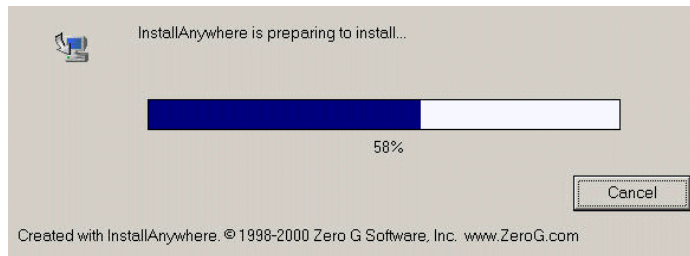
```
/mnt> export DISPLAY=xx.xx.xx.xx:0.0
```

(`xx.xx.xx.xx` is the IP address of the machine from which you are installing)

- b. Execute the installation program.

```
/mnt> sh jampi_Unix.bin
```

2. The **InstallAnywhere** screen displays, preparing to install.



3. The **BEA WebLogic Java Adapter for Mainframe** splash screen displays.

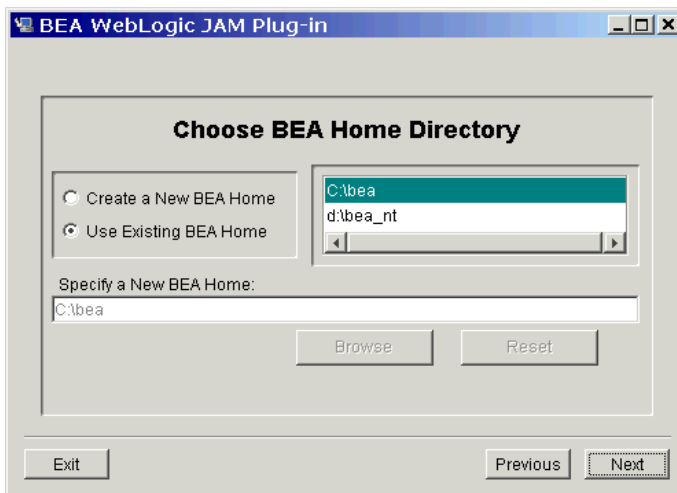




4. Then, the **Introduction** screen displays.



5. Click **Next** to continue the installation process. The **Choose BEA Home Directory** displays.

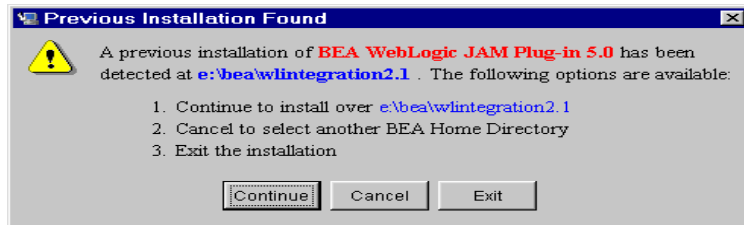


6. Select a BEA Home option, then click **Next**.
  - If you select the default setting, **Use Existing BEA Home**, the default location is the BEA Home directory where other BEA WebLogic software is located.

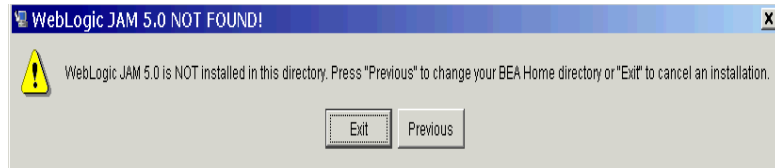
## 2 Installing the WebLogic JAM Plug-In

---

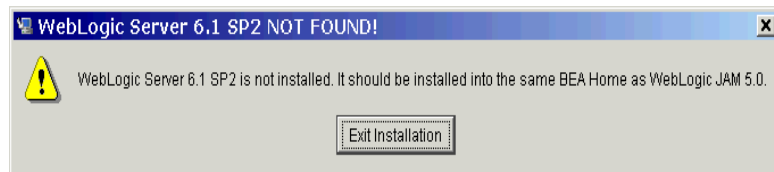
- If you select **Create New BEA Home**, you can enter the location for the new directory or click **Browse** to select a new location.
7. If a previous installation of the WebLogic JAM Plug-in is detected, the following message displays:



- Click **Continue** to install over the previous version.
  - Click **Exit** to exit the installation process.
  - Click **Previous** to return to the previous screen to change the location of the directory.
8. If WebLogic JAM is not installed, the following warning displays:

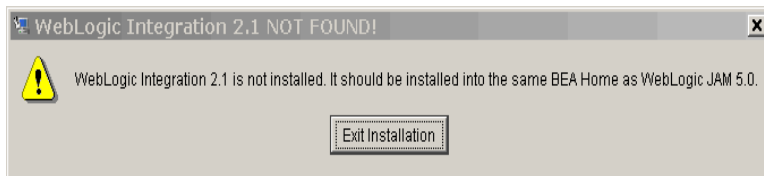


- Click **Exit** to exit the installation process.
  - Click **Previous** to return to the previous screen to change the location of the directory.
9. If WebLogic Server 6.1, Service Pack 2 is not installed, the following warning displays:



- Click **Exit** to exit the installation process. You can restart the installation process after you install WebLogic Server.

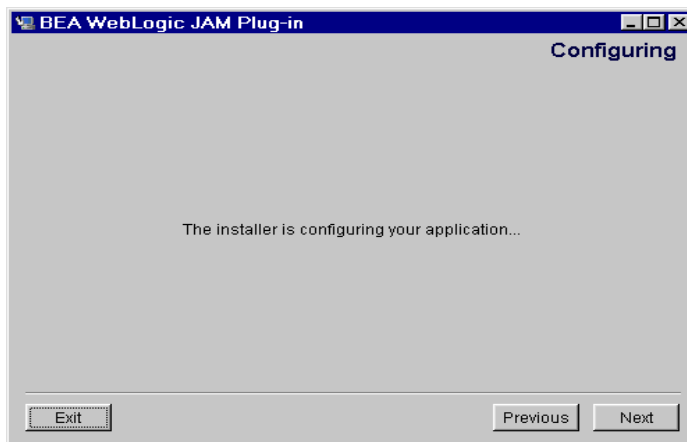
10. If WebLogic Integration is not installed, the following warning displays:



- Click **Exit** to exit the installation process. You can restart the installation process after you install WebLogic Integration.

**Note:** You can have several BEA Home directories, but products that are dependent on each other must be installed in the same BEA Home directory. For example, WebLogic Server and WebLogic JAM must be in the same BEA Home directory.

11. When you have selected the appropriate BEA Home directory, the **Configuring** screen displays.



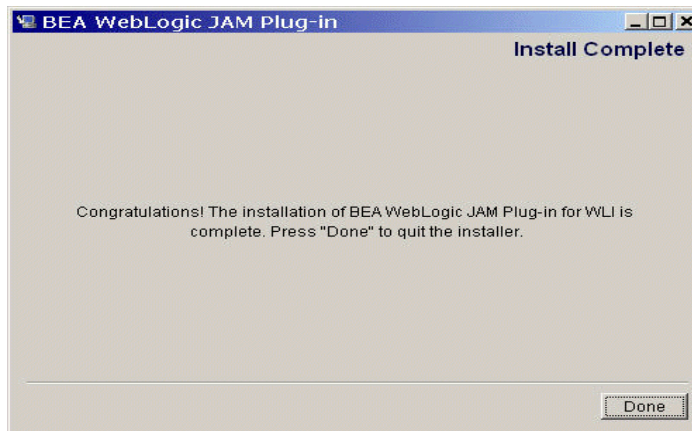
12. Click **Next** to begin the installation process. A screen showing the status of the installation process displays.

## 2 Installing the WebLogic JAM Plug-In

---



13. When the installation process is complete, the **Install Complete** screen displays.



14. Click **Done** to exit the installation program. The Windows installation process updates the Windows program group named BEA WebLogic E-Business Platform with the WebLogic JAM Plug-in program item.

## Using Console Mode Installation

To install the WebLogic JAM Plug-in on Unix systems using the console-mode installation, run the BEA standard installation shell script as follows:

```
jampi_Unix.bin
```

Perform the following steps:

1. Set the `JAVA_HOME` and `PATH` environment variables as follows:

```
/mnt> export JAVA_HOME=<BEA_HOME>/<JDK_DIR>
```

```
/mnt> export PATH=<BEA_HOME>/<JDK_DIR>/bin:$PATH
```

Example: If your BEA Home directory is `/usr/BEA`, then these environment variables should be set as follows:

```
/mnt> export JAVA_HOME=/usr/BEA/jdk131
```

```
/mnt> export PATH=/usr/BEA/jdk131/bin:$PATH
```

2. Execute the installation program you downloaded (see [Download WebLogic JAM Plug-in from the Web](#)). For example:

```
mnt> sh jampi_Unix.bin -i console
```

3. The installation program runs and prompts you for responses. [Listing 2-1](#) shows a sample of the console mode installation script. User input is shown in bold.

### Listing 2-1 Console Mode Installation Script

---

```
Preparing to install...
Preparing CONSOLE Mode Installation...
=====
BEA WebLogic JAM Plug-in (created with InstallAnywhere by Zero G)
-----
=====

Introduction
-----

This installer will guide you through the installation of 'BEA
WebLogic JAM Plug-in'. On each step, answer the questions provided.
Type "quit" at any time to exit the installer.
```

## 2 Installing the WebLogic JAM Plug-In

---

```
PRESS <ENTER> TO CONTINUE:

=====

Choose BEA Home Directory
-----

    1- Create a New BEA Home
    2- Use Existing BEA Home

Enter a number: 2

    1- /bea
    2- /tmp/install.dir.15582

Existing BEA Home: 1

Installing...
-----

[-----|-----|-----|-----]
=====

Install Complete
-----

Congratulations. 'BEA WebLogic JAM Plug-in' has been successfully
installed to:

    /bea/wlintegration2.1

PRESS <ENTER> TO EXIT THE INSTALLER:
```

---

### Installing Over a Previous Version

If you are installing over a previous version of the WebLogic JAM Plug-in, you will see the following section of the installation script:

```
>>>> THIS PART ONLY APPEARS WITH A PREVIOUS INSTALL <<<<
-----

A previous installation of BEA WebLogic JAM Plug-in 5.0 has been
detected at /bea/wlintegration2.1.

The following options are available:

    1- Continue to install over /bea/wlintegration2.1
```

- 2- Cancel to select another BEA Home Directory
- 3- Exit the installation

Choose a number: 1

```
>>>> THIS PART ONLY APPEARS WITH A PREVIOUS INSTALL <<<
=====
```

## Installing Without WebLogic JAM

If you attempt to install the WebLogic JAM Plug-in without a valid installation of WebLogic JAM, you will see the following section of the installation script:

```
=====
WebLogic JAM Plug-in requires a valid installation of the WebLogic
JAM 5.0. Check your BEA Home directory to ensure that product is
installed. Hit <ENTER> to quit an installation.:

[machine_name]:/home/tmp
.....
```

## Installing Without WebLogic Server

If you attempt to install the WebLogic JAM Plug-in without a valid installation of WebLogic Server 6.1, SP2, you will see the following section of the installation script:

```
=====
WebLogic Server 6.1 SP2 NOT FOUND!
-----

WebLogic Server 6.1 SP2 is not installed. It should be installed
into the same BEA Home as WebLogic JAM 5.0. Hit <ENTER> to quit an
installation.:

[machine_name]:/home/tmp
.....
```

## Installing Without WebLogic Integration

If you attempt to install the WebLogic JAM Plug-in without a valid installation of WebLogic Integration 2.1, you will see the following section of the installation script:

```
=====
WebLogic Integration 2.1 NOT FOUND!
-----

WebLogic Integration 2.1 is not installed. It should be installed
into the same BEA Home as WebLogic JAM 5.0. Hit <ENTER> to quit an
installation.:

[machine_name] : /home/tmp

[===== | ===== | ===== | =====]
```

## Verifying Your Installation

You can verify the installation of the WebLogic JAM Plug-in by verifying the installation of the following directories and files for the supported platforms.

The following platforms are supported for the WebLogic JAM Plug-in.

- AIX 4.3.3
- HP-UX 11.0
- Microsoft Windows NT 4.0, Windows 2000, or Windows 2000 Advanced Server
- Solaris 7
- Solaris 8
- SuSE LINUX for OS/390

Directory	Files
<BEA_HOME>/	jampi.jar
wlintegration2.1/	jampi.war
lib	



# Uninstalling BEA WebLogic Java Adapter for Mainframe Plug-in

The WebLogic JAM Plug-in may be removed or uninstalled in the following ways:

- For Windows:

- Graphical mode

- For Unix:

- Either graphical or console mode.

**Note:** You must use the same method to uninstall the product that you used to install it

## Uninstalling in Graphics Mode

To uninstall WebLogic JAM Plug-in in graphics mode, perform the following steps:

1. On Windows platforms, choose Start—>Programs—>BEA WebLogic E-Business Platform—>BEA WebLogic JAM Plug-in 5.0—>Uninstall WebLogic JAM Plug-in.

On Unix platforms, do the following:

- a. Change directories to your `uninstaller` directory in the WebLogic Integration root directory with the following command. The default is `<BEA_HOME>/jampi/uninstaller`.

```
>cd <BEA_HOME>/wlintegration2.1/uninstaller_JAMPI
```

- b. Start the uninstallation program with the following command:

```
/usr/BEA/wlintegration2.1/uninstaller_JAMPI> sh Uninstall
```

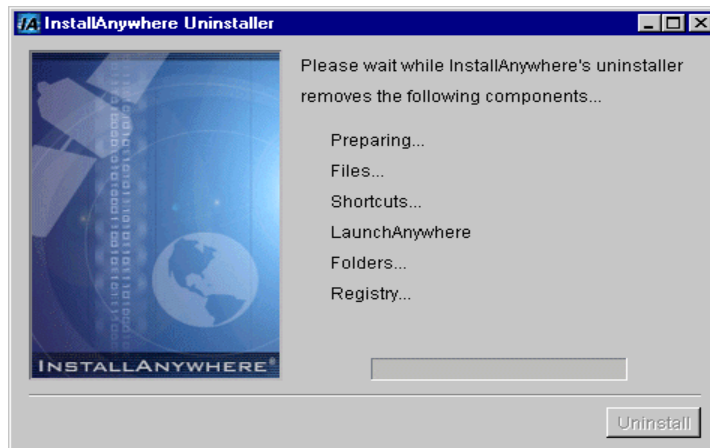
The following screen displays.

## 2 Installing the WebLogic JAM Plug-In

---



2. Click **Uninstall** to begin the uninstall process. The following screen displays.



- When the uninstall process is complete, the following screen displays.



- Click **Exit** to close the uninstallation program

## Uninstalling in Console Mode

To uninstall WebLogic JAM Plug-in in console mode, perform the following steps:

- Change directories to the `uninstaller` directory under the root directory with the following command. The default is `/usr/BEA/jampi/uninstaller`.

```
> cd <BEA_HOME>/wlintegration2.1/uninstaller_JAMPI
```

- Start the uninstallation program with the following command:

```
<BEA_HOME>/wlintegration2.1/uninstaller_JAMPI> sh Uninstall -i  
console
```

- At the prompt, press **ENTER** to begin the uninstall.  
If you want to exit the uninstall program, type **QUIT** at the prompt.

**Note:** The single-dash bar just beneath the double-dash bar in [Listing 2-1](#) is a progress indicator. When the single-dash bar becomes the same length as the double-dash bar, uninstallation is complete.

- When the uninstall is complete, press **ENTER** to exit the program.



# 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 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 Up Data Translation](#)
- [Step 3: Set Up the Workflow to Interact With the Mainframe](#)

## 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>/<WLI_HOME>/config/<domain.name>
```

Where `<WLI_HOME>` 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. The WebLogic JAM Plug-in components should have the highest `DeploymentOrder` values so that they will be deployed last.

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

### Listing 3-1 Example 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-mdb-ejb.jar" DeploymentOrder="3"/>
<EJBComponent Name="pobean.jar" Targets="myserver"
URI="pobean.jar" DeploymentOrder="4"/>
<EJBComponent Name="jampi.jar" Targets="myserver"
URI="jampi.jar" DeploymentOrder="5"/>
<WebAppComponent Name="com.bea.jam.JamPlugin"
Targets="myserver" URI="jampi.war" DeploymentOrder="6"/>
</Application>
```

---

## Step 2: Set 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 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 Guide*.
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](#)).

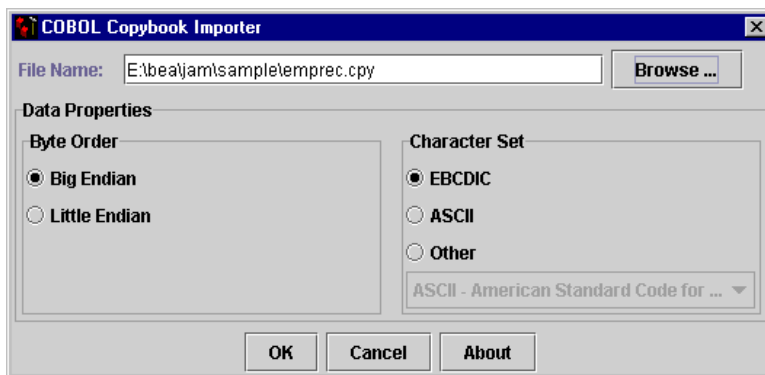


Figure 3-1 WebLogic Integration Repository Login

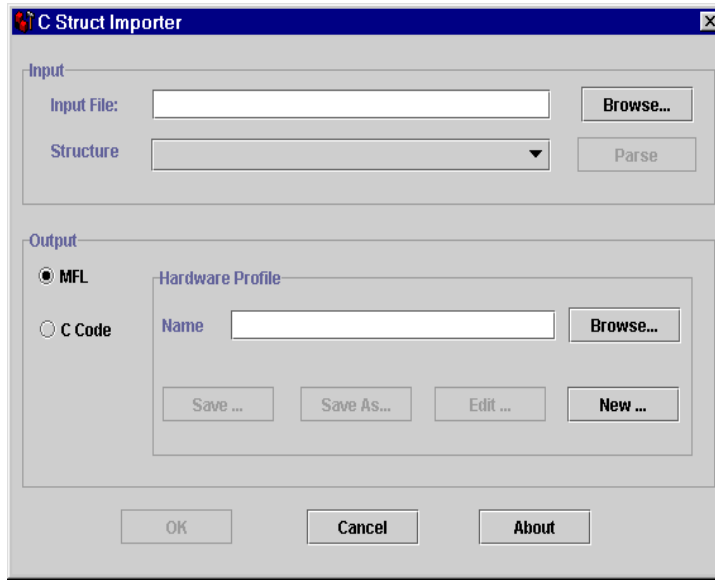


- c. Enter your user name and password. If your server is different than the default, enter your server address.
- d. Choose Tools→Import→COBOL Copybook Importer (if you are working with a copybook) or C Struct Importer (if you are working with a C Struct). The appropriate dialog box displays (Figure 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

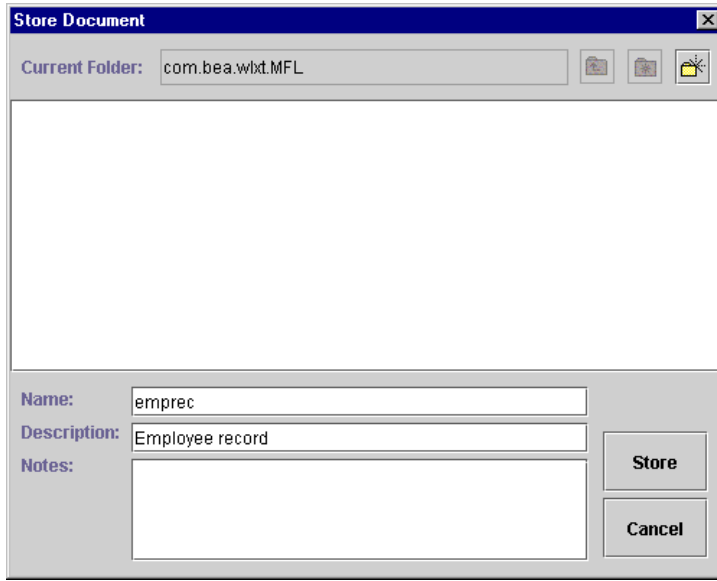


**Figure 3-3 WebLogic Integration C Struct Importer Dialog Box.**



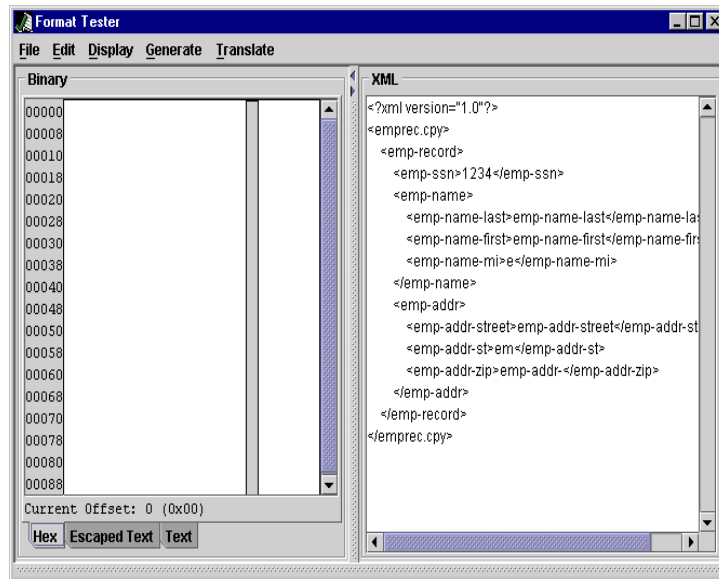
- e. Fill in the fields in the dialog box as appropriate for your environment.
- f. After your file has been imported, choose Repository→Store As. The Store Document dialog box displays (see [Figure 3-4](#)).

Figure 3-4 WebLogic Integration Store Document Dialog Box



- 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**



- 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 to Interact With the Mainframe

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

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

# Requesting Mainframe Services from a Workflow

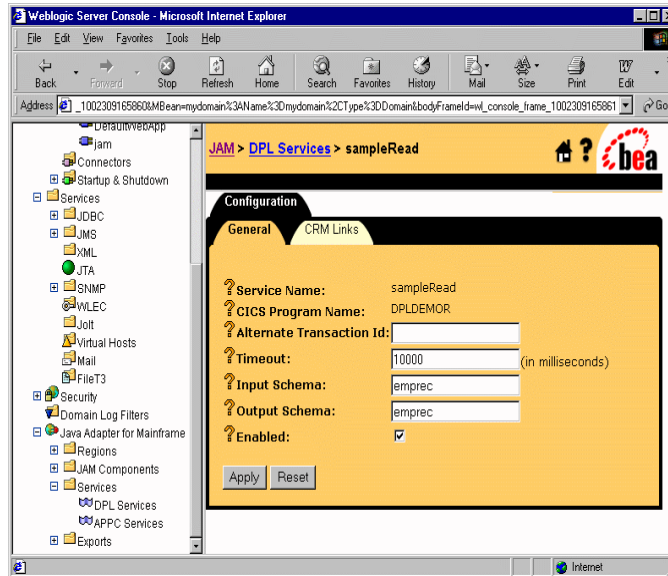
To request mainframe services from a workflow in WebLogic Integration, you must:

- Configure mainframe services using the Administration Console (see [Configuring Mainframe Services](#)).
- Add appropriate mainframe actions to tasks contained in the WebLogic Integration workflow template definition (see [Using Mainframe Actions in Workflows](#)).

## Configuring Mainframe Services

To access mainframe applications from a WebLogic Integration workflow, remote services must be specified. Remote services are configured using the WebLogic Administration Console.

Figure 3-6 Configuring Remote Services



The **Input Schema** and **Output Schema** fields should specify data translation schemas used by the service in the following way:

- **Input Schema** specifies the translation schema for the request sent to the mainframe.
- **Output Schema** specifies the translation schema for the response received from the mainframe.

For specific information about configuring the WebLogic JAM Gateway, refer to the *BEA 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](#)

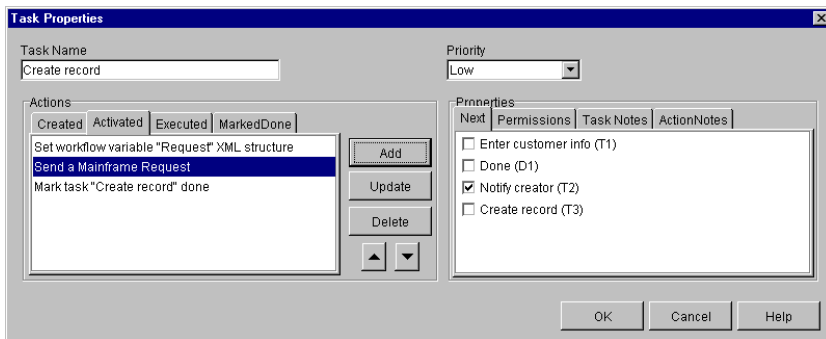
- [Deleting a Mainframe Action](#)
- [Changing the Sequence of Mainframe Actions](#)

## 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 3-7](#).

**Figure 3-7 Task Properties**



3. In the Task Properties dialog box shown in [Figure 3-7](#), 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 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.

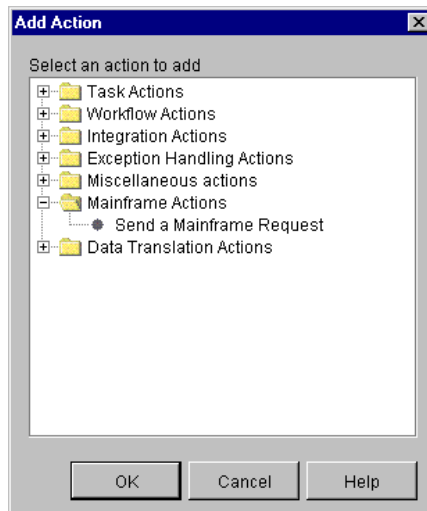
**Table 3-1 Business Process Management Action Definitions**

Tab	Description
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 <b>Execute Task</b> action is performed for that task.
MarkedDone	The task has been completed. This event usually occurs in response to the execution of a <b>Mark Task as Done</b> 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 3-8](#).

**Figure 3-8 Adding Mainframe Actions**

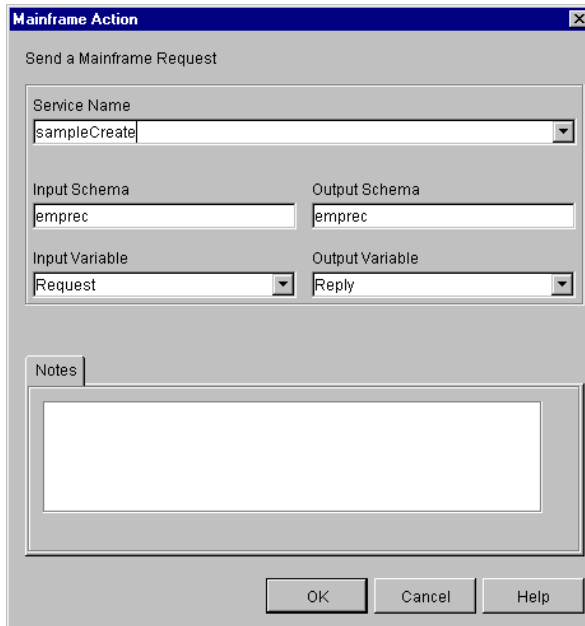


5. Double-click the Mainframe Actions folder to expand it. Select **Send a mainframe request**, and click **OK** to display the Mainframe Action dialog box.



- Enter data in the fields as shown in [Figure 3-9](#). A description of the input fields for this dialog box is shown in [Table 3-2](#).

**Figure 3-9 Mainframe Actions Dialog Box**



[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
<b>Mainframe Action Definition</b>	
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 configuration. If you enter the name of a new service, you must be sure to add the service to the WebLogic JAM configuration before you can run the workflow.

<b>Field</b>	<b>Description</b>
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 want 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 want to create it.
Notes	Enter your own custom notes about the mainframe action.
<b>Mainframe Action Buttons</b>	
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 3-7](#), select the **Actions** tab that contains the mainframe action to be 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. In the Task Properties dialog box shown in [Figure 3-7](#), select the **Actions** tab that contains the mainframe action to be executed. The Actions are defined in [Table 3-1](#).
4. Select the desired action in the **Actions** section of the Task Properties dialog box, and click the **Delete** button.
5. 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. In the Task Properties dialog box shown in [Figure 3-7](#), select the **Actions** tab that contains the mainframe action to be executed. The Actions are defined in [Table 3-1](#).
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 to 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 3-2](#) 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 3-2 Example of Initiating a Workflow From the Mainframe

---

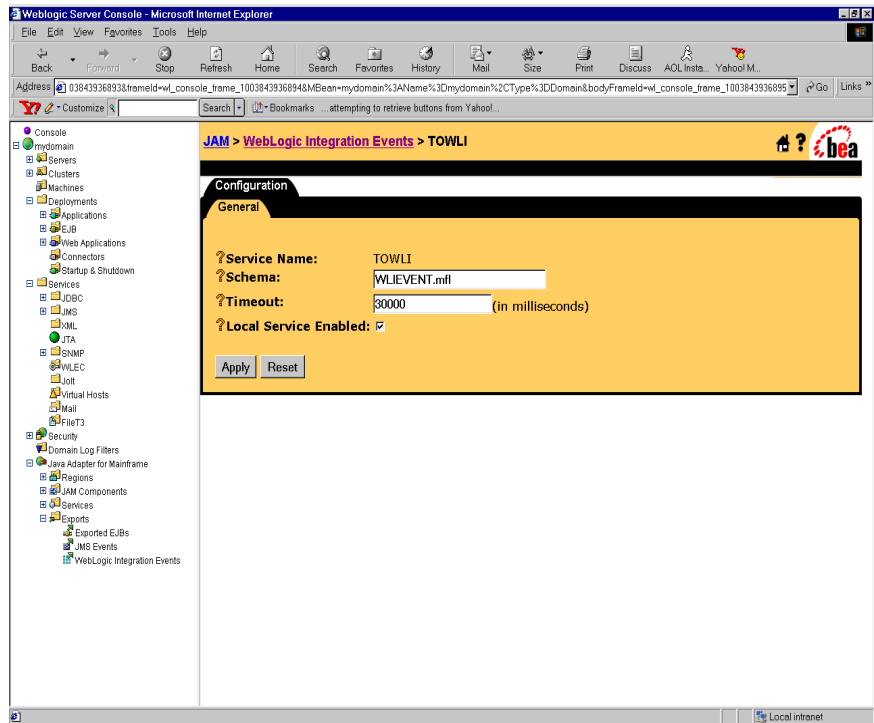
```
DATA DIVISION.
WORKING-STORAGE SECTION.
01 WS-VARIABLES.
   COPY EMPREC.
01 WS-CONSTANTS.
   05 EMP-REQUEST                PIC X(8)
                                   VALUE 'DPL1SVR'.
   05 REMOTE-SYSID                PIC X(4)
                                   VALUE 'BEAL'.
.
.
.
PROCEDURE DIVISION.
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 configuration file must have a WebLogic Integration event for each service to be requested by the mainframe program. This definition must specify a schema that was created when you set up data translation.

**Figure 3-10** Event Console



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 3-11](#), select **Event** as the start method for the workflow.

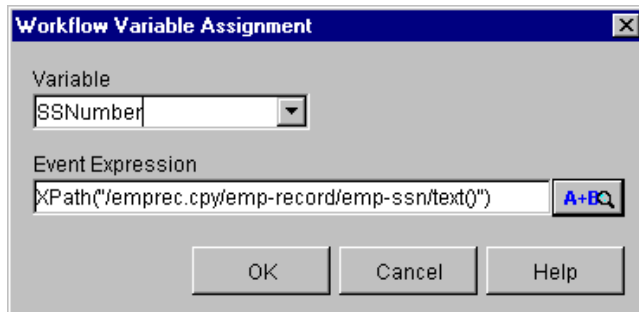
**Figure 3-11 Start Node**

The screenshot shows the 'Start Properties' dialog box. The 'Description' field contains 'Start'. The 'Event' radio button is selected. The 'Document Type / Root Element' field contains 'emprec.cpy'. The 'Start Organization' dropdown is set to 'ORG1'. The 'Variables' tab is active, showing an empty table with columns 'Variable' and 'Expression', and buttons for 'Add', 'Update', and 'Delete'. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

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 3-12](#), 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 3-12 Workflow Variable Assignment Dialog Box**



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



# 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 example workflow templates initiate mainframe requests to read and write from a mainframe database. The examples are located in the following directory:

```
<BEA_HOME>/<JAM_INSTALL_DIR>/samples/examples/wlintegration
```

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

Directory	File	Description
wlintegration	create.xml	Workflow that creates an employee record.
wlintegration	read.xml	Workflow that reads an employee record.
wlintegration	readString.xml	Workflow that reads an employee record using string data type.
wlintegration	emprec.cpy	Copybook for the employee record.

These examples use mainframe components from the Employee Record CICS outbound sample. Mainframe components for the Employee Record sample are located in the following directory:

```
<BEA_HOME>/>JAM_INSTALL_DIR>/samples/examples/CICS/outbound/  
mainframe/source
```

A sample WebLogic JAM configuration file can be found in the following directory:

```
<BEA_HOME>/>JAM_INSTALL_DIR>/config/examples/jamconfig_CICS.xml
```

For information on setting up the Employee Record sample, refer to the *BEA WebLogic Java Adapter for Mainframe Samples Guide*.

# 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.

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 `dpldemor.cbl`, `dpldemou.cbl`, and `dpldemod.cbl`. Refer to the BEA WebLogic Java Adapter for Mainframe *Samples Guide* for more information.

## Step 2: Configure and Run WebLogic Integration using WebLogic Server

1. Move the `jamconfig_CICS.xml` file from the `config/examples` directory to: `<BEA_HOME>/<WLI_HOME>/config/<domain.name>`  
where `<domain.name>` is your WebLogic Integration domain.
2. Start the CRM (refer to the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*).
3. Start the WebLogic Integration Server.
4. Using the WebLogic Administration Console, update the WebLogic JAM configuration to communicate with the CRM (refer to the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*).
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**

<b>Workflow Template</b>	<b>File Name</b>	<b>Description</b>
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

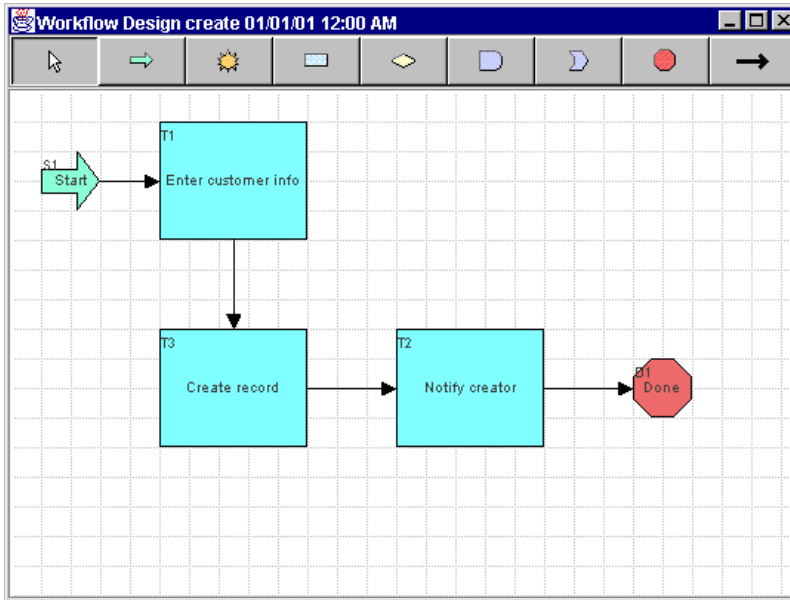
## Step 4: Open and Activate the Template

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 4-1](#).

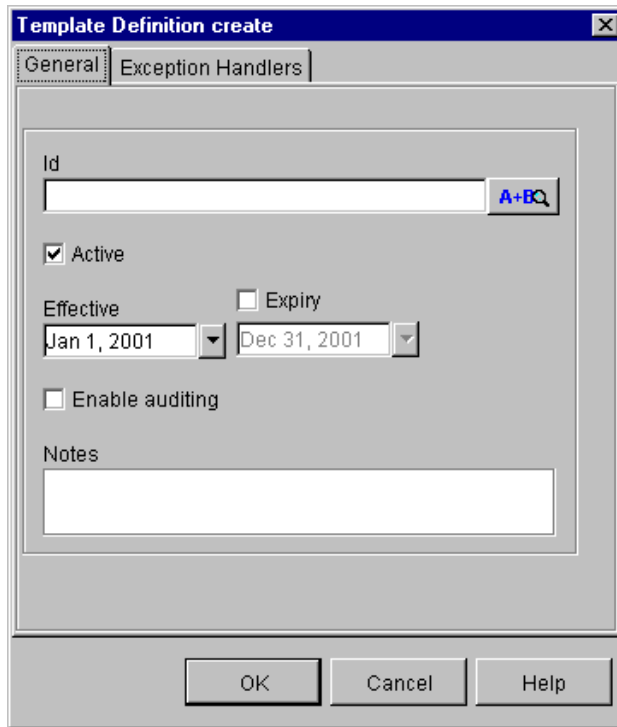
**Figure 4-1 Workflow for JAM create**



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



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

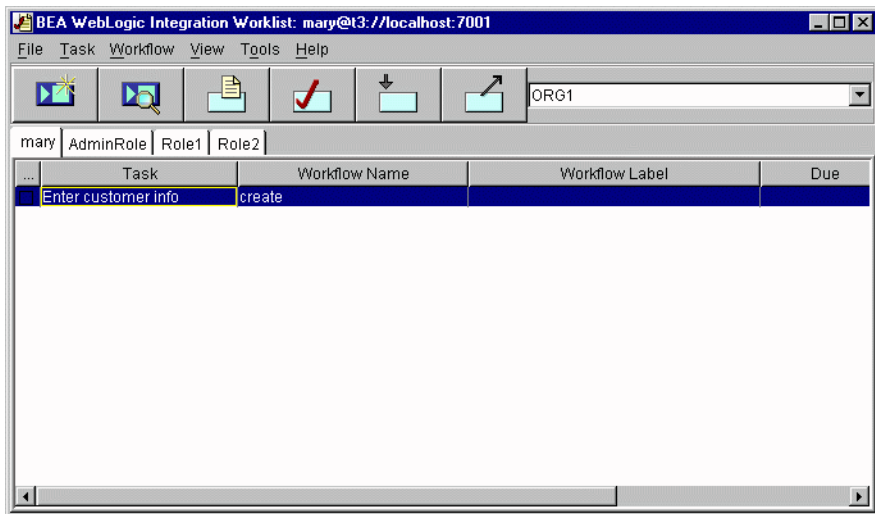
This section provides step-by-step instructions for executing workflows to create an employee record and to read an employee record.

### 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**



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**



The image shows a standard Windows-style dialog box titled "Data entry form". It contains seven text input fields, each with a label to its left. The labels are "Last name", "First name", "Middle", "Street", "State", "Zip", and "SSN". The corresponding values entered in the fields are "Smith", "John", "Q", "123 Elm St.", "AL", "12345", and "123456789". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

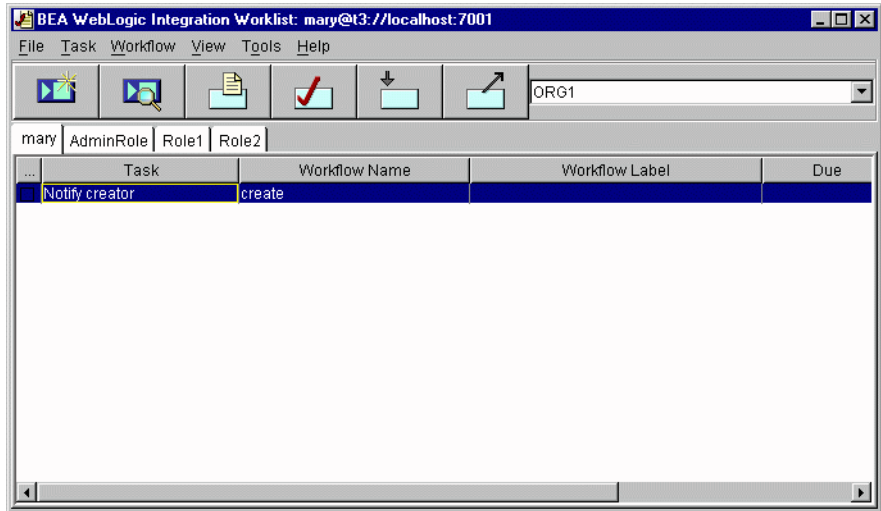
Field Label	Value
Last name	Smith
First name	John
Middle	Q
Street	123 Elm St.
State	AL
Zip	12345
SSN	123456789

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



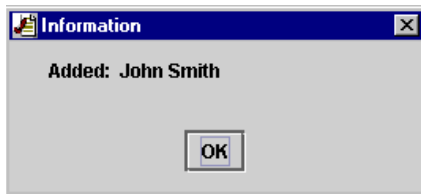
- A new task, **Notify Creator** will appear on the worklist, as shown in [Figure 4-5](#). Select the **Notify Creator** task.

**Figure 4-5** Notify Creator Task



- 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



## 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.

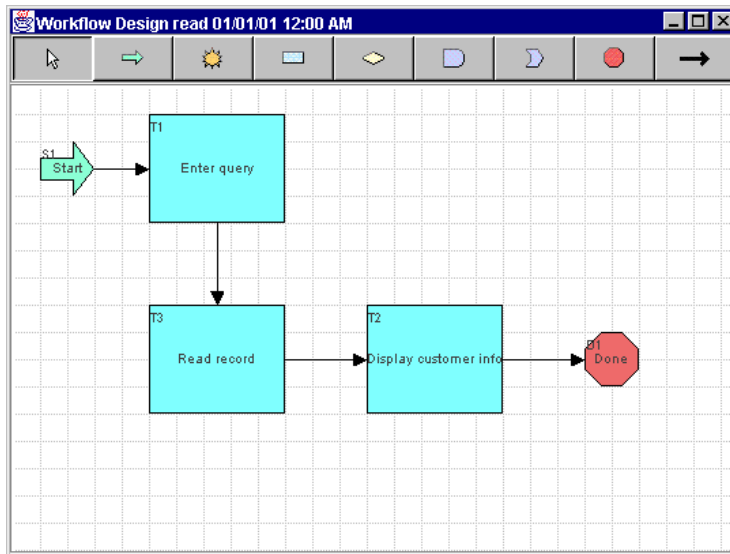
## 4 Running the WebLogic Integration Examples

---

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

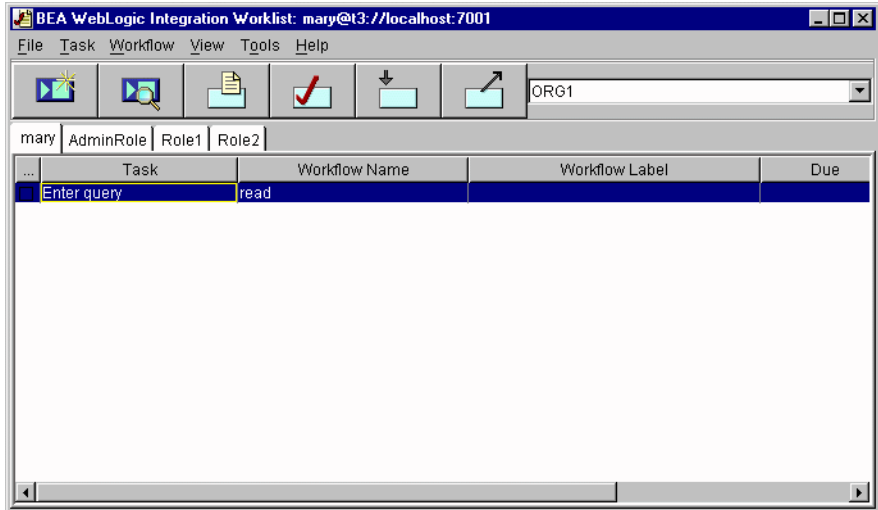
1. Start WebLogic Integration Worklist and choose **Workflow**→**Start a Workflow**.
2. Select `JAM read` or `JAM readString`. The template opens, as shown in [Figure 4-7](#) and the first task is displayed.

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



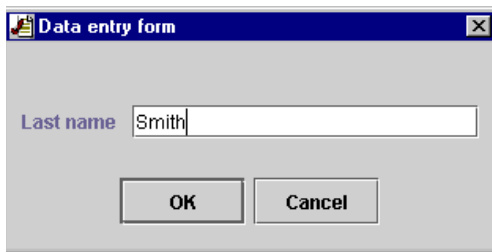
3. Select the **Enter Query** task, as shown in [Figure 4-8](#), and click the right mouse button.

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



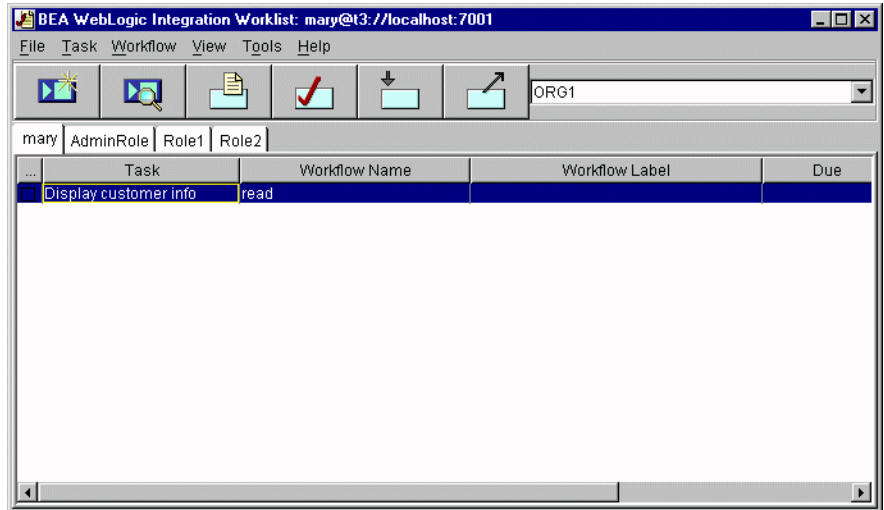
4. Select **Execute**. The **Data entry form** dialog box displays, as shown in [Figure 4-9](#).

**Figure 4-9** Data entry form



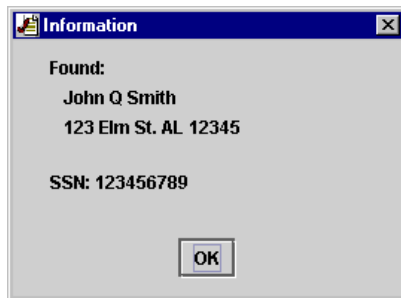
5. Enter the last name of the employee to be read. For example, the last name entered when you created an employee record was Smith. Click **OK**.
6. A new task, **Display Customer Information**, will appear on the worklist. Select the **Display Customer Information** task, as shown in [Figure 4-10](#).

**Figure 4-10 Display Customer Information Task**



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**



---

# Index

- B**  
BEA home directory 2-5
- C**  
config.xml 3-2  
    location 3-2  
console mode 2-9, 2-15
- D**  
data translation  
    setup 3-4
- G**  
Gateway 1-3, 3-18  
graphics mode 2-3, 2-13
- I**  
input schema 3-10  
installation verification 2-12
- M**  
mainframe actions 3-10  
    adding 3-11  
    changing sequence 3-16  
    deleting 3-15  
    updating 3-15  
mainframe applications 1-5  
    mainframe services  
        requesting mainframe services 3-9
- O**  
output schema 3-10
- P**  
platforms 2-2, 2-12
- R**  
Roadmap 1-6
- S**  
sample files 4-2  
set up 3-4  
software requirements 2-2
- W**  
web downloading 2-2  
WebLogic Integration 1-1, 1-2, 2-2, 2-7, 2-11, 2-13, 3-1, 3-18, 4-1, 4-2, 4-3, 4-10  
    interaction with mainframe 1-3  
WebLogic Server 2-6, 2-11, 3-1, 3-2, 4-3  
    configuration file 3-2  
        location 3-2  
workflow templates 4-4

---

workflows

edit start node 3-19

initiating from the mainframe 3-16

## **X**

XML 1-1, 1-3, 1-5, 3-19

data translation 3-4