bea®

BEA WebLogic Java Adapter for Mainframe™

Installation Guide

Copyright

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

Restricted Rights Legend

Trademarks or Service Marks

BEA WebLogic Java Adapter for Mainframe Installation Guide

Part Number	Date	Software Version
870-001038-006	August 2002	5.1

Contents

6. Uninstalling BEA WebLogic Java Adapter for Mainframe

A. BEA WebLogic Java Adapter for Mainframe Installation Files and Directories

Index

1 Introduction to Installing WebLogic JAM

Proper installation is essential to successfully operating BEA WebLogic Java Adapter for Mainframe (WebLogic JAM). The following sections will provide an overview of the WebLogic JAM installation process.

About BEA WebLogic Product Distribution

BEA WebLogic Java Adapter for Mainframe (WebLogic JAM) is distributed on both the Web and CD-ROM. Service Packs for WebLogic JAM are distributed only on the Web.

Web Distribution of WebLogic JAM

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 evaluation license. At the end of the evaluation period, you can purchase a non-expiring license for your WebLogic JAM product.

CD-ROM Distribution of WebLogic JAM

If you purchased WebLogic JAM from your local sales representative, you will find the following items in the product box:

- Two CD-ROMs:
  - BEA WebLogic Java Adapter for Mainframe product CD-ROM
  - BEA WebLogic Java Adapter for Mainframe Online Documentation CD-ROM
- The following printed documents:
  - *WebLogic Java Adapter for Mainframe Installation Guide* (this document)
  - *WebLogic Java Adapter for Mainframe Release Notes*
  - "BEA Software License and Limited Warranty" pamphlet
  - "Customer Support Quick Reference and Other Important Information" card

You can also access the WebLogic JAM Online Documentation at:

http://edocs.bea.com/index.html

About the WebLogic JAM Components

To meet the demands of today's business environment, WebLogic JAM integrates back-end systems with Web-based e-business applications. Business transactions supported by these applications often require direct integration with existing mainframe legacy applications. WebLogic JAM meets these integration requirements by providing bi-directional, online request-response integration between Java applications and mainframe applications.

WebLogic JAM enables applications running on WebLogic Server and mainframe systems to interact using two main components:

■ WebLogic JAM Gateway

The WebLogic JAM Gateway component is a server that runs within WebLogic Server and communicates with the CRM using the TCP/IP protocol. The WebLogic JAM Gateway acts as a gateway to route requests and responses between WebLogic Server (Java) and mainframe systems, such as CICS and IMS. The Gateway component also forwards configuration information to the CRM at startup.

A WebLogic JAM Gateway used in conjunction with the CRM enables applications running on an instance of WebLogic Server to connect to back-end applications. Each WebLogic Server instance that needs to access back-end applications should have a WebLogic JAM Gateway defined for it. This WebLogic JAM Gateway routes requests received from Java client applications running on this instance of WebLogic Server to the CRM, which then will forward them to the back-end application. The Gateway also routes requests made by mainframe client applications that have come through the CRM to the necessary WebLogic Server applications.

■ Communications Resource Manager (CRM)

The CRM is the WebLogic JAM component that manages communications resources. The CRM coordinates the flow of data between Java applications running on a WebLogic Server platform and applications running on a mainframe.

The CRM uses SNA and TCP/IP communication protocols to communicate with the mainframe and WebLogic JAM Gateway respectively. TCP/IP protocols always flow between the WebLogic JAM Gateway and the CRM. SNA protocols always flow from the CRM to the mainframe and from VTAM on the mainframe to the CRM depending upon the configuration option.

About the BEA WebLogic Installation Program

The InstallAnywhere installation program is the BEA standard tool for WebLogic product installation. InstallAnywhere installs the WebLogic JAM Gateway on the target Windows or Unix system.

Installation Methods

InstallAnywhere supports the following installation methods:

- Graphical user interface (GUI) mode installation

  See the "Installing the WebLogic JAM Gateway Using Graphical-Mode Installation" section.

- Console-mode installation (Unix systems only)

  See the "Installing the WebLogic JAM Gateway on Unix Systems Using Console-Mode Installation" section.

About the BEA Home Directory

When installing WebLogic JAM, the installer program prompts you to specify a BEA Home directory. The BEA Home directory is a repository for common files that are used by multiple BEA products installed on the same machine. For this reason, the BEA Home directory can be considered a "central support directory" for the BEA products installed on your system.

The files in the BEA Home directory are essential to ensuring that the BEA software operates correctly on your system. They perform the following types of functions:

- Ensure that licensing works correctly for the installed BEA products

- Facilitate checking of cross-product dependencies during installation

- Facilitate Service Pack installation

The structure of a sample BEA Home directory created by the WebLogic installer program is shown in the following illustration.

Figure 1-1   BEA Home Directory
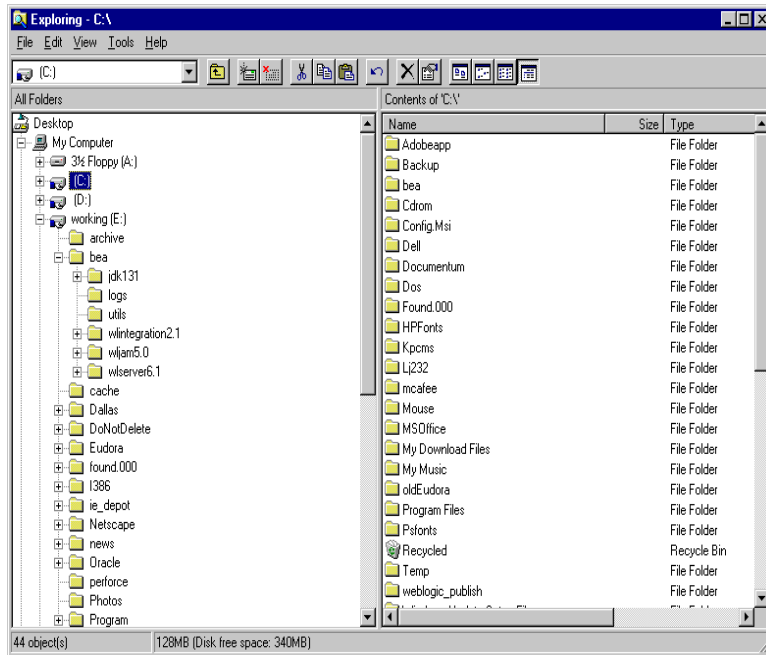
About the BEA WebLogic License

WebLogic JAM uses an XML-format license called `license.bea`. This license file, stored in the BEA Home directory, is used for the BEA WebLogic products installed on the target system. For details about the BEA Home directory, see the "About the BEA Home Directory" section.

The first time you download and install a BEA WebLogic product that uses the BEA Home directory convention, the installer program installs a `license.bea` file in the BEA Home directory that you created during installation and populates the file with an evaluation product license. Subsequently, when you download and install additional BEA products that contain a `license.bea` file as part of the distribution, the installer program automatically adds the new evaluation product licenses to the `license.bea` file.

Evaluation Licenses

An evaluation copy of WebLogic JAM comes with a 30-day evaluation license so that you can start using WebLogic JAM immediately. BEA also provides an extended 60-day evaluation license that can be downloaded from BEA web site. To use WebLogic JAM beyond the evaluation period, contact your salesperson about purchasing a non-expiring license.

Non-Expiring Licenses

If you purchase WebLogic JAM, you will receive a development or production product license through e-mail. After installing the WebLogic JAM software and receiving the product license, you must update the existing license.bea file with the license file you received.

Roadmap for Installing WebLogic JAM

Proper installation of the WebLogic JAM software involves the following tasks:

- Prepare to Install WebLogic JAM.

  See the "Preparing to Install BEA WebLogic Java Adapter for Mainframe" section of this guide.

  - Determine the Installation Method.

  - Determine Your Hardware Requirements.

  - Verify WebLogic JAM Software Requirements.

  - Choose a BEA Home Directory.

- Install WebLogic JAM.

  See the "Installing WebLogic Java Adapter for Mainframe" section of this guide.

  - Install the WebLogic JAM Gateway.

  - Install the CRM.

■ Verify Your Installation.

See the "Verifying Your Installation" section of this guide.

- Verify Installation Files and Directories.

- Run the Installation Verification Sample.

■ Perform Post-installation Task.

See the "Performing Post-Installation Tasks" section of this guide.

- Update Your license.bea File.

- Deploy WebLogic JAM in the WebLogic Server Environment.

2 Preparing to Install BEA WebLogic Java Adapter for Mainframe

You must complete several tasks before you install the BEA WebLogic Java Adapter for Mainframe (WebLogic JAM). Information about these tasks is included in the following sections:

- Determining the Installation Method

- Determining Your Hardware Requirements

- Verifying WebLogic JAM Software Requirements

- Choosing a BEA Home Directory

If you are migrating from a previous version of WebLogic JAM, see the Migration section at http://edocs/jam/v51/index.html.

Determining the Installation Method

The components of WebLogic JAM may be installed on a variety of platforms. See the *BEA WebLogic Java Adapter for Mainframe Release Notes* for a complete list of supported platforms.

The WebLogic JAM Gateway can be installed in the following ways:

- On Microsoft Windows platforms, a graphical, InstallAnywhere interface is the only choice.

- On Unix platforms, the installation is accomplished with InstallAnywhere, using either a graphical interface or a console (text mode) interface.

The Communications Resource Manager (CRM) can be installed in the following ways:

- On z/OS or OS/390 Unix, installation occurs using a Unix script.

- On z/OS or OS/390 MVS, installation involves Job Control Language (JCL) files.

After the installation is complete, the installation may be validated using the supplied installation verification samples. Verification samples are provided for CICS or IMS.

After WebLogic JAM is installed, it may be removed or uninstalled. Similar to the install interface, the uninstall interface on Microsoft Windows platforms is solely graphical. The uninstall interface on Unix platforms may be either console or graphical. No WebLogic JAM uninstall function is provided for the CRM.

Determining Your Hardware Requirements

Before you begin installing the WebLogic JAM software, verify that your system meets the size requirements for installing and running the WebLogic JAM components. The following requirements must be met for the WebLogic JAM Gateway and the CRM to be successfully installed:

- Platform Requirements

  A certified WebLogic Server platform is required. See the *BEA WebLogic Java Adapter for Mainframe Release Notes* for a complete list of supported platforms.

- Hard Disk Drive Requirements

  In addition to the requirements for a WebLogic Server installation, an additional 13-MB of free storage space is required to install WebLogic JAM. Up to an additional 15-MB will be required temporarily for extracting the mainframe files

during installation. The mainframe files can be deleted after installation is complete.

The mainframe files require approximately 13-MB of free storage space on OS/390 Unix. An additional 5-MB of free storage is required during installation for extracting the compressed files.

The mainframe files require approximately 30 cylinders of free storage space on MVS. An additional 20 cylinders of free space is required during installation for extracting the compressed files.

■ Memory Requirements

For a Microsoft Windows or Unix system, adding a basic WebLogic JAM configuration to a WebLogic Server installation does not significantly increase the system RAM requirements. But increasing the number of service definitions will increase the memory requirement for WebLogic JAM.

Verifying WebLogic JAM Software Requirements

Verify that the following software is installed before you install WebLogic JAM:

■ Java Development Kit (JDK)

A JDK is required by the InstallAnywhere installation product to install the WebLogic JAM software. A JDK is installed with WebLogic Server. If you are installing WebLogic JAM without first installing WebLogic Server, set your PATH environment to the appropriate JDK directory.

■ WebLogic Server

WebLogic Server is required for proper operation of WebLogic JAM. If you attempt to install WebLogic JAM without installing WebLogic Server, a warning will display, but you can continue installing WebLogic JAM.

Choosing a BEA Home Directory

During installation, WebLogic JAM prompts you to choose an existing BEA Home directory or to specify a path for a new BEA Home directory. If you choose to create a new directory, the WebLogic installer program automatically creates the directory for you.

Note: For a BEA Home directory, you are allowed to install only one instance of each type and version of a BEA product that uses the BEA Home directory convention. For example, you can install only one instance of WebLogic JAM 5.1 in a BEA Home directory, but that BEA Home directory may also contain another version of WebLogic JAM, such as WebLogic JAM 5.0.

Creating More Than One BEA Home Directory

Although it is possible to create more than one BEA Home directory, BEA recommends that you avoid doing so. In almost all situations, a single BEA Home directory is sufficient. In some circumstances, however, you may prefer to maintain separate development and production environments, each containing a separate product stack. With two directories, you can update your development environment (in a BEA Home directory) without modifying the production environment until you are ready to do so.

3 Installing WebLogic Java Adapter for Mainframe

After you have reviewed the "Preparing to Install BEA WebLogic Java Adapter for Mainframe" section, you are ready to begin installing WebLogic JAM. Instructions for installing WebLogic JAM are provided in the following sections:

- Locating the Installation Files

- Installing the WebLogic JAM Gateway

- Installing the CRM

Locating the Installation Files

Locate the installation files:

- `wljam51_Win.exe`

- `wljam51_Unix.bin`

If you purchased WebLogic JAM, select the file that corresponds to your system. If you downloaded WebLogic JAM from the Web for evaluation, only the file you requested will be downloaded.

Installing the WebLogic JAM Gateway

Determine the platform on which you will install the WebLogic JAM Gateway. The Gateway can be installed in the following ways:

- In Graphical Mode on Microsoft Windows or Unix

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

  To run Graphical-mode installation, the console attached to the machine on which you are installing the software must support a Java-based GUI. All consoles for Microsoft Windows systems support Java-based GUIs, but not all consoles for Unix systems do.

- 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 Gateway Using Graphical-Mode Installation

Graphical-mode installation of WebLogic JAM may be used on Microsoft Windows or Unix systems. This installation will install the WebLogic JAM Gateway and the files for the installation of the CRM.

Prepare to Install the WebLogic JAM Gateway

To prepare to install the WebLogic JAM Gateway using the graphical mode, complete the steps in the following section that correspond to your platform.

On Microsoft Windows Platforms

Prepare to install the WebLogic JAM Gateway on Microsoft Windows platforms by completing the following steps:

1. Verify that the correct version of the JDK is in the system path. See the *BEA WebLogic Java Adapter for Mainframe Release Notes* for the required version. For example, enter the following command at a command prompt:

   ```
   java -version
   ```

   The java version will display. For example:

   ```
   java version "1.3.1_02"
   ```

   If the version displayed is lower than the required version, you will need to configure your PATH environment variable to use the required JDK that is installed with WebLogic Server.

2. Access the wljam51_Win.exe file in one of the following ways:

   ● Download the .exe file from the Web. Click on the .exe file to begin the installation.

   ● Insert the WebLogic JAM CD-ROM in the appropriate drive. The wljam51_Win.exe should automatically run.

   If it does not, click **Run** from the **Start** menu. The **Run** window displays.

   ■ Click **Browse** to navigate to the CD-ROM drive.

   ■ Select the wljam51_Win.exe file.

   ■ Click **Open** to make the path and executable appear in the **Run** dialog box, then select **OK** to run the executable and begin the installation.

   The InstallAnywhere screen displays. Continue with the installation of the WebLogic JAM Gateway by completing the steps in the "Install WebLogic JAM Gateway" section.

On Unix Platforms

Prepare to install the WebLogic JAM Gateway on Unix in graphical mode by completing the following steps:

1. Verify that the correct version of the JDK is in the system path. See the *BEA WebLogic Java Adapter for Mainframe Release Notes* for the required version. For example, enter the following command at a command prompt:

```
java -version
```

The java version will display. For example:

```
java version "1.3.1_02"
```

If the version displayed is lower than the required version, you will need to configure your PATH environment variable to use the required JDK that is installed with WebLogic Server.

2. Access the `wljam51_Unix.bin` file in one of the following ways:

   - Download the `.bin` file from the Web.

   - Access the CD-ROM device.

     ```
     > ls -1 /dev/cdrom
     total 0
     brw-rw-rw-   1 root    sys    22,   0 April 14 10:55 c1b0t010
     ```

     - Mount the CD-ROM. Mount commands vary depending on your system configuration. See your System Administrator for the specific command for your system.

     - Change the directory to the `mnt` directory on your CD-ROM device. For example,
       ```
       > cd /mnt
       ```

     - List the CD-ROM contents.
       ```
       ls
       wljam51_Win.exe
       wljam51_Unix.bin
       ```

3. Set the JAVA_HOME and PATH environment variables. For example, if your BEA Home directory is `/usr/bea` and you are using the JDK installed in the BEA Home directory, then the environment variables should be set in the following way:

   ```
   export JAVA_HOME=/usr/bea/jdk131_02
   ```

   ```
   export PATH=/usr/bea/jdk131_02/bin:$PATH
   ```

4. If you are installing remotely (from a machine other than the one upon which you are installing WebLogic JAM), define the DISPLAY environment variable as follows:

   ```
   export DISPLAY=xx.xx.xx.xx:0.0
   ```

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

> **Note:** If you are installing remotely from Microsoft Windows, you must use an X-Windows terminal emulator.

5. Execute the installation program.

```
sh wljam51_Unix.bin
```

The InstallAnywhere screen displays. Continue with the installation of the WebLogic JAM Gateway by completing the steps in the "Install WebLogic JAM Gateway" section.

Install WebLogic JAM Gateway

After the InstallAnywhere screen displays, the BEA WebLogic Java Adapter for Mainframe splash screen displays, followed by the Introduction screen.

1. Select **Next** on the Introduction screen to begin the installation.

   Note: To stop the installation process and exit the program, click **Exit** on any screen.

The License Agreement screen displays.
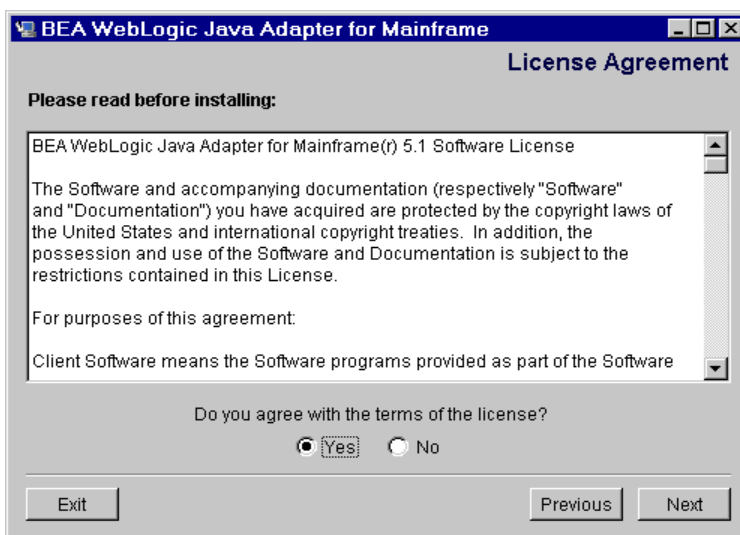
2. After reading the license agreement, select **Yes** to accept the license agreement. Click **Next** to proceed with the installation.

   Note: If you do not accept the license agreement, you will not be able to continue with the installation. You must either exit the installation or go back to the previous screen.

   The Choose Install Set screen displays.
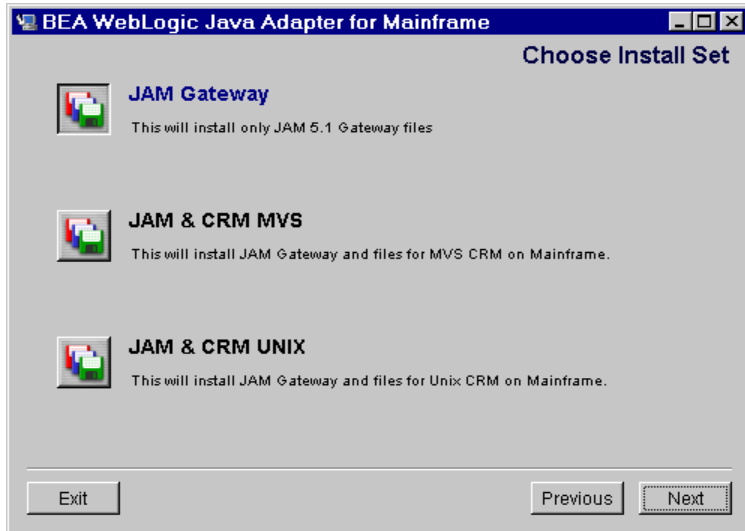
3. Select the Install Set that corresponds to your system from the following list. Click **Next**.

   - JAM Gateway

     Installs only the WebLogic JAM Gateway.

   - JAM & CRM MVS

     Installs the WebLogic JAM Gateway and copies the CRM install files for the z/OS and OS/390 MVS platform.

   - JAM & CRM Unix

     Installs the WebLogic JAM Gateway and copies the CRM install file for z/OS and OS/390 Unix.

   The Choose BEA Home Directory screen displays.
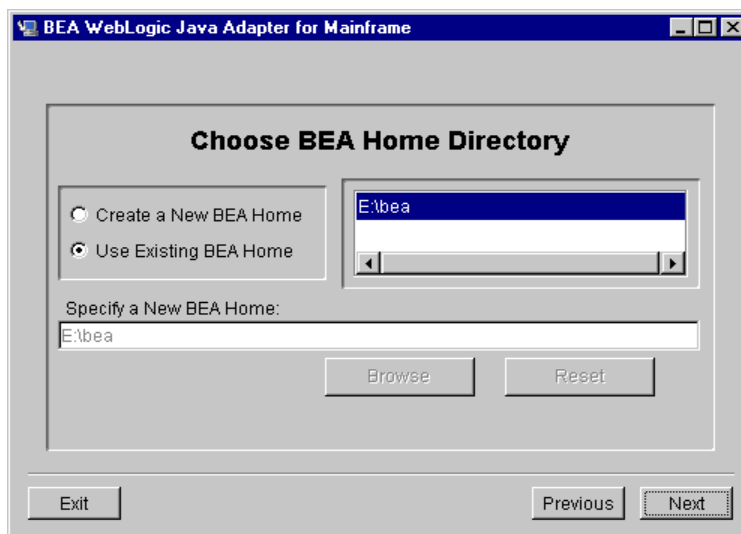
4. Select one of the following options, then click **Next**.

   - **Use Existing BEA Home** is the default setting for the BEA Home directory where WebLogic Server is located.

   - **Create New BEA Home** allows you to enter the location for a new directory or to click **Browse** to select a new location.

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, any BEA Home directory containing WebLogic JAM must also contain WebLogic Server.

If you already have WebLogic JAM installed in the BEA Home directory you have selected, the installation program will notify you. You can choose to continue installing WebLogic JAM into the same directory, but the new installation will overwrite existing patches and configuration files.

Note: If WebLogic Server is not installed, you will receive the following warning. If you click **Next**, installation continues. However, WebLogic JAM will not function fully unless WebLogic Server is installed. If WebLogic Server is not installed first, many automated configuration tasks must be completed manually.
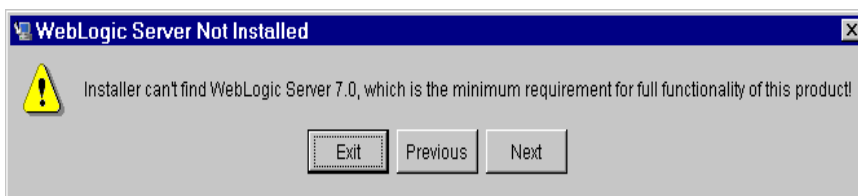
The Choose Product Directory screen displays.
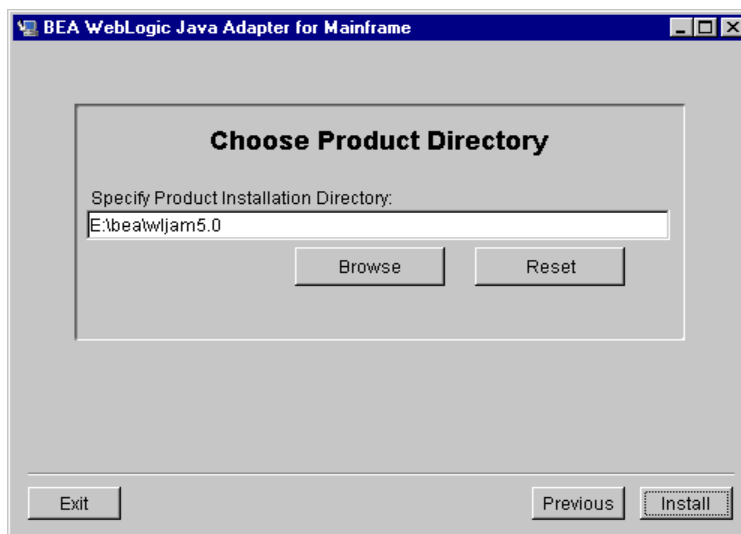
5. Click **Install** to accept the default product directory and continue with the installation. WebLogic JAM will create a `wljam5.1` folder in the BEA Home directory. If you want to select a different product directory, click **Browse**. If you want to return to the default directory, click **Reset**.

   The Install Complete screen displays.

6. Click **Done** to exit the installation program.

   On a Microsoft Windows platform, WebLogic JAM will create or update a Microsoft Windows program group named BEA WebLogic E-Business Platform. The WebLogic JAM folder will be "WebLogic Java Adapter for Mainframe 5.1." Groups of shortcuts are listed within the program group folder.

7. Install the CRM as described in the "Installing the CRM" section. After installing the CRM, see the "Verifying Your Installation" section to verify the files and run the installation verification sample. Complete post-installation tasks located in the "Performing Post-Installation Tasks" section.

Installing the WebLogic JAM Gateway on Unix Systems Using Console-Mode Installation

To install the WebLogic JAM Gateway and the files for the installation of the CRM on Unix systems using the console-mode installation, run the BEA standard installation shell script for the platform on which you want to install WebLogic JAM. Select the following script:

```
wljam51_Unix.bin
```

To install WebLogic JAM on a Unix system using console mode, perform the following steps:

1. Access the `wljam51_Unix.bin` file in one of the following ways:

   - Download the `.bin` file from the Web.

   - Access the CD-ROM device.

     ```
     > ls -1 /dev/cdrom

     total 0

     brw-rw-rw-    1 root     sys     22,   0 April 14 10:55 c1b0t010
     ```

     - Mount the CD-ROM. Mount commands vary depending on your system configuration. See your System Administrator for the specific command for your system.

     - Change the directory to the `mnt` directory on your CD-ROM device. For example,
       ```
       > cd /mnt
       ```

     - List the CD-ROM contents.
       ```
       ls
       wljam51_Win.exe
       wljam51_Unix.bin
       ```

2. Set the `JAVA_HOME` and `PATH` environment variables. For example, if your BEA Home directory is `/usr/bea` and you are using the JDK installed in the BEA Home directory, then the environment variables should be set in the following way:

   ```
   export JAVA_HOME=/usr/bea/jdk131_02

   export PATH=/usr/bea/jdk131_02/bin:$PATH
   ```

3. Execute the appropriate installation program for your non-mainframe Unix system in character-based mode. For example, to execute the HP/UX installation program, enter the following:

```
sh wljam51_Unix.bin -i console
```

4. The installation program runs and prompts you for responses.

Listing 3-1   Console-Mode Installation Example

```
Preparing to install...

Preparing CONSOLE Mode Installation...

================================================================
BEA WebLogic Java Adapter for Mainframe(created with
InstallAnywhere by Zero G)
----------------------------------------------------------------
================================================================
Introduction
------------

This installer will guide you through the installation of 'WebLogic
Java Adapter for Mainframe'. On each step, answer the questions
provided. Type "quit" at any time to exit the installer.

PRESS <ENTER> TO CONTINUE: <ENTER>

================================================================
License Agreement
-----------------

Please read before installing:
BEA WebLogic Java Adapter for Mainframe(r) 5.1 Software License

The Software and accompanying documentation (respectively
"Software"and "Documentation") you have acquired are protected by
the copyright laws of the United States and international copyright
treaties. In addition, the possession and use of the Software and
Documentation is subject to the restrictions contained in this
License.
   . . . .

DO YOU AGREE WITH THE TERMS OF THE LICENSE? (Y/N):  y

================================================================
Choose Install Set
------------------
```

```
Please Choose the Install Set to be installed by this installer.

  ->1- JAM Gateway

    2- JAM & CRM MVS

    3- JAM & CRM UNIX


ENTER THE NUMBER FOR THE INSTALL SET, OR <ENTER> TO ACCEPT THE
DEFAULT: 1

================================================================
Choose BEA Home Directory
-------------------------

    1- Create a New BEA Home

    2- Use Existing BEA Home


Enter a number: 2

    1- /bea/loads

    2- /cmhome/wls61sp2

Existing BEA Home: 1

------------------------------------------------
A previous installation of WebLogic Java Adapter for Mainframe 5.1
has been detected at /bea/loads/wljam5.1. The following options are
available:

    1- Continue to install over /bea/loads/wljam5.1

    2- Cancel to select another BEA Home Directory

    3- Exit the installation

Choose a number: 1

================================================================
Installing...
-------------
 [=============|=============|=============|=============]


[-------------|-------------|-------------|-------------]
```

```
================================================================
Install Complete
----------------

Congratulations!

You have successfully installed BEA WebLogic Java Adapter for
Mainframe.To verify the success of your installation, refer to the
WebLogic JAM Installation Guide at the BEA corporate Web site for
instructions on running installation verification samples for IMS
and CICS regions.

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

Note: The single-dash bar just beneath the double-dash bar in Listing 3-1 is a progress indicator. When the single-dash bar becomes the same length as the double-dash bar, installation is complete.

5. Install the CRM as described in the "Installing the CRM" section. After installing the CRM, see the "Verifying Your Installation" section to verify the files and run the installation verification sample. Complete post-installation tasks located in the "Performing Post-Installation Tasks" section.

Installing the CRM

Installation files for the CRM are placed in the `<BEA_HOME>/<JAM_INSTALL_DIR>/Mainframe` directory during the installation process based on your selection of the install set. To install the CRM, use FTP to send the CRM files to the mainframe and run the specific install procedure for that platform. The following sections provide installation instruction for supported platforms.

Note: The mainframe, CICS, and IMS do not need to be restarted for a WebLogic JAM installation.

Installing the CRM on z/OS and OS/390 Unix

If you selected the JAM & CRM Unix install set during the installation of the WebLogic JAM Gateway, complete the following steps to install the CRM on the mainframe in a z/OS and OS/390 Unix environment.

Note: This procedure assumes the installer is familiar with File Transfer Protocol (FTP) functions.

1. Create a temporary directory in your Unix file system for the installation files.

2. FTP the file `os390.tar` (in binary mode) from
   `<BEA_HOME>/<JAM_INSTALL_DIR>/Mainframe`

3. To extract the installation script from the `tar` file on the mainframe, use the following command:

   `tar xvf os390.tar`

   Executing the `tar` command extracts the `install.sh` script and subdirectories containing the software to be installed from the `tar` file.

4. Execute the `install.sh` script using the following command:

   `sh install.sh`

The installation script prompts you for responses. Listing 3-2 is a sample installation that shows the installation of the CRM on the mainframe in an OS/390 Unix environment. User responses are in bold.

Listing 3-2   Sample Script for CRM Installation on an OS/390 Unix Platform

```
>sh install.sh

01) os390/os390r8

Install which platform's files? [01-   1 , q to quit, l for list]: 1

** You have chosen to install from os390/os390r8 **

Communications Resource Manager Release 5.0

This directory contains the Communications Resource Manager 5.0
for IBM OS/390 Unix R8 on IBM OS/390.
```

```
Is this correct? [y,n,q]: y

To terminate the installation at any time press the interrupt key,
typically <del>, <break>, or <ctrl+c>.

The following packages are available:
 1  crm      Communications Resource Manager

Select the package(s) you wish to install [?,??,q]: 1

Communications Resource Manager
OS390) Release 5.0
Copyright (c) 2000 BEA Systems, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.

Directory where Communications Resource Manager files are to be
installed
(Enter your directory path) [?,q]:

Using /work/user/jam as the Communications Resource Manager base
directory

The following CRM stack support options are available:

  1     vtm28          OpenEdition MVS 390

Select an option [?,??,q]: 1

Determining if sufficient space is available ...
24152 blocks are required
8476704 blocks are available to /work/user/jam

Unloading /bea/loads/dist/juniper-9/os390/os390r8/crm/CRMSA.Z ...
2415 blocks
x bin/CRM
x bin/CRMLOGS
x bin/crmdown
x bin/crmlkoff
x bin/crmlkon
x bin/unix.env
x lib/csxappc.dll
x lib/csxcrm.dll
x lib/csxgpw.dll
x lib/csxscrm.dll
x lib/csxxfm.dll
x lib/ctxdebugs_12.dll
x lib/ctxos_12.dll
x lib/ctxplat_12.dll
x lib/ctxprim_12.dll
x lib/libgp40.dll
```

```
x lib/libgpnet.dll
... finished

Unloading /bea/loads/dist/juniper-9/os390/os390r8/crm/STKSA.Z ...
226 blocks
x lib/csxvtm28.dll
... finished

Changing file permissions...
... finished

Installation of Communications Resource Manager was successful

Please don't forget to fill out and send in your registration card
```

Installing the CRM on z/OS and OS/390 MVS

If you selected the JAM & CRM MVS install set during the installation of the WebLogic JAM Gateway, complete the following steps to install the CRM on the mainframe in a z/OS and OS/390 Unix environment.

The installation for this environment is semi-automated. Two JCL files are provided to facilitate prerequisites for allocating data sets in the MVS environment and to unload WebLogic JAM product software into the allocated data sets.

The following files relating to this installation procedure are downloaded during the installation of the WebLogic JAM Gateway:

- `Readme` file contains notes on how to install the product in this environment. This file is a summary of the procedure in this section.

- The `createds.jcl` is a JCL example for creating MVS data sets to receive the WebLogic JAM software.

- The `uncmprss.jcl` file is a JCL example to uncompress and unload into the product data sets.

- `JAM50.LOAD` file contains the WebLogic JAM program objects. (Required)

- `JAM50.DATA` file contains WebLogic JAM data files. (Required)

- `JAM50.SAMPLE` file contains sample JCL files.

Installation Procedure for CRM on z/OS and OS/390 MVS

Perform the following steps to install the CRM component on a z/OS and OS/390 MVS platform.

Note: The installation process assumes the installer is familiar with Job Control Language (JCL) job operations and File Transfer Protocol (FTP) functions.

You may transfer the files with IND$FILE protocol, if you prefer, provided the text files are transferred in text mode, and the LOAD, DATA, and SAMPLE files are in binary mode. See your Terminal Emulation Software for instructions.

Step 1: Upload the Install JCL Files to the MVS Environment

1. Identify the install JCL files that were downloaded to the <BEA_HOME>/<JAM_INSTALL_DIR>/Mainframe directory during the installation of the WebLogic JAM Gateway. The installation JCL files will be used to allocate data sets for receiving the product distribution files.

2. Using FTP, send the following JCL files in ASCII to the MVS system using any PDS you choose as the destination.

   createds.jcl

   uncmprss.jcl

   Note: Use conventional MVS naming standards for the destination member name. For example

   ```
   ftp>put createds.jcl CREATEDS
   ```

Step 2: Allocate the Initial Data Sets

1. Change the CREATEDS job to suit your environment. This JCL allocates the data sets needed to receive and install the product software files from the product distribution.

   Use the SET statements in the CREATEDS job to set the high-level qualifiers of the receiving data sets according to the following syntax:

   ```
   SET HLQPDSE=[QUALIFIER]
   ```

   In this statement, [QUALIFIER] is the high-level qualifier for the CRM load library. The load library containing the JAM50.LOAD program objects must be a partitioned data set with extended attributes (PDSE).

```
SET SCLASS=[CLASS NAME]
```

In this statement, `[CLASS NAME]` is the storage class name of the PDSE. This parameter is required for OS/390 v2.8, but can be left blank for most systems. The PDSE minimally must have storage class SMS declaration for OS/390 v2.8. See the *BEA WebLogic Java Adapter for Mainframe Release Notes* for detailed information about platform support.

```
SET HLQPDS=[QUALIFIER]
```

In this statement, `[QUALIFIER]` is the high-level qualifier for the sample and data sets.

```
SET TMPHLQ=[QUALIFIER]
```

In this statement, `[QUALIFIER]` is the high-level qualifier for the temporary data sets. These temporary data sets will be used to receive the distribution files from the FTP.

The CREATEDS job has the following steps:

```
DELETE
```

Deletes any existing data sets previously created so you can run this JCL multiple times, if necessary.

```
CREATE
```

Allocates the data sets needed to receive and install the product software files.

2. Run the CREATEDS job.

Step 3: FTP the Files to the Allocated Data Sets

Using FTP, put the following product files into their corresponding data sets. The data sets were created in the previous step and have the high-level qualifier identified by TMPHLQ.

- JAM50.LOAD

- JAM50.DATA

- JAM50.SAMPLE

- From the Unix or Microsoft Windows command prompt, set the binary data transfer mode using the FTP program.

Transfer the data sets using the PUT command.

Note the following example of a PUT command:

```
put JAM50.LOAD 'BEA.JAM50.TEMP.LOAD'
```

Step 4: Unload the Data in Data Sets

1. Change the UNCMPRSS job to suit your environment.

   Use the SET statement in the UNCMPRSS job to set the high-level qualifiers of the temporary data sets using the same value that was used in CREATEDS:

   ```
   SET TMPHLQ = <TMPHLQ>
   ```

   In this statement, TMPHLQ identifies the high-level qualifier used for the temporary data sets created in step 1.

   Modify the following variables in the SYSIN JCL to the high-level qualifiers used in step 1:

   - HLQPDSE

     The value from the SET HLQPDSE in CREATEDS

   - HLQPDS

     The value from the SET HLQPDS in CREATEDS

   The UNCMPRSS job has the following steps:

   UNLOAD

       Performs a TSO RECEIVE to uncompress and unload the data in your installation data sets created with the CREATEDS job. This step will delete the temporary data sets created by CREATEDS.

   Note: Instead of running the UNCMPRSS job, you may choose to manually perform the TSO RECEIVE command. This command unloads the data in your temporary installation data sets. Run TSO RECEIVE in the following way:

   ```
   TSO RECEIVE INDS('xxx')
   ```

   When prompted, enter:

   ```
   DA("yyy")
   ```

In these statements, the following definitions apply:

xxx

Is the temporary data set from CREATEDS. For example, [TMPHLQ].LOAD.

yyy

Is the data set that from CREATEDS that corresponds to the HLQPDS data sets and the HLQPDSE data set.

2. Run the UNCMPRSS job.

What Do I Do Next?

After you have installed the WebLogic JAM components, BEA recommends that you verify your installation as described in the "Verifying Your Installation" section. After you have verified your installation, see the "Performing Post-Installation Tasks" section for additional tasks that must be performed before you begin to use WebLogic JAM.

If you choose not to use the installation verification sample, see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide* for information about starting and administering the CRM.

4 Performing Post-Installation Tasks

After you have completed the installation process and verified your installation, complete the post-installation tasks in the following sections to ensure that WebLogic JAM operates properly.

■ Updating Your license.bea File

■ Deploying WebLogic JAM in the WebLogic Server Environment

Updating Your license.bea File

You must update your license.bea file if at least one of the following is true:

■ You are downloading the 60-day extension to your 30-day evaluation period of WebLogic JAM.

■ You installed WebLogic JAM from the CD-ROM.

■ You possess a non-expiring license file for a previous version of WebLogic JAM, and you want to upgrade to a WebLogic JAM license after downloading WebLogic JAM from the BEA Web site.

In each case, you receive a license update file in an e-mail attachment. Update your license.bea file by performing the following steps for your system.

Updating Your license.bea File on a Microsoft Windows System

To update your existing `license.bea` file on a Microsoft Windows system follow these steps:

1. Log in to the Microsoft Windows system.

2. Save the license update file that you received through e-mail with a name other than license.bea in the target BEA Home directory.

3. Open a command prompt window.

4. Navigate to the target BEA Home directory.

5. Include the path to the JDK at the front of your `PATH`. Note the following example:

   ```
   set PATH=.\jkdl31_02\bin;%PATH%
   ```

6. Merge the license update file into your existing license by entering the following command:

   ```
   UpdateLicense license_update_file
   ```

   license_update_file is the name to which you saved the license update file that you received through e-mail. Running this command updates your `license.bea` file.

7. Save a copy of your license.bea file in a safe place outside the WebLogic JAM product directory. Although no one else can use your license file, you should save this information in a place protected from either malicious or innocent tampering by others.

Updating Your license.bea File on a Unix System

To update your existing `license.bea` file on a Unix system, follow these steps:

1. Log in to the Unix system.

2. Save the license update file that you received through e-mail with a name other than license.bea in the target BEA Home directory.

3. Open a command prompt window.

4. Navigate to the target BEA Home directory.

5. Include the path to the JDK at the front of your `PATH`. Note the following example:

```
PATH=./jkd131_02/bin;$PATH

export PATH
```

6. Merge the license update file into your existing license by entering the following command:

```
sh UpdateLicense.sh license_update_file
```

license_update_file is the name to which you saved the license update file that you received through e-mail. Running this command updates your `license.bea` file.

7. Save a copy of your license.bea file in a safe place outside the WebLogic JAM product directory. Although no one else can use your license file, you should save this information in a place protected from either malicious or innocent tampering by others.

Deploying WebLogic JAM in the WebLogic Server Environment

After WebLogic JAM has been installed, it must be deployed in the WebLogic Server environment. The following tasks must be completed if you are not running a pre-configured sample.

You can deploy WebLogic JAM in the following ways:

■ If you are creating applications, the `example` domain and the `verify` domain are examples of domains deployed in the WebLogic Server environment. You may want to use these domains as a starting point, adding other J2EE applications to meet your specific requirements.

- ■ If you are adding WebLogic JAM to existing WebLogic Server applications, you need to manually deploy WebLogic JAM by completing the following steps:

  - Adding the `jam.jar` file to the WebLogic Server CLASSPATH

  - Deploying the `jam.ear` file

  - Defining the WebLogic JAM startup class

See the BEA WebLogic Java Adapter for Mainframe *Configuration and Administration Guide* for additional information about deploying WebLogic JAM.

5 Verifying Your Installation

When you have completed the installation process, BEA WebLogic Java Adapter for Mainframe (WebLogic JAM) provides the following ways to verify your installation

- Verifying Installation of Files and Directories

- Using the Installation Verification Samples

Verifying Installation of Files and Directories

After you have installed WebLogic JAM, you can verify the files and directories that are installed on your system. For a list of these files, see Appendix A, "BEA WebLogic Java Adapter for Mainframe Installation Files and Directories."

Using the Installation Verification Samples

After you have installed WebLogic JAM, run the installation verification sample that corresponds to your system:

■ Installation verification sample for IMS

■ Installation verification sample for CICS

About the Installation Verification Samples

As the name implies, the installation verification samples are designed to allow you to verify that WebLogic JAM is installed and running on your system. The samples allow a client to make a request of a mainframe server program on either IMS or CICS.

The samples are designed to run "out of the box" with very limited modification. The installation verification samples include:

■ Java Files

  ● Java `.class` files

  ● Java source

  ● eGen scripts

  ● Build scripts

  ● Preconfigured WebLogic Server domain

■ Mainframe Files

  ● Mainframe source

  ● Sample JCL

  ● IMS or CICS sample configurations

> **Note:** JCL to compile the mainframe sample programs is included as examples to provide completeness. The JCL that is shipped may not conform to your standards or installations. The sample JCL may need to be modified or replaced to meet your needs.
>
> Program definitions and other configuration for your IMS or CICS region may require coordination with the IMS or CICS system programmer.

Most configuration tasks are preconfigured or completed during the installation process. Working from the WebLogic Administration Console, you can make the modifications required for the sample to run on your system. When these modifications are complete, you can run the simple client supplied by the sample to verify that WebLogic JAM is installed and running properly. The installation verification sample runs in the supplied `verify` domain. The client uses the Gateway to send a request from the WebLogic Server environment to the mainframe. The mainframe server converts the supplied string from mixed case to uppercase.

The only difference between the IMS and the CICS sides of this sample is that for IMS, `TOUPPER` is an APPC Service that is mapped to an IMS transaction; for CICS, `TOUPPER` is a DPL Service that is mapped to a CICS program.

How to Use the Samples

While the WebLogic JAM samples contain precompiled source for the Java portion of the sample, you may choose to generate and compile source. Depending on your skill, experience, and time-constraints, you may choose to work with the Java portion of the samples in the following ways:

■ Run the sample

   Because of the contents shipped with the samples, in most cases, you can simply run the sample with minimal configuration to run on your system. The sample will use the supplied `.class` files.

■ Compile source and run the sample

   Each sample provides build scripts that allow you to compile the source. This option allows you to see how the source is compiled and run the sample.

■ Generate source, compile, and run the sample

Each sample provides eGen scripts. This option allows you to use eGen Application Code Generator to generate source. Afterward, the build scripts can be run. By using this option, you can see how the eGen utility generates source and run the sample.

Preconfigured WebLogic Server Domains

The WebLogic JAM installation includes a directory named config. This config directory contains subdirectories for two preconfigured WebLogic Server domains:

■ verify

WebLogic Server domain that is set up for running the installation verification. verify is the directory that contains the files necessary to configure and run the verify domain

■ examples

WebLogic Server domain that is set up for examples of WebLogic JAM running with WebLogic Server and mainframe applications

Roadmap for Installation Verification Samples

To run each of the installation verification samples, follow the roadmap listed below:

1. Verify prerequisite tasks.

For a listing of prerequisite tasks, see the "Before You Use the Samples" section.

2. Set up the sample.

See the "Setting Up the Sample" for IMS or the "Setting Up the Sample" section for CICS.

Before You Use the Samples

a. Start the CRM.

b. Update the WebLogic JAM configuration file.

c. Start the `verify` domain.

d. Configure the WebLogic JAM Gateway.

e. Enable services.

f. Set the environment.

g. Generate and build source (optional).

h. Complete mainframe tasks.

3. Run the sample.

Before You Use the Samples

Before you run the samples, the following tasks must be completed:

■ Install WebLogic Server.

For information about installing WebLogic Server, see the BEA WebLogic Server documentation.

■ Install WebLogic JAM.

For information about installing WebLogic JAM, see the "Installing WebLogic Java Adapter for Mainframe" section of this guide.

■ Define the Logical Unit for the CRM and vary it active.

For information about defining the Logical Unit, see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide.*

- If using IMS, verify that the APPC communication to IMS is active.

  For information about APPC communication with WebLogic JAM, see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide.*

- If using CICS, the connection must be defined to the CICS region.

  For information about CICS connection with WebLogic JAM, see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide.*

Using the Installation Verification Sample for IMS

After completing the tasks in the "Before You Use the Samples" section, you are ready to set up and run the sample.

How the Sample Works

The program TOUPIMS is a simple COBOL IMS server program that dequeues the request data, converts the data to uppercase, then inserts the data back on the message queue. The copybook CHARDATA provides the structure of the user data in the request and response messages. No special considerations are required in this program as a result of being used in an application with a Java client making requests through WebLogic JAM.

Understanding the Sample Configuration

For this simple sample, no special configuration is required. The Java client calls the service TOUPPER.

- Gateway

  The APPC Service TOUPPER is mapped to the name of the IMS transaction program, TOUPIMS.

- On the mainframe

  The server program `TOUPIMS` is defined to the IMS region in the usual way any program is defined to an IMS region.

Understanding the Sample Programming

The programming for this sample is described in the following sections.

WebLogic Application

Three classes compose the WebLogic side of this sample application:

- `Chardata`

- `BaseClient`

- `Client`

`Chardata` is a DataView class that is generated by the eGen Application Code Generator. The data member in the `Chardata` class corresponds to the data field in the `CHARDATA` copybook. The `Chardata` class is responsible for all data translation between the mainframe format of the data and the Java format of the data.

The `BaseClient` class that is generated by the eGen Application Code Generator is an extension of the `EgenClient` class. `BaseClient` is the class that implements the actual calls to the `TOUPPER` mainframe service in its `toupper` method. The `toupper` method is a wrapper for the `callService` method of the `EgenClient` class with the `TOUPPER` service as a parameter.

The `Client` class is the actual user interface. The `Client` class has a `BaseClient` member. The `Client` class receives string input as a command line argument. You may also enter a URL if the WebLogic JAM Gateway is running on a different machine or the corresponding instance of WebLogic Server is listening on a different port than 7001. The URL is set in the `BaseClient` member. In the `Client` class, a `Chardata` DataView is initialized with the input data. The `toupper` method of the `BaseClient` member is called with this DataView as a parameter. The converted string value encapsulated in the returned DataView is then displayed.

IMS Program

The program TOUPIMS is a simple COBOL IMS server program that dequeues the request data, converts the data to uppercase, then inserts the data back on the message queue. The copybook, CHARDATA, provides the structure of the user data in the request and response messages. No special considerations are required in this program as a result of being used in an application with a Java client making requests through WebLogic JAM.

Sample Files

The files for the WebLogic side of the sample are installed in the following directory:

```
<BEA_HOME>/<JAM_INSTALL_DIR>/samples/verify/gateway/outbound
```

The following table lists the sample files and their purpose:

Table 5-1 Files for the WebLogic JAM Application

File Name	File Purpose
chardata.cpy	COBOL copybook that defines the structure of the string mainframe data.
chardata.egen	eGen script that generates the Chardata.java DataView class.
Chardata.java	DataView class that corresponds to the chardata.cpy COBOL copybook.
baseClient.egen	eGen script that generates the Chardata.java DataView class and the BaseClient.java EgenClient class.
BaseClient.java	Java class that extends EgenClient class that calls the TOUPPER service.
Client.java	The user-interface client class that receives the input string from the user, and displays the result of the TOUPPER service to the user. It invokes the TOUPPER service by calling the callService method of its BaseClient member.
build.cmd	Script that builds the Chardata, BaseClient, and Client classes. The built class files are under the `<BEA_HOME>\<JAM_INSTALL_DIR>\config\verify\clientclasses\verify\gateway\outbound` directory.

File Name	File Purpose
`build.sh`	Unix script that builds the `Chardata`, `BaseClient`, and `Client` classes. The built class files are under the `<BEA_HOME>/<JAM_INSTALL_DIR>/config/verify/clientclasses/verify/gateway/outbound` directory.

The files for the IMS side of the sample are installed in the following directory:

`<BEA_HOME>/<JAM_INSTALL_DIR>/samples/verify/IMS/outbound/source`

The following table lists the samples files and their purpose:

Table 5-2 Files for IMS Application

File Name	File Purpose
`CHARDATA`	COBOL copybook that defines the structure of the string input and output data.
`COMPTOUP`	JCL that compiles and links the `TOUPIMS` program.
`JAMDEFI`	Contains sample IMS stage 1 and `PSBGEN` input for the IMS configuration of the `TOUPIMS` program.
`TOUPIMS`	IMS server program that receives a string in mixed case and returns the string in uppercase.

Setting Up the Sample

To set up the installation verification sample for IMS, perform the following steps.

Step 1: Start the CRM

Before starting the WebLogic JAM Gateway, start the CRM. The CRM must be configured with certain parameter values at startup. These parameter values include:

- The address of the machine on which the CRM is running

- The port on which the CRM listens

- The name the Gateway will use to refer to the CRM

For running the samples, you must set the machine address and port. The values that you set for the machine address and port when the CRM is started, must agree with the values that you set for the CRM in the WebLogic Administration Console for the samples CRM. The name of the CRM that is preconfigured for running all of the samples is CRM1. Use this name when the CRM is started to run any of the samples.

The way you start the CRM depends on whether the CRM will be started under a Unix or MVS system. On Unix, start the CRM using a shell script. On MVS, start the CRM using JCL.

Starting the CRM on z/OS or OS/390 Unix

On z/OS or OS/390 Unix, you may use a script to start the CRM. Scripts are installed with the Gateway in the `<BEA_HOME>/<JAM_INSTALL_DIR>/samples/crm/unix` directory. The script, `crm.env`, appends the necessary values to your environment variables. The script, `startcrm.sh`, starts the CRM. To use these scripts, complete the following steps:

1. FTP the following two scripts to the directory from which the CRM will run:

   - `crm.env`
   - `startcrm.sh`

2. Edit `crm.env`. Supply the correct values for the `APPDIR` and `CRMDIR` variables. `APPDIR` is the directory from which the CRM will run. `CRMDIR` is the CRM installation directory.

3. Edit `startcrm.sh`. To use a different port than the default port, `7101`, change the port number. However, if you change the port number, make sure to change it in the corresponding field in the WebLogic Administration Console CRM1 pane. You do not need to change the address because the script will run on the machine where the CRM is installed.

Note: BEA recommends that you do not change the CRM name from CRM1. This name for the CRM is preconfigured for all of the samples.

4. Execute the `startcrm.sh` script:

```
. ./startcrm.sh
```

Compare Figure 4-1 with the script in Listing 4-1. Notice how the parameters in the script correspond to the fields in the WebLogic Administration Console. The script illustrates the values for `startcrm.sh` script parameters for running the samples.

Note: The port number is `7101`. You can change the port number; however, if the port number is changed make sure to change it in the corresponding field in the Gateway configuration CRM1 pane of the WebLogic Administration Console.

Figure 5-1   Fields for the CRM
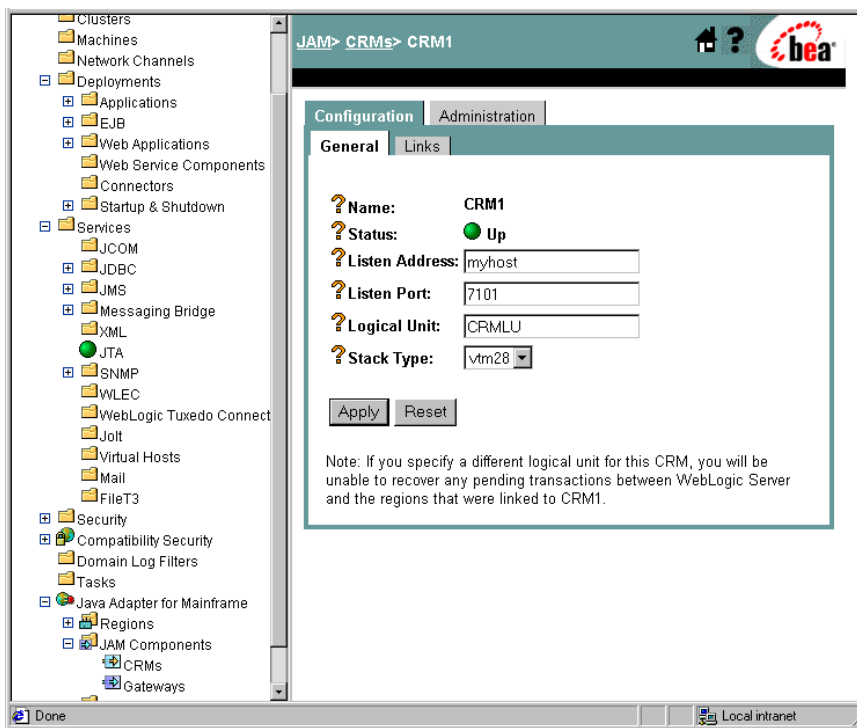
Listing 5-1   Command to Run the CRM

```
$CRMDIR/bin/CRM //127.0.0.1:7101 CRM1 < /dev/null > std.out
2>std.err &
```

Starting the CRM on z/OS or OS/390 MVS

On z/OS or OS/390 MVS, start the CRM by submitting the CRMSTART JCL that is installed with the CRM. The CRMSTART JCL must be modified for your environment. For information about modifying the CRMSTART JCL, see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*.

As you modify the CRMSTART JCL, make sure that you note the following parameters in the value of the STARTCMD parameter in the JCL. These parameters correspond to fields in the WebLogic Administration Console. These values must be the same in the JCL and in the WebLogic Administration Console.

- The machine address where the CRM will run

- The port number on which the CRM will listen

- The name by which the Gateway will refer to the CRM

Compare Figure 4-1 with the JCL in Listing 4-2. Notice how the parameters in the JCL correspond to the fields in the WebLogic Administration Console. The JCL illustrates the values for STARTCMD parameters for running the samples.

- The machine where the CRM will run is myhost in this sample. You must replace myhost with the hostname or IP address of your mainframe to allow IP-based communication from the Windows or Unix machine where WebLogic Server is running. You may verify this parameter with the ping command on Windows or Unix.

- The port number is 7101. You can change the port number; however, if you change the port number, make sure to change it in the corresponding field in the Gateway configuration CRM1 pane of the WebLogic Administration Console.

Note: BEA recommends that you do not change the CRM name from CRM1, because this name for the CRM is preconfigured for all of the samples.

Listing 5-2   The `STARTCMD` parameter in the `CRMSTART` JCL

```
//  SET STARTCMD='"//myhost:7101" CRM1'
```

Step 2: Update the WebLogic JAM Configuration File

On the machine where the Gateway is located, update the WebLogic JAM configuration file from the command prompt by completing the following steps:

1. Locate the `jamconfig_IMS.xml` file in the following directory:

   `<BEA_HOME>/<JAM_INSTALL_DIR>/config/verify`

2. Copy `jamconfig_IMS.xml` to `jamconfig.xml`.

Step 3: Start the verify Domain

From the command prompt, execute the following command from the same directory to start the `verify` domain:

- For Microsoft Windows:

  `startVerifyServer.cmd`

- For Unix:

  `. ./startVerifyServer.sh`

Step 4: Configure the WebLogic JAM Gateway

Most configuration tasks were preconfigured or were completed during the installation process by the installer program. For additional information about configuring WebLogic JAM, see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*. Make the following configuration changes for the installation verification sample for IMS to run on your system. These changes can be made in the WebLogic Administration Console in the following way.

1. From your browser, open the WebLogic Administration Console using the following address:

   ```
   http://hostname:7001/console
   ```

   In this address, the following definitions apply:

   `hostname` is the address of the machine where WebLogic Server is running.

   `7001` is the port for WebLogic Server that has been configured for the `verify` domain.

2. When prompted, supply the following user and password information:

   - user: system

     This user name cannot be changed.

   - password: security

     To change the password, see the BEA WebLogic Server documentation.

The WebLogic Administration Console displays.

3. To configure the CRM to the WebLogic JAM Gateway, complete the following steps:

a. In the left pane, click on **Java Adapter for Mainframe → JAM Components → CRMs**. In the right pane, click **CRM1**. On the **General** tab, set the following fields to correspond with your system. Click **Apply**. When the CRM is active, **Status** turns from red to green.

Field	Field Description
Listen Address	The address of the machine where the CRM is installed and running. This address must match the address set in the CRM startup JCL or script.
Listen Port	The port for the CRM. This entry must match the port set in the CRM startup JCL or script.
Logical Unit	The name of the Logical Unit defined for the CRM.
Stack Type	The stack type.

b. To configure the IMS region, click **Java Adapter for Mainframe** → **Regions** → **IMS Regions** in the left pane. In the right pane, click **VS10IMS**. Enter the Logical Unit name that supplies your IMS control region with APPC communication. You will find this name with the DISPLAY APPC operator command in IMS. Do not enter the Logical Unit name for APPLID of the IMS control region. This APPLID does not support APPC communication. If your IMS control region does not currently support APPC communication, you will need to set up this communication in APPC/MVS. Then activate the communication within IMS using the START APPC Operator command. Click **Apply** to set the Logical Unit.

Note: This Logical Unit is not the same as the Logical Unit for the CRM in (3a).
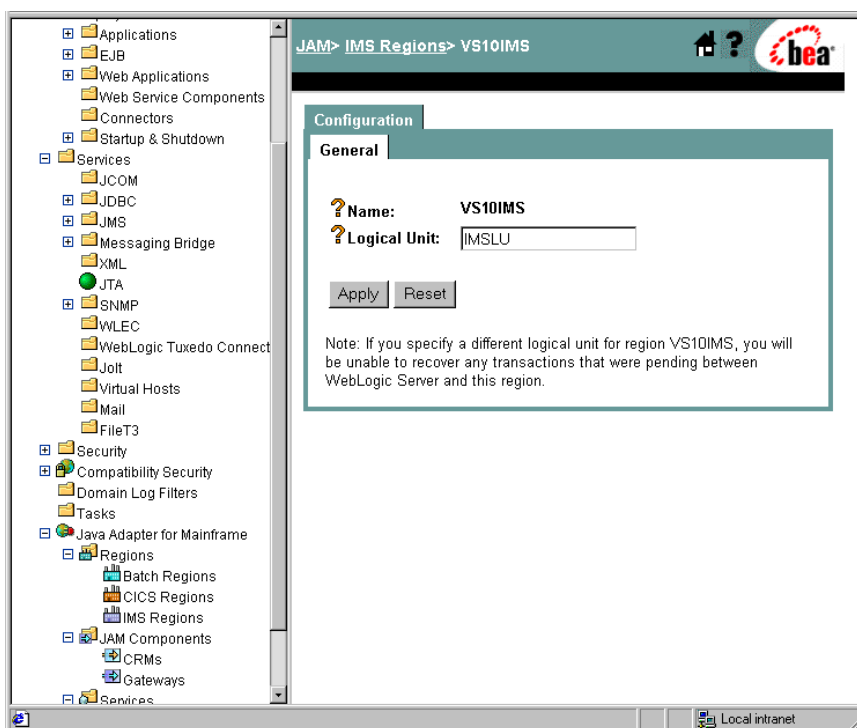
c. At the top of the right pane, click **IMS Regions**→**CRM1toIMS**. In the new window, click **CRM1toIMS**. On the **Links** tab, check **Deployed** and click **Apply**.
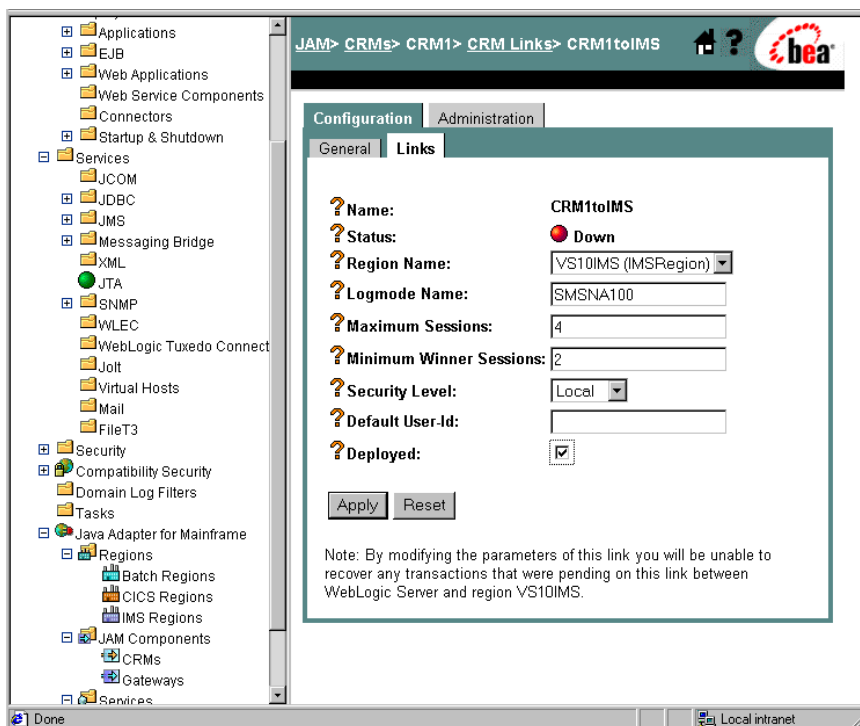
d. In the left pane, click **Gateways**. Click **JAM5.1** in the right pane. On the
    General tab, check **Deployed**. Click **Apply**.
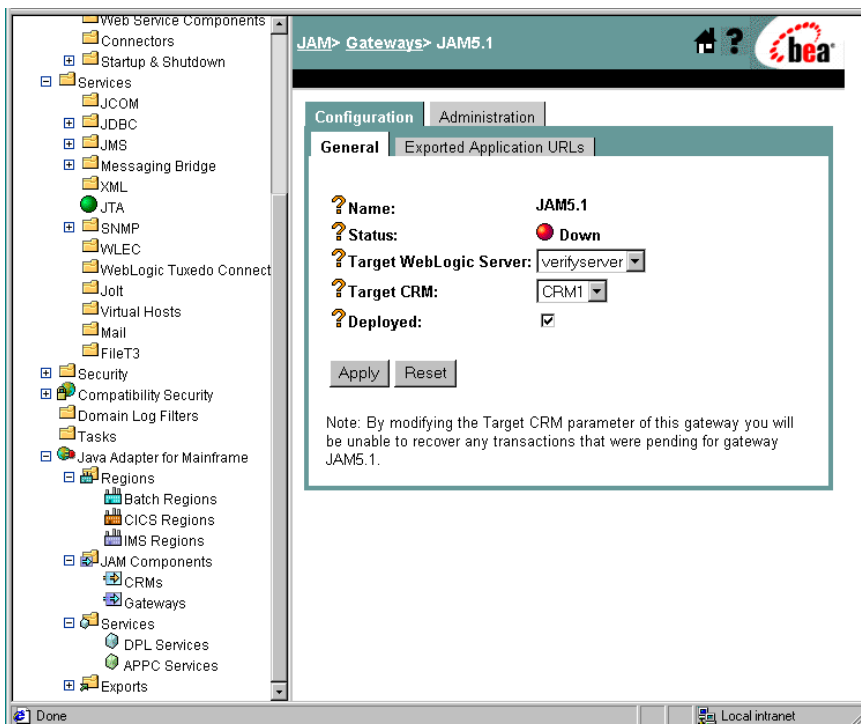
4. To start the Gateway, select the **Administration** tab → **Start/Stop** tab. Click **Start** to start the Gateway.
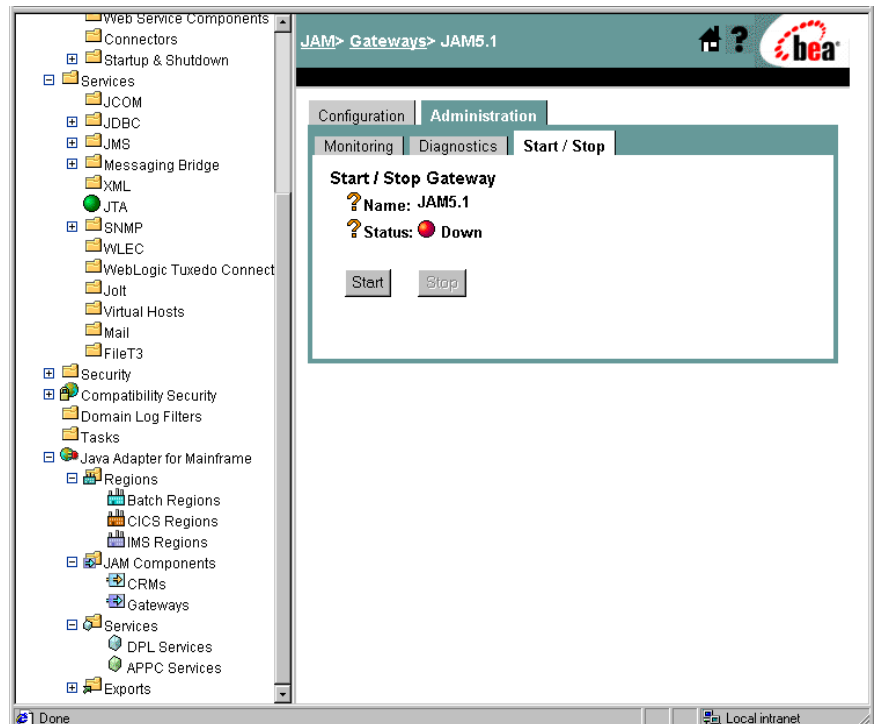
If the Gateway is running, **Status** changes to green in the WebLogic Administration Console and the following message appears in the WebLogic Server log:

"JAM Gateway ready for use. Current link status: up(1)."

Step 5: Enable the Service

To enable the APPC service, click **Services** → **APPC Services** in the left pane. In the right pane, click **TOUPPER**. On the **General** tab, check **Enabled** and click **Apply**.
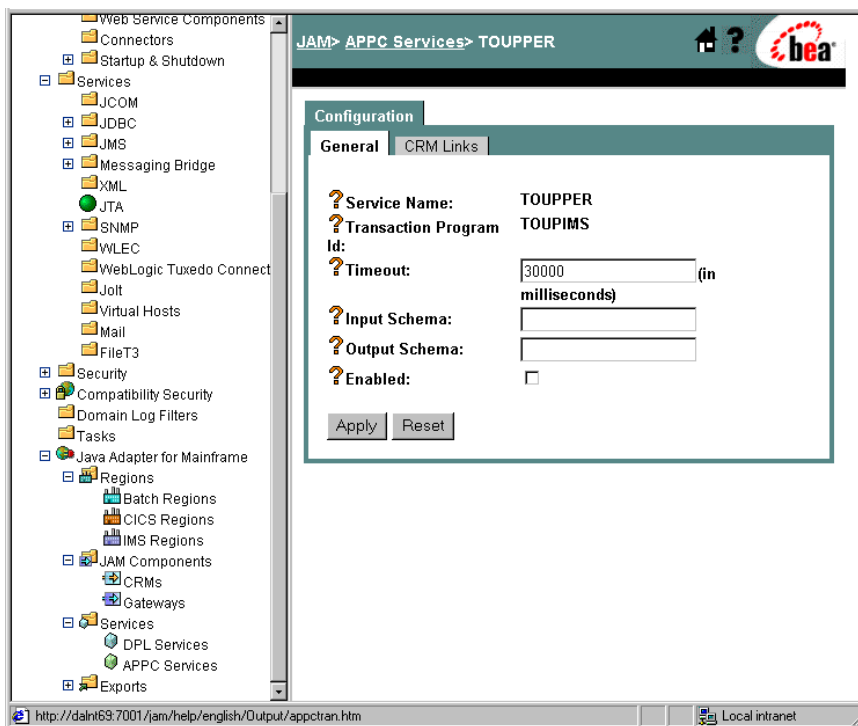
Step 6: Set the environment

On the machine where the Gateway is installed, set the environment by completing the following step:

- From a command prompt, change to the `<BEA_HOME>/<JAM_INSTALL_DIR>/config/verify` directory and execute the following command to set the environment:

  For Microsoft Windows:

  ```
  setVerifyEnv.cmd
  ```

For Unix:

```
. ./setVerifyEnv.sh
```

The following message will display:

"Your environment has been set."

Step 7: Generate and Build Source (Optional)

The WebLogic JAM samples provide generated source. Compiled `.class` files are also provided. If you want to see how the source is generated and the classes are built, change to the
`<BEA_HOME>/<JAM_INSTALL_DIR>/samples/verify/gateway/outbound`
directory and complete the following steps.

Warning: Using the following options will overwrite files that are installed with the WebLogic JAM samples.

■ Run the `build.cmd (sh)` script to build the client classes. The build script puts the built `.class` files in the
`<BEA_HOME>/<JAM_INSTALL_DIR>/config/verify/clientclasses/`
`verify/gateway/outbound` directory.

■ Run `egencobol` to use the eGen Application Code Generator on:

● `chardata.egen` to generate `Chardata.java`

● `baseClient.egen` to generate `BaseClient.java`.

This option will generate the source. To compile the source, use the previous option to run the `build.cmd (sh)`script.

For information about running the eGen Application Code Generator, see the *BEA WebLogic Java Adapter for Mainframe Programming Guide*.

Step 8: Complete Mainframe Tasks

On the machine with the IMS region:

1. Create a Partitioned Data Set (PDS) to store the source and JCL for the installation verification sample.

2. From the machine where the gateway is installed, use FTP to send the following files from the `<BEA_HOME>/<JAM_INSTALL_DIR>/samples/verify/IMS/outbound/source` directory to the PDS that you created:

   - `CHARDATA`

   - `COMPTOUP`

   - `JAMDEFI`

   - `TOUPIMS`

3. In the `COMPTOUP` JCL, make the following changes:

   - Change the `JOB` statement.

   - Change `YOURHLQ.SOURCE` to the PDS you created.

   - Change `YOUR.PGMLIB` to a `PGMLIB` for your IMS region.

   - Change `YOUR.PROCLIB` to the location of `CBLTDLI`.

   - Change `YOUR.RESLIB` to the proper value.

   - Change `RESLIB` to `SDFSRESL` if you are using a newer IMS version. Refer to `CBLTDLI` in your `PGMLIB` to determine which library name your version uses. Make sure, if your IMS version requires this change, that you make the change to both the `DD` name and the library name.

4. Submit the `COMPTOUP` JCL. Make sure that all the condition codes are zero.

5. Define the program `TOUPIMS` to the IMS region. `JAMDEFI` contains sample IMS stage 1 and `PSBGEN` input. See your IMS systems programmer for assistance.

Running the Sample

To run the installation verification sample for IMS, type the following command in the window used in step 6:

```
java verify.gateway.outbound.Client hello
```

This command will convert the string 'hello' to uppercase: HELLO.

If the client is being run on a different machine than the Gateway or if the WebLogic Server instance is listening on a port other than 7001, you may enter a URL as a second command line argument. For example:

```
java verify.gateway.outbound.Client hello "t3://hostname:port"
```

In this statement, the following definitions apply:

> hostname is the address of the machine where WebLogic Server is running.

> port is the port for WebLogic Server.

If you have successfully run the installation verification sample for IMS, your installation is complete and verified. To complete the post-installation tasks, see the "Performing Post-Installation Tasks" section.

Using the Installation Verification Sample for CICS

After completing the tasks in the "Before You Use the Samples" section, you are ready to set up and run the sample.

How the Sample Works

The program TOUPCICS is a simple COBOL CICS server program that is linked to and passed a COMMAREA. The characters in the COMMAREA are converted to uppercase. The structure of the data in the COMMAREA is given in the copybook CHARDATA. No special considerations are required in this program as a result of being used in an application with a Java client making requests through WebLogic JAM.

Understanding the Sample Configuration

For this simple sample, no special configuration is required. The Java client calls the service TOUPPER.

■ Gateway

The DPL service TOUPPER is mapped to the name of the CICS program TOUPCICS.

■ On the mainframe

The server program TOUPCICS is defined to the CICS region, in this case using RDO, in the usual way any program is defined to a CICS region.

Understanding the Sample Programming

The programming for this sample is described in the following sections.

WebLogic Application

Three classes compose the WebLogic side of this sample application:

■ Chardata

■ BaseClient

■ Client

Chardata is a DataView class that is generated by the eGen Application Code Generator. The data member in the Chardata class corresponds to the data field in the CHARDATA copybook. The Chardata class is responsible for all data translation between the mainframe format of the data and the Java format of the data.

The `BaseClient` class that is generated by the eGen Application Code Generator is an extension of the `EgenClient` class. The `BaseClient` is the class that implements the actual calls to the `TOUPPER` mainframe service in its `toupper` method.The `toupper` method is a wrapper for the `callService` method of the `EgenClient` class with the `TOUPPER` service as a parameter.

The `Client` class is the actual user interface. The `Client` class has a `BaseClient` member. The `Client` class receives string input as a command line argument. You may also enter a URL if the WebLogic JAM Gateway is running on a different machine or the corresponding instance of WebLogic Server is listening on a different port than `7001`. The URL is set in the `BaseClient` member. In the `Client` class a `Chardata` DataView is initialized with the input data. The `toupper` method of the `BaseClient` member is called with this DataView as a parameter. The converted string value encapsulated in the returned DataView displays.

CICS Program

The program `TOUPCICS` is a simple COBOL CICS server program that is linked to and passed a `COMMAREA`. The characters in the `COMMAREA` are upper-cased. The structure of the data in the `COMMAREA` is given in the copybook `CHARDATA`. No special considerations are required in this program as a result of being used in an application with a Java client making requests through WebLogic JAM.

Sample Files

The files for the WebLogic side of the sample are installed in the following directory:

`<BEA_HOME>/<JAM_INSTALL_DIR>/samples/verify/gateway/outbound`

The following table lists the sample files and their purpose:

Table 5-3 Files for the WebLogic JAM Application

File Name	File Purpose
chardata.cpy	COBOL copybook that defines the structure of the string mainframe data.
chardata.egen	eGen script that generates the `Chardata.java` DataView class.
Chardata.java	DataView class that corresponds to the `chardata.cpy` COBOL copybook.

File Name	File Purpose
baseClient.egen	eGen script that generates the `Chardata.java` DataView class and the `BaseClient.java` EgenClient class.
BaseClient.java	Java class that extends `EgenClient` class that calls the `TOUPPER` service.
Client.java	The user interface client class that receives the input string from the user, and displays the result of the `TOUPPER` service to the user. It invokes the `TOUPPER` service by calling the `callService` method of its `BaseClient` member.
build.cmd	Script that builds the `Chardata`, `BaseClient`, and `Client` classes. The built class files are under the `<BEA_HOME>\<JAM_INSTALL_DIR>\config\verify\clientclasses\verify\gateway\outbound` directory.
build.sh	Unix script that builds the `Chardata`, `BaseClient`, and `Client` classes. The built class files are under the `<BEA_HOME>/<JAM_INSTALL_DIR>/config/verify/clientclasses/verify/gateway/outbound` directory.

The files for the CICS side of the sample are installed in the following directory:

`<BEA_HOME>/<JAM_INSTALL_DIR>/samples/verify/CICS/outbound/source`

The following table lists the sample files and their purpose:

Table 5-4 Files for CICS Application

File Name	File Purpose
CHARDATA	COBOL copybook that defines the structure of the string input and output data.
CMPPROC	Procedure used to compile and link the CICS programs.
COMPILE	JCL that executes the `CMPROC` for the program `TOUPCICS`.
CSDU	RDO cards to define `TOUPCICS` to the CICS region.
CSDUPD	JCL that executes `CSDU`.

File Name	File Purpose
TOUPCICS	CICS server program that receives a string in mixed case and returns the string in uppercase.

Setting Up the Sample

To set up the installation verification sample for CICS, perform the following steps.

Step 1: Start the CRM

Before starting the WebLogic JAM Gateway, start the CRM. The CRM must be configured with certain parameter values at startup. These parameter values include:

- The address of the machine on which the CRM is running

- The port on which the CRM listens

- The name the Gateway will use to refer to the CRM

For running the samples, you must set the machine address and port. The values that you set for the machine address and port when the CRM is started, must agree with the values that you set for the CRM in the WebLogic Administration Console for the samples CRM. The name of the CRM that is preconfigured for running all of the samples is CRM1. Use this name when the CRM is started to run any of the samples.

The way you start the CRM depends on whether the CRM will be started under a Unix or MVS system. On Unix, start the CRM using a shell script. On MVS, start the CRM using JCL.

Starting the CRM on z/OS or OS/390 Unix

On z/OS or OS/390 Unix, you may use a script to start the CRM. Scripts are installed with the Gateway in the <BEA_HOME>/<JAM_INSTALL_DIR>/samples/crm/unix directory. The script, crm.env, appends the necessary values to your environment variables. The script, startcrm.sh, starts the CRM. To use these scripts, complete the following steps.

1. FTP the following two scripts to the directory from which the CRM will run:

   - `crm.env`

   - `startcrm.sh`

2. Edit `crm.env`. Supply the correct values for the APPDIR and CRMDIR variables. APPDIR is the directory from which the CRM will run. CRMDIR is the CRM installation directory.

3. Edit `startcrm.sh`. To use a different port than the default port, `7101`, change the port number. However, if you change the port number, make sure to change it in the corresponding field in the WebLogic Administration Console CRM1 pane. You do not need to change the address because the script will run on the machine where the CRM is installed.

   Note: BEA recommends that you do not change the CRM name from CRM1. This name for the CRM is preconfigured for all of the samples.

4. Execute the `startcrm.sh` script:

   ```
   . ./startcrm.sh
   ```

Compare Figure 5-2 with the script in Listing 5-3. Notice how the parameters in the script correspond to the fields in the WebLogic Administration Console. The script illustrates the values for `startcrm.sh` script parameters for running the samples.

Note: The port number is `7101`. You can change the port number; however, if the port number is changed make sure to change it in the corresponding field in the Gateway configuration CRM1 pane of the WebLogic Administration Console.

Listing 5-3   Command to Run the CRM

```
$CRMDIR/bin/CRM //127.0.0.1:7101 CRM1 < /dev/null > std.out
2>std.err &
```

Starting the CRM on z/OS or OS/390 MVS

On z/OS or OS/390 MVS, start the CRM by submitting the CRMSTART JCL that is installed with the CRM. The CRMSTART JCL must be modified for your environment. For information about modifying the CRMSTART JCL, see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*.

As you modify the CRMSTART JCL, make sure that you note the following parameters in the value of the STARTCMD parameter in the JCL. These parameters correspond to fields in the WebLogic Administration Console. These values must be the same in the JCL and in the WebLogic Administration Console.

- The machine address where the CRM will run

- The port number on which the CRM will listen

- The name by which the Gateway will refer to the CRM

Compare Figure 5-2 with the JCL in Listing 5-4. Notice how the parameters in the JCL correspond to the fields in the WebLogic Administration Console. The JCL illustrates the values for STARTCMD parameters for running the samples.

- The machine where the CRM will run is myhost in this sample. You must replace myhost with the hostname or IP address of your mainframe to allow IP-based communication from the Windows or Unix machine where WebLogic Server is running. You may verify this parameter with the ping command on Windows or Unix.

- The port number is 7101. You can change the port number; however, if you change the port number, make sure to change it in the corresponding field in the Gateway configuration CRM1 pane of the WebLogic Administration Console.

Note: BEA recommends that you do not change the CRM name from CRM1, because this name for the CRM is preconfigured for all of the samples.

Figure 5-2   Fields for the CRM
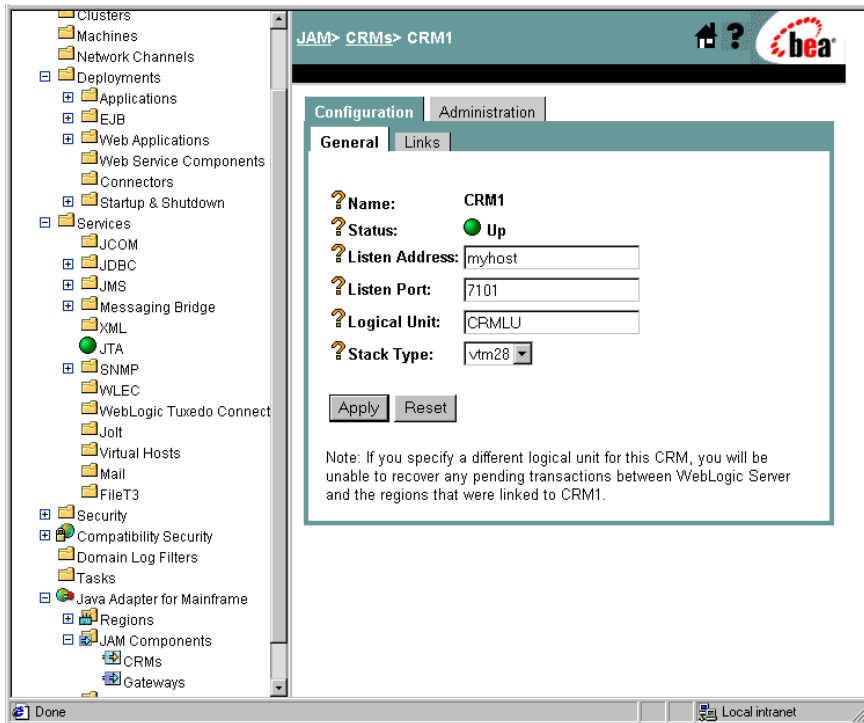
Listing 5-4   The STARTCMD parameter in the CRMSTART JCL

```
//   SET STARTCMD='"//myhost:7101" CRM1'
```

Step 2: Update the WebLogic JAM Configuration File

On the machine where the Gateway is located, update the WebLogic JAM configuration file from the command prompt by completing the following steps:

1. Locate the `jamconfig_CICS.xml` file in the following directory:

   `<BEA_HOME>/<JAM_INSTALL_DIR>/config/verify`

2. Copy `jamconfig_CICS.xml` to `jamconfig.xml`.

Step 3: Start the verify Domain

From the command prompt, execute the following command from the same directory to start the `verify` domain:

- For Microsoft Windows:

  `startVerifyServer.cmd`

- For Unix:

  `. ./startVerifyServer.sh`

Step 4: Configure the WebLogic JAM Gateway

Most configuration tasks were preconfigured or were completed during the installation process by the installer program. For additional information about configuring WebLogic JAM, see the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*. Make the following configuration changes for the installation verification sample for CICS to run on your system. These changes can be made in the WebLogic Administration Console in the following way.

1. From your browser, open the WebLogic Administration Console using the following address:

   `http://hostname:7001/console`

   In this address, the following definitions apply:

   `hostname` is the address of the machine where WebLogic Server is running.

   `7001` is the port for WebLogic Server that has been configured for the `verify` domain.

2. When prompted, supply the following user and password information:

   ● user: system

     This user name cannot be changed.

   ● password: security

     To change the password, see the BEA WebLogic Server documentation.

   The WebLogic Administration Console displays.

3. To configure the CRM to the WebLogic JAM Gateway, complete the following steps:

   a. In the left pane, click on **Java Adapter for Mainframe** → **JAM Components** → **CRMs**. In the right pane, click **CRM1**. On the **General** tab, set the following fields to correspond with your system. Click **Apply**. When the CRM is active, **Status** turns from red to green.

Field	Field Description
Listen Address	The address of the machine where the CRM is installed and running. This address must match the address set in the CRM startup JCL or script.
Listen Port	The port for the CRM. This entry must match the port set in the CRM startup JCL or script.
Logical Unit	The name of the Logical Unit defined for the CRM.
Stack Type	The stack type.

b. To configure the CICS region, click **Java Adapter for Mainframe → Regions → CICS Regions** in the left pane. In the right pane, click on **CICS3** and enter the name of the **Logical Unit**. Click **Apply** to set the Logical Unit.

Note: This name is the `ACBNAME` in the VTAM Logical Unit definition or the `APPLID` of the region.
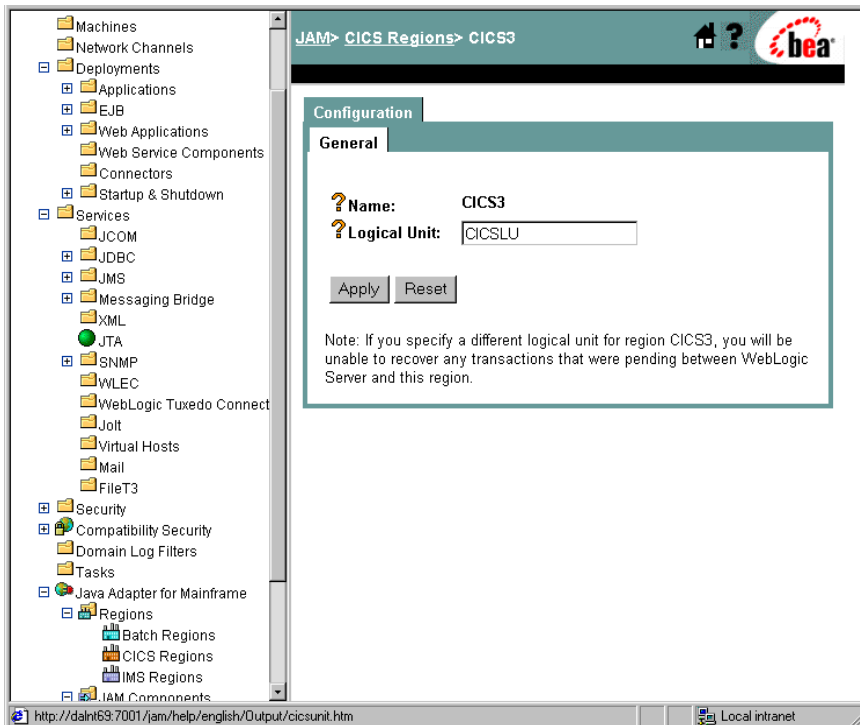
c. At the top of the pane, click **CICS Regions→CRM1CICS3**. In the new window, click **CRM1CICS3**. On the **Links** tab, check **Deployed** and click **Apply**.
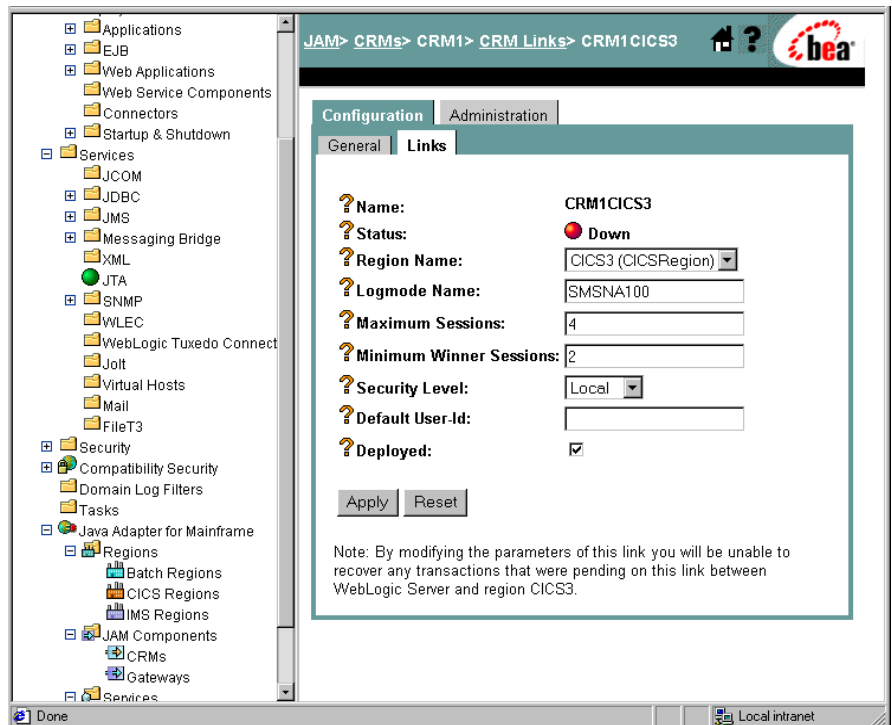
d. In the left pane, click **Gateways**. Click **JAM5.1** in the right pane. On the **General** tab, check **Deployed** and click **Apply**.
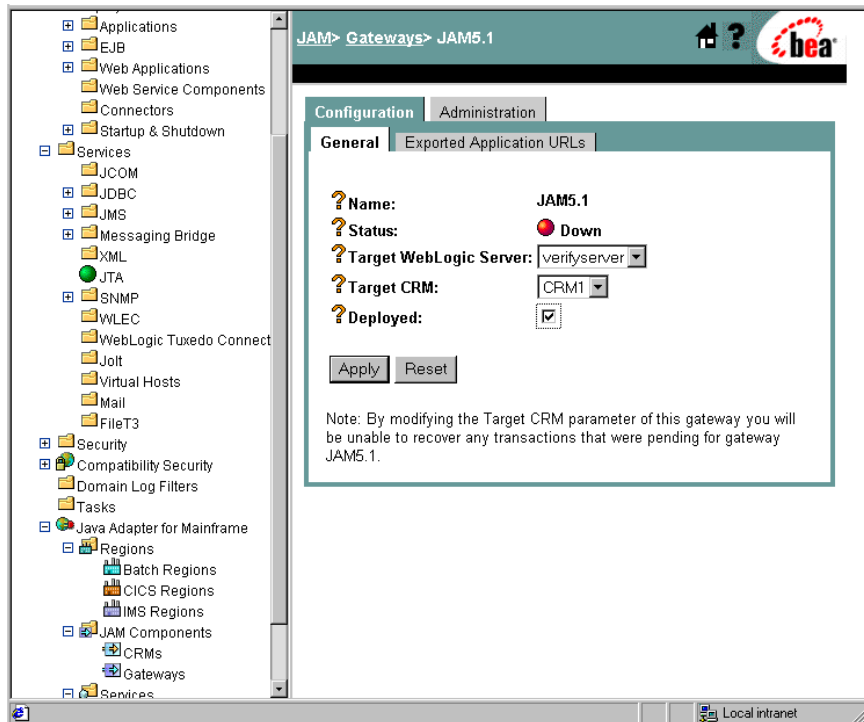
4. To start the WebLogic JAM Gateway, select the **Administration** tab →
   Start/Stop tab. Click **Start** to start the Gateway.
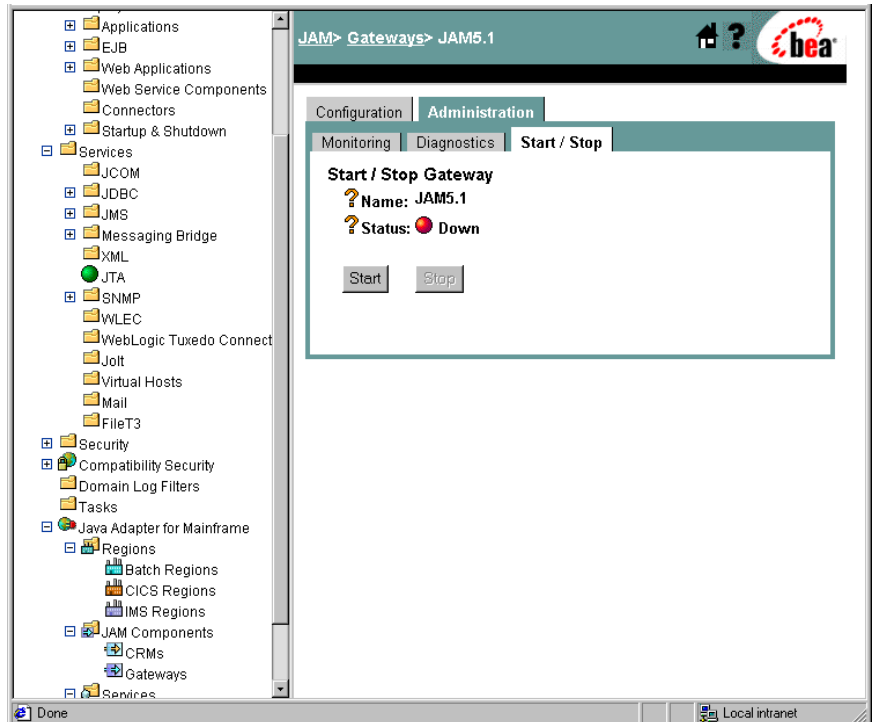
If the Gateway is running, **Status** changes to green in the WebLogic
Administration Console and the following message appears in the WebLogic
Server log:

"JAM Gateway ready for use. Current link status: up(1)."

Step 5: Enable the Service

To enable the DPL service, click **Java Adapter for Mainframe** → **Services**→**DPL Services** in the left pane. In the right pane, click **TOUPPER** under Service Name. Check **Enabled** and click **Apply**.
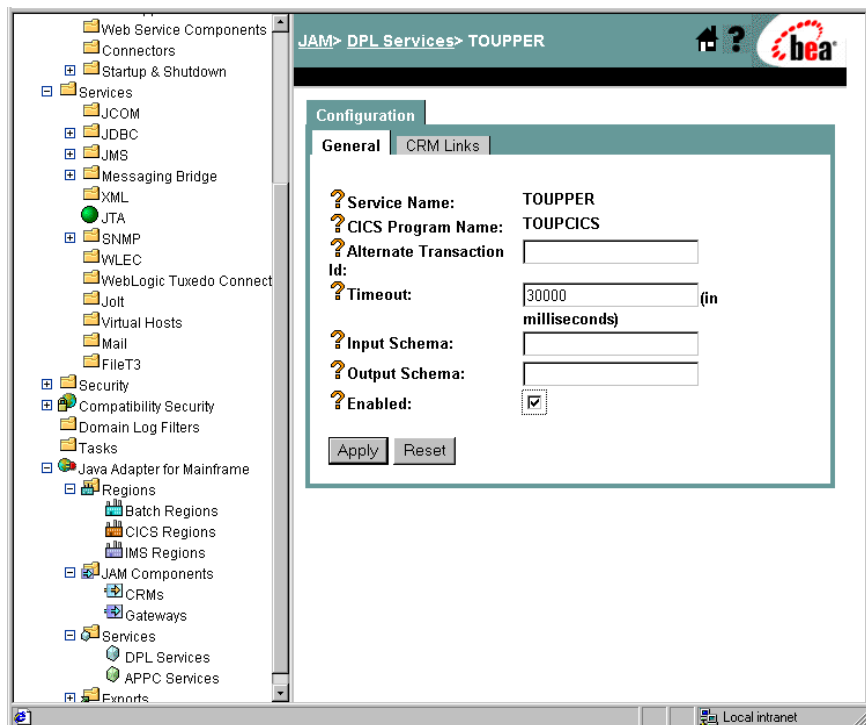
Step 6: Set the Environment

On the machine where the Gateway is installed, set the environment by completing the following step:

- From a command prompt, change to the `<BEA_HOME>/<JAM_INSTALL_DIR>/config/verify` and execute the following command to set the environment:

For Microsoft Windows:

```
setVerifyEnv.cmd
```

For Unix:

```
. ./setVerifyEnv.sh
```

The following message will display:

"Your environment has been set."

Step 7: Generate and Build Source (Optional)

The WebLogic JAM samples provide generated source. The samples also provide classes to run the samples. If you want to see how the source is generated and the classes are built, change to the
`<BEA_HOME>/<JAM_INSTALL_DIR>/samples/verify/gateway/outbound`
directory and complete the following steps.

Warning: Using the following options will overwrite files that are installed with the WebLogic JAM samples.

■   Run the `build.cmd (sh)` script to build the client classes. The build script puts the `built.class` files in the
`<BEA_HOME>/<JAM_INSTALL_DIR>/config/verify/clientclasses/verify/gateway/outbound` directory.

■   Run `egencobol` to use the eGen Application Code Generator on:

●   `chardata.egen` to generate `Chardata.java`

●   `baseClient.egen` to generate `BaseClient.java`.

This option will generate the source. To compile the source, use the previous option to run the `build.cmd (sh)`script.

For information about running the eGen Application Code Generator, see the *BEA WebLogic Java Adapter for Mainframe Programming Guide*.

Step 8: Complete Mainframe Tasks

On the machine with the CICS region:

1. Create a Partitioned Data Set (PDS) to store the source and JCL for the installation verification sample for CICS.

2. From the machine where the Gateway was installed, use FTP to send the following files from the `<BEA_HOME>/<JAM_INSTALL_DIR>/samples/verify/CICS/outbound/sourc e` directory to the PDS that you created:

   - CHARDATA

   - CMPPROC

   - COMPILE

   - CSDU

   - CSDUPD

   - TOUPCICS

3. In the procedure, CMPPROC, you do not need to set LNKLIB, PDSSRC, and PROC. The values are supplied by the COMPILE JCL that will exec COMPROC. Set INDEX, COMPHLQ, COMPHL2. You may need to change OUTC and the unit of WORK.

4. In the COMPILE JCL, make the following changes:

   - Change the JOB statement.

   - Change the LNKLIB to a load lib for your CICS region.

   - Change the YOURHLQ.JCLLIB.ORDER to the PDS name containing CMPPROC.

5. Submit the COMPILE JCL. Make sure that the condition code is zero.

6. In the RDO script CSDU, make the following changes:

   - Change the name of the LIST of a valid list name for your CICS region.

   - You can also change the GROUP name, but make sure it is changed everywhere.

7. In the CSDUPD JCL, make the following changes:

   - Change the STEPLIB, DFHCSD, and SYSIN DSNs.

   - You may need to change the unit of WORK.

8. Submit the CSDUPD JCL. Note that you may get a warning on the Delete step in the CSDU, because the TOUPCICS program probably was not defined before. The condition code should not be above 4.

9. Log on to your CICS region.

10. To install the WebLogic JAM installation verification GROUP:

    ```
    CEDA INSTALL GROUP(JAMVRFY)
    ```

11. To verify the CICS TOUPCICS program:

    ```
    CEMT INQUIRE PROG(TOUPCICS)
    ```

Running the Sample

To run the installation verification sample for CICS, type the following in the command prompt in the window you used to set the environment:

```
java verify.gateway.outbound.Client hello
```

This command will change the string, 'hello', to uppercase: HELLO.

If the client is being run on a different machine than the Gateway or if the WebLogic Server instance is listening on a port other than 7001, you may enter a URL as a second command line argument. For example:

```
java verify.gateway.outbound.Client hello "t3://hostname:port"
```

In this statement, the following definitions apply:

hostname is the address of the machine where WebLogic Server is running.

port is the port for WebLogic Server.

If you have successfully run the installation verification sample for CICS, your installation is complete and verified. To complete the post-installation tasks, see the "Performing Post-Installation Tasks" section.

6 Uninstalling BEA WebLogic Java Adapter for Mainframe

The following sections describe the process for uninstalling BEA WebLogic Java Adapter for Mainframe (WebLogic JAM). The WebLogic JAM product may be removed or uninstalled in the following ways, however, the CRM on the mainframe is not uninstalled:

- Uninstalling WebLogic JAM Using Graphical-Mode
  - On Microsoft Windows
  - On Unix
- Uninstalling WebLogic JAM Using Console-Mode

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

Uninstalling WebLogic JAM Using Graphical-Mode

The following sections provide instructions for uninstalling the WebLogic JAM product in graphical mode based on your platform.

Prepare to Uninstall

Prepare to uninstall WebLogic JAM on the platform that corresponds to your system by completing the following steps.

On Microsoft Windows

To prepare to uninstall WebLogic JAM on Microsoft Windows:

1. Choose **Start→Programs→BEA WebLogic E-Business Platform→ WebLogic Java Adapter for Mainframe 5.1→Uninstall Java Adapter for Mainframe**.

   The screen in Figure 6-1 displays. Continue to uninstall WebLogic JAM by completing the steps in the "Uninstall WebLogic JAM" section.

On Unix

To prepare to uninstall WebLogic JAM on Unix using graphical mode:

1. Change directories to your uninstaller directory. The following command is an example.

   ```
   >cd /usr/bea/wljam5.1/uninstaller
   ```

2. Start the uninstallation program. The following command is an example:

   ```
   /usr/bea/wljam5.1/uninstaller> sh Uninstall
   ```

The screen in Figure 6-1 displays. Continue to uninstall WebLogic JAM by completing the steps in the "Uninstall WebLogic JAM" section.

Figure 6-1   Uninstalling WebLogic JAM
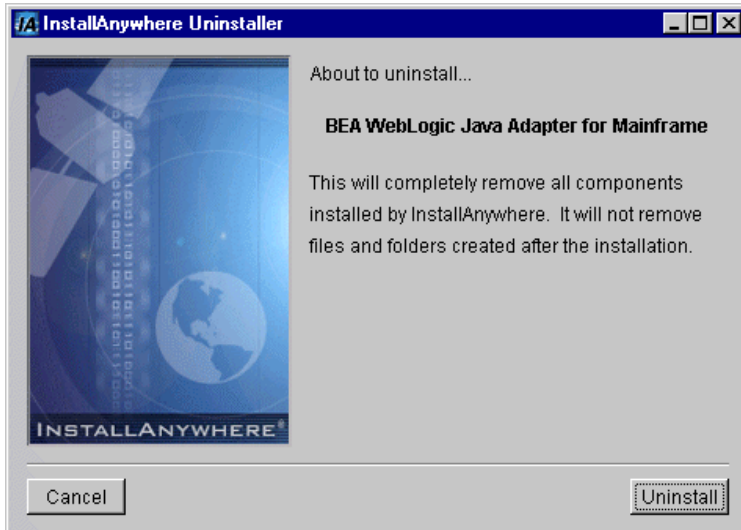
Uninstall WebLogic JAM

1. Click **Uninstall** on the screen in Figure 6-1 to begin the uninstall process. The following screen displays.
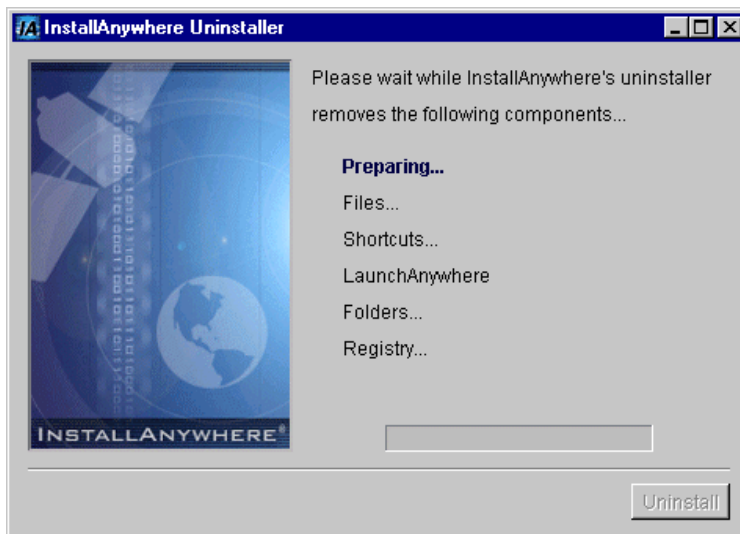
2. When the following screen displays, the uninstall is complete. Click **Exit** to close the uninstall program.
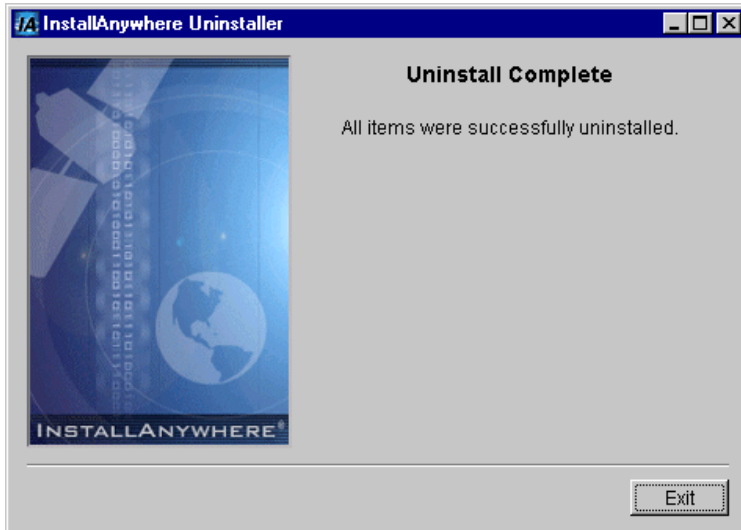
The uninstall process removes all effects of the installation, including:

- Removing files

- Resetting environment variables

- Removing registry entries

- Uninstalling the WebLogic JAM service on Microsoft Windows

Uninstalling WebLogic JAM Using Console-Mode

To uninstall WebLogic JAM on Unix in console mode, perform the following steps:

1. Change directories to the `uninstaller` directory with command such as the following example.

   ```
   > cd /usr/bea/wljam5.1/uninstaller
   ```

2. Start the uninstallation program. The following command is an example:

   ```
   /usr/bea/wljam5.1/uninstaller> sh Uninstall -i console
   ```

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

Listing 6-1 displays an example of a WebLogic JAM uninstallation.

Note: The single-dash bar just beneath the double-dash bar in Listing 6-1 is a progress indicator. When the single-dash bar becomes the same length as the double-dash bar, uninstallation is complete.

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

Listing 6-1   Console Mode Uninstallation Example

```
Preparing CONSOLE Uninstall...

================================================================

BEA WebLogic Java Adapter for Mainframe (InstallAnywhere
Uninstaller by Zero G)

----------------------------------------------------------------

About to uninstall...

   BEA WebLogic Java Adapter for Mainframe
```

```
This will completely remove all components installed by
InstallAnywhere. It will not remove files and folders created after
the installation.

PRESS <ENTER> TO BEGIN THE UNINSTALL, OR 'QUIT' TO EXIT THE
UNINSTALLER: <ENTER>

================================================================

Uninstall...
------------

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

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

================================================================

Uninstall Complete

------------------

All items were successfully uninstalled.

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

A BEA WebLogic Java Adapter for Mainframe Installation Files and Directories

The following files and directories are installed with BEA WebLogic Java Adapter for Mainframe (WebLogic JAM).

Communications Resource Manager (CRM) Installation Files

The following Communications Resource Manager (CRM) files are installed on your system.

z/OS and OS/390 Unix Platforms

The following CRM application files are installed by the WebLogic JAM software for the z/OS and OS/390 Unix platforms. All directories are relative to the specified installation directory. For specific platform and version information, see the *BEA WebLogic Java Adapter for Mainframe Release Notes.*

Table A-1 z/OS and OS/390 Unix CRM Installation Files and Directories

Directory	Files
bin	crmlkoff crmlkon crmdown CRMLOGS CRM unix.env
lib	ctxdebugs_12.dll ctxos_12.dll ctxprim_12.dll ctxplat_12.dll csxappc.dll csxxfm.dll csxgpw.dll csxcrm.dll csxscrm.dll csxvtm28.dll libgp.dll libgp40.dll libgpnet.dll

z/OS and OS/390 MVS Platforms

The following CRM application files are installed by the WebLogic JAM software for the z/OS and OS/390 MVS platforms. All directories are relative to the specified installation directory. For specific platform and version information, see the *BEA WebLogic Java Adapter for Mainframe Release Notes.*

Table A-2 z/OS and OS/390 MVS CRM Installation Files and Directories

Directory	Files
load library (PDSE)	CRMLOGS CRM CRMLKON CRMLKOFF CRMDOWN CSXAPPC CSXXFM CSXGPW CSXCRM CSXSCRM CSXVTM28 DEBUGS OS PRIM PLAT LIBGPNET LIBGP LIBGP40
data library (PDS)	ENV
sample (PDS)	CRMSTART CRMLKON CRMLKOFF CRMDOWN CRMLOGS

WebLogic JAM Gateway Installation

The following files and directories are installed by the WebLogic JAM software. All directories are relative to the specified installation directory. For specific platform and version information, see the *BEA WebLogic Java Adapter for Mainframe Release Notes.*

Table A-3 WebLogic JAM Gateway Installation Files and Directories

Directory	Files
`bin`	`egencobol`
`classdocs`	`jamdoc.jar`

`config`	`examples`
	`verify`
	Refer to the platform that corresponds to your system in the Samples Files section for a complete listing of files for these directories.

`lib`	`jam.jar` `jam.ear`
`samples`	Refer to the platform that corresponds to your system in the Samples Files section.

`uninstaller`	`complist`
	`install.log`
	`lax.jar`
	`Uninstall`
	`Uninstall.lax`
	`uninstaller.jar`
	`user.zip`
	`resource/remove.sh`
	`resource/il8nresources/custom_en`

Samples Files

The following sections list the samples files that are installed by the WebLogic JAM software. See the section that corresponds to your platform.

Samples Files for Microsoft Windows Platforms

The following samples files are installed for Microsoft Windows platforms.

Directory	Files
config\examples	ca.pem
	config.xml
	democert.pem
	demokey.pem
	fileRealm.properties
	jamconfig_BATCH.xml
	jamconfig_CICS.xml
	jamconfig_IMS.xml
	SerializedSystemIni.dat
	startExamplesEnv.cmd
	startExamplesServer.cmd
	applications\jam.ear
	applications\JAM_ToupperServer.jar
	applications\JAM_TradeServer.jar
	applications\DefaultWebApp\Web-inf\web.xml
	clientclasses\dataviews\examples\IMS\inbound\gateway\ Chardata.class

Directory	Files
	clientclasses\examples\CICS\outbound\gateway\ BaseClient.class
	clientclasses\examples\CICS\outbound\gateway\ Client.class
	clientclasses\examples\CICS\outbound\gateway\ EmployeeRecord$EmpRecord1V$EmpAddr7V.class
	clientclasses\examples\CICS\outbound\gateway\ EmployeeRecord$EmpRecord1V$EmpName3V.class
	clientclasses\examples\CICS\outbound\gateway\ EmployeeRecord$EmpRecord1V.class
	clientclasses\examples\CICS\outbound\gateway\ EmployeeRecord.class
	clientclasses\examples\IMS\inbound\gateway\ TopicReceive.class
	clientclasses\examples\IMS\outbound\gateway\ BaseClient.class
	clientclasses\examples\IMS\outbound\gateway\Client.class
	clientclasses\examples\IMS\outbound\gateway\ IvtnoInRecord$IvtnoInRecord1V.class
	clientclasses\examples\IMS\outbound\gateway\ IvtnoInRecord.class
	clientclasses\examples\IMS\outbound\gateway\ IvtnoOutRecord$IvtnoOutRecord1V.class
	clientclasses\examples\IMS\outbound\gateway\ IvtnoOutRecord.class

Directory	Files
	`clientclasses\examples\transactional\CICS\outbound\gateway\` `BaseClient.class`
	`clientclasses\examples\transactional\CICS\outbound\gateway\` `Client.class`
	`clientclasses\examples\transactional\CICS\outbound\gateway\` `EmployeeRecord$EmpRecord1V$EmpAddr7V.class`
	`clientclasses\examples\transactional\CICS\outbound\gateway\` `EmployeeRecord$EmpRecord1V$EmpName3V.class`
	`clientclasses\examples\transactional\CICS\outbound\gateway\` `EmployeeRecord$EmpRecord1V.class`
	`clientclasses\examples\transactional\CICS\outbound\gateway\` `EmployeeRecord.class`
	`clientclasses\examples\transactional\IMS\outbound\gateway\` `BaseClient.class`
	`clientclasses\examples\transactional\IMS\outbound\gateway\` `Client.class`
	`clientclasses\examples\transactional\IMS\outbound\gateway\` `IvtnoInRecord$IvtnoInRecord1V.class`
	`clientclasses\examples\transactional\IMS\outbound\gateway\` `IvtnoInRecord.class`
	`clientclasses\examples\transactional\IMS\outbound\gateway\` `IvtnoOutRecord$IvtnoOutRecord1V.class`
	`clientclasses\examples\transactional\IMS\outbound\gateway\` `IvtnoOutRecord.class`

`config\verify`	`ca.pem`
	`config.xml`
	`democert.pem`
	`demokey.pem`
	`fileRealm.properties`
	`jamconfig_CICS.xml`
	`jamconfig_IMS.xml`
	`SerializedSystemIni.dat`
	`setVerifyEnv.cmd`
	`startVerifyServer.cmd`

	`applications\jam.ear`

Directory	Files
	`applications\DefaultWebApp\Web-inf\web.xml`
	`clientclasses\verify\gateway\outbound\baseClient.class`
	`clientclasses\verify\gateway\outbound\Chardata.class`
	`clientclasses\verify\gateway\outbound\Client.class`
`samples\crm\`	`unix\crm.env`
	`unix\startcrm.sh`
`samples\examples\`	`CICS\inbound\gateway\build.cmd`
	`CICS\inbound\gateway\ejb-jar.xml`
	`CICS\inbound\gateway\ExtTradeServerBean.java`
	`CICS\inbound\gateway\TradeRecord.cpy`
	`CICS\inbound\gateway\TradeRecord.java`
	`CICS\inbound\gateway\tradeserver.egen`
	`CICS\inbound\gateway\TradeServer.java`
	`CICS\inbound\gateway\TradeServerBean.java`
	`CICS\inbound\gateway\TradeServerHome.java`
	`CICS\inbound\gateway\TradeServer-jar.xml`
	`CICS\inbound\gateway\weblogic-ejb-jar.xml`
	`CICS\inbound\gateway\wl-TradeServer-jar.xml`
	`CICS\inbound\gateway\build\examples\CICS\inbound\gateway\` `ExtTradeServerBean.class`
	`CICS\inbound\gateway\build\examples\CICS\inbound\gateway\` `TradeRecord$TradeRecord1V.class`
	`CICS\inbound\gateway\build\examples\CICS\inbound\gateway\` `TradeRecord.class`
	`CICS\inbound\gateway\build\examples\CICS\inbound\gateway\` `TradeServer.class`
	`CICS\inbound\gateway\build\examples\CICS\inbound\gateway\` `TradeServerBean.class`
	`CICS\inbound\gateway\build\examples\CICS\inbound\gateway\` `TradeServerHome.class`
	`CICS\inbound\gateway\build\META-INF\ejb-jar.xml`
	`CICS\inbound\gateway\build\META-INF\weblogic-ejb-jar.xml`

Directory	Files
	`CICS\inbound\mainframe\source\CMPPROC`
	`CICS\inbound\mainframe\source\COMPTRCL`
	`CICS\inbound\mainframe\source\CSDUPDTR`
	`CICS\inbound\mainframe\source\CSDUTRCL`
	`CICS\inbound\mainframe\source\TRADCLNT`
	`CICS\inbound\mainframe\source\TRADRCRD`
	`CICS\outbound\gateway\baseClient.egen`
	`CICS\outbound\gateway\BaseClient.java`
	`CICS\outbound\gateway\build.cmd`
	`CICS\outbound\gateway\Client.java`
	`CICS\outbound\gateway\EmployeeRecord.java`
	`CICS\outbound\gateway\emprec.cpy`
	`CICS\outbound\gateway\emprec.egen`
	`CICS\outbound\mainframe\source\CMPPROC`
	`CICS\outbound\mainframe\source\COMPCRUD`
	`CICS\outbound\mainframe\source\CSDUCRUD`
	`CICS\outbound\mainframe\source\CSDUPDCO`
	`CICS\outbound\mainframe\source\DPLDEMOC`
	`CICS\outbound\mainframe\source\DPLDEMOD`
	`CICS\outbound\mainframe\source\DPLDEMOR`
	`CICS\outbound\mainframe\source\DPLDEMOU`
	`CICS\outbound\mainframe\source\EMPREC`

Directory	Files
	`explicitAPPC\gateway\build.cmd`
	`explicitAPPC\gateway\chardata.cpy`
	`explicitAPPC\gateway\chardata.egen`
	`explicitAPPC\gateway\Chardata.java`
	`explicitAPPC\gateway\ejb-jar.xml`
	`explicitAPPC\gateway\ExtToupperServerBean.java`
	`explicitAPPC\gateway\toupperServer.egen`
	`explicitAPPC\gateway\ToupperServer.java`
	`explicitAPPC\gateway\ToupperServerBean.java`
	`explicitAPPC\gateway\ToupperServerHome.java`
	`explicitAPPC\gateway\ToupperServer-jar.xml`
	`explicitAPPC\gateway\weblogic-ejb-jar.xml`
	`explicitAPPC\gateway\wl-ToupperServer-jar.xml`
	`explicitAPPC\gateway\build\examples\explicitAPPC\gateway\` `Chardata.class`
	`explicitAPPC\gateway\build\examples\explicitAPPC\gateway\` `ExtToupperServerBean.class`
	`explicitAPPC\gateway\build\examples\explicitAPPC\gateway\` `ToupperServer.class`
	`explicitAPPC\gateway\build\examples\explicitAPPC\gateway\` `ToupperServerBean.class`
	`explicitAPPC\gateway\build\examples\explicitAPPC\gateway\` `ToupperServerHome.class`
	`explicitAPPC\gateway\build\META-INF\ejb-jar.xml`
	`explicitAPPC\gateway\build\META-INF\weblogic-ejb-jar.xml`
	`explicitAPPC\mainframe\source\ASEJBAPI`
	`explicitAPPC\mainframe\source\CLCLIENT`
	`explicitAPPC\mainframe\source\EJBAPI`
	`explicitAPPC\mainframe\source\EXCLIENT`
	`explicitAPPC\mainframe\source\WLCLIENT`
	`explicitAPPC\mainframe\source\VTAMDEFINITION`

Directory	Files
	`IMS\inbound\gateway\build.cmd`
	`IMS\inbound\gateway\chardata.cpy`
	`IMS\inbound\gateway\chardata.egen`
	`IMS\inbound\gateway\Chardata.java`
	`IMS\inbound\gateway\TopicReceive.java`
	`IMS\inbound\mainframe\source\COMPIMSC`
	`IMS\inbound\mainframe\source\DFS62DTI`
	`IMS\inbound\mainframe\source\IMSINDEF`
	`IMS\inbound\mainframe\source\IMSTOJMS`
	`IMS\outbound\gateway\baseClient.egen`
	`IMS\outbound\gateway\BaseClient.java`
	`IMS\outbound\gateway\build.cmd`
	`IMS\outbound\gateway\Client.java`
	`IMS\outbound\gateway\ivtno.egen`
	`IMS\outbound\gateway\ivtno-in.cpy`
	`IMS\outbound\gateway\IvtnoInRecord.java`
	`IMS\outbound\gateway\ivtno-out.cpy`
	`IMS\outbound\gateway\IvtnoOutRecord.java`
	`transactional\CICS\outbound\gateway\baseClient.egen`
	`transactional\CICS\outbound\gateway\BaseClient.java`
	`transactional\CICS\outbound\gateway\build.cmd`
	`transactional\CICS\outbound\gateway\Client.java`
	`transactional\CICS\outbound\gateway\EmployeeRecord.java`
	`transactional\CICS\outbound\gateway\emprec.cpy`
	`transactional\CICS\outbound\gateway\emprec.egen`

Directory	Files
	`transactional\CICS\outbound\mainframe\source\BLDVSAM`
	`transactional\CICS\outbound\mainframe\source\CMPPROC`
	`transactional\CICS\outbound\mainframe\source\COMPILEV`
	`transactional\CICS\outbound\mainframe\source\CSDUPDCT`
	`transactional\CICS\outbound\mainframe\source\DPLDEMVC`
	`transactional\CICS\outbound\mainframe\source\DPLDEMVD`
	`transactional\CICS\outbound\mainframe\source\DPLDEMVR`
	`transactional\CICS\outbound\mainframe\source\DPLDEMVU`
	`transactional\CICS\outbound\mainframe\source\EMPREC`
	`transactional\CICS\outbound\mainframe\source\JAMVSAMC`
	`transactional\CICS\outbound\mainframe\source\JVSAMRDO`
	`transactional\IMS\outbound\gateway\baseClient.egen`
	`transactional\IMS\outbound\gateway\BaseClient.java`
	`transactional\IMS\outbound\gateway\build.cmd`
	`transactional\IMS\outbound\gateway\Client.java`
	`transactional\IMS\outbound\gateway\ivtno.egen`
	`transactional\IMS\outbound\gateway\ivtno-in.cpy`
	`transactional\IMS\outbound\gateway\IvtnoInRecord.java`
	`transactional\IMS\outbound\gateway\ivtno-out.cpy`
	`transactional\IMS\outbound\gateway\IvtnoOutRecord.java`
	`wlintegration\create.xml`
	`wlintegration\read.xml`
	`wlintegration\readString.xml`
`samples\verify`	`CICS\outbound\source\CHARDATA`
	`CICS\outbound\source\CMPPROC`
	`CICS\outbound\source\COMPILE`
	`CICS\outbound\source\CSDU`
	`CICS\outbound\source\CSDUPD`
	`CICS\outbound\source\TOUPCICS`

Directory	Files
	`gateway\outbound\baseClient.egen`
	`gateway\outbound\baseClient.java`
	`gateway\outbound\build.cmd`
	`gateway\outbound\chardata.cpy`
	`gateway\outbound\chardata.egen`
	`gateway\outbound\Chardata.java`
	`gateway\outbound\Client.java`
	`IMS\outbound\source\CHARDATA`
	`IMS\outbound\source\COMPTOUP`
	`IMS\outbound\source\JAMDEFI`
	`IMS\outbound\source\TOUPIMS`

Samples Files for Unix Platforms

The following files are installed for Unix platforms.

Directory	Files
`config/examples/`	`ca.pem`
	`config.xml`
	`democert.pem`
	`demokey.pem`
	`fileRealm.properties`
	`jamconfig_BATCH.xml`
	`jamconfig_CICS.xml`
	`jamconfig_IMS.xml`
	`SerializedSystemIni.dat`
	`setExamplesEnv.sh`
	`startExamplesServer.sh`
	`applications/jam.ear`
	`applications/JAM_ToupperServer.jar`
	`applications/JAM_TradeServer.jar`

Directory	Files
	`applications/DefaultWebApp/Web-inf/web.xml`
	`clientclasses/dataviews/examples/IMS/inbound/gateway/` `Chardata.class`
	`clientclasses/examples/CICS/outbound/gateway/` `BaseClient.class`
	`clientclasses/examples/CICS/outbound/gateway/Client.class`
	`clientclasses/examples/CICS/outbound/gateway/` `EmployeeRecord$EmpRecord1V$EmpAddr7V.class`
	`clientclasses/examples/CICS/outbound/gateway/` `EmployeeRecord$EmpRecord1V$EmpName3V.class`
	`clientclasses/examples/CICS/outbound/gateway/` `EmployeeRecord$EmpRecord1V.class`
	`clientclasses/examples/CICS/outbound/gateway/` `EmployeeRecord.class`
	`clientclasses/examples/IMS/inbound/gateway/` `TopicReceive.class`
	`clientclasses/examples/IMS/outbound/gateway/BaseClient.class`
	`clientclasses/examples/IMS/outbound/gateway/Client.class`
	`clientclasses/examples/IMS/outbound/gateway/` `IvtnoInRecord$IvtnoInRecord1V.class`
	`clientclasses/examples/IMS/outbound/gateway/` `IvtnoInRecord.class`
	`clientclasses/examples/IMS/outbound/gateway/` `IvtnoOutRecord$IvtnoOutRecord1V.class`
	`clientclasses/examples/IMS/outbound/gateway/` `IvtnoOutRecord.class`

Directory	Files
	clientclasses/examples/transactional/CICS/outbound/gateway/ BaseClient.class
	clientclasses/examples/transactional/CICS/outbound/gateway/ Client.class
	clientclasses/examples/transactional/CICS/outbound/gateway/ EmployeeRecord$EmpRecord1V$EmpAddr7V.class
	clientclasses/examples/transactional/CICS/outbound/gateway/ EmployeeRecord$EmpRecord1V$EmpName3V.class
	clientclasses/examples/transactional/CICS/outbound/gateway/ EmployeeRecord$EmpRecord1V.class
	clientclasses/examples/transactional/CICS/outbound/gateway/ EmployeeRecord.class
	clientclasses/examples/transactional/IMS/outbound/gateway/ BaseClient.class
	clientclasses/examples/transactional/IMS/outbound/gateway/ Client.class
	clientclasses/examples/transactional/IMS/outbound/gateway/ IvtnoInRecord$IvtnoInRecord1V.class
	clientclasses/examples/transactional/IMS/outbound/gateway/ IvtnoInRecord.class
	clientclasses/examples/transactional/IMS/outbound/gateway/ IvtnoOutRecord$IvtnoOutRecord1V.class
	clientclasses/examples/transactional/IMS/outbound/gateway/ IvtnoOutRecord.class
config/verify	ca.pem config.xml democert.pem demokey.pem fileRealm.properties jamconfig_CICS.xml jamconfig_IMS.xml SerializedSystemIni.dat setVerifyEnv.sh startVerifyServer.sh
	applications/jam.ear

Directory	Files
	`applications/DefaultWebApp/Web-inf/web.xml`
	`clientclasses/verify/gateway/outbound/BaseClient.class`
	`clientclasses/verify/gateway/outbound/Chardata.class`
	`clientclasses/verify/gateway/outbound/Client.class`
`samples/crm/`	`unix/crm.env`
	`unix/startcrm.sh`
`samples/examples/`	`CICS/inbound/gateway/build.sh`
	`CICS/inbound/gateway/ejb-jar.xml`
	`CICS/inbound/gateway/ExtTradeServerBean.java`
	`CICS/inbound/gateway/TradeRecord.cpy`
	`CICS/inbound/gateway/TradeRecord.java`
	`CICS/inbound/gateway/TradeServer-jar.xml`
	`CICS/inbound/gateway/tradeserver.egen`
	`CICS/inbound/gateway/TradeServer.java`
	`CICS/inbound/gateway/TradeServerBean.java`
	`CICS/inbound/gateway/TradeServerHome.java`
	`CICS/inbound/gateway/weblogic-ejb-jar.xml`
	`CICS/inbound/gateway/wl-TradeServer-jar.xml`
	`CICS/inbound/gateway/build/examples/CICS/inbound/gateway/` `ExtTradeServerBean.class`
	`CICS/inbound/gateway/build/examples/CICS/inbound/gateway/` `TradeRecord$TradeRecord1V.class`
	`CICS/inbound/gateway/build/examples/CICS/inbound/gateway/` `TradeRecord.class`
	`CICS/inbound/gateway/build/examples/CICS/inbound/gateway/` `TradeServer.class`
	`CICS/inbound/gateway/build/examples/CICS/inbound/gateway/` `TradeServerBean.class`
	`CICS/inbound/gateway/build/examples/CICS/inbound/gateway/` `TradeServerHome.class`
	`CICS/inbound/gateway/build/META-INF/ejb-jar.xml`
	`CICS/inbound/gateway/build/META-INF/weblogic-ejb-jar.xml`

Directory	Files
	`CICS/inbound/mainframe/source/CMPPROC`
	`CICS/inbound/mainframe/source/COMPTRCL`
	`CICS/inbound/mainframe/source/CSDUPDTR`
	`CICS/inbound/mainframe/source/CSDUTRCL`
	`CICS/inbound/mainframe/source/TRADCLNT`
	`CICS/inbound/mainframe/source/TRADRCRD`
	`CICS/outbound/gateway/baseClient.egen`
	`CICS/outbound/gateway/BaseClient.java`
	`CICS/outbound/gateway/build.sh`
	`CICS/outbound/gateway/Client.java`
	`CICS/outbound/gateway/EmployeeRecord.java`
	`CICS/outbound/gateway/emprec.cpy`
	`CICS/outbound/gateway/emprec.egen`
	`CICS/outbound/mainframe/source/CMPPROC`
	`CICS/outbound/mainframe/source/COMPCRUD`
	`CICS/outbound/mainframe/source/CSDUCRUD`
	`CICS/outbound/mainframe/source/CSDUPDCO`
	`CICS/outbound/mainframe/source/DPLDEMOC`
	`CICS/outbound/mainframe/source/DPLDEMOD`
	`CICS/outbound/mainframe/source/DPLDEMOR`
	`CICS/outbound/mainframe/source/DPLDEMOU`
	`CICS/outbound/mainframe/source/EMPREC`

Directory	Files
	`explicitAPPC/gateway/build.sh`
	`explicitAPPC/gateway/chardata.cpy`
	`explicitAPPC/gateway/chardata.egen`
	`explicitAPPC/gateway/Chardata.java`
	`explicitAPPC/gateway/ejb-jar.xml`
	`explicitAPPC/gateway/ExtToupperServerBean.java`
	`explicitAPPC/gateway/ToupperServer-jar.xml`
	`explicitAPPC/gateway/toupperServer.egen`
	`explicitAPPC/gateway/ToupperServer.java`
	`explicitAPPC/gateway/ToupperServerBean.java`
	`explicitAPPC/gateway/ToupperServerHome.java`
	`explicitAPPC/gateway/weblogic-ejb-jar.xml`
	`explicitAPPC/gateway/wl-ToupperServer-jar.xml`
	`explicitAPPC/gateway/build/examples/explicitAPPC/gateway/Chardata.class`
	`explicitAPPC/gateway/build/examples/explicitAPPC/gateway/ExtToupperServerBean.class`
	`explicitAPPC/gateway/build/examples/explicitAPPC/gateway/ToupperServer.class`
	`explicitAPPC/gateway/build/examples/explicitAPPC/gateway/ToupperServerBean.class`
	`explicitAPPC/gateway/build/examples/explicitAPPC/gateway/ToupperServerHome.class`
	`explicitAPPC/gateway/build/META-INF/ejb-jar.xml`
	`explicitAPPC/gateway/build/META-INF/weblogic-ejb-jar.xml`
	`explicitAPPC/mainframe/source/ASEJBAPI`
	`explicitAPPC/mainframe/source/CLCLIENT`
	`explicitAPPC/mainframe/source/EJBAPI`
	`explicitAPPC/mainframe/source/EXCLIENT`
	`explicitAPPC/mainframe/source/VTAMDEFINITION`
	`explicitAPPC/mainframe/source/WLCLIENT`

Directory	Files
	`IMS/inbound/gateway/build.sh`
	`IMS/inbound/gateway/chardata.cpy`
	`IMS/inbound/gateway/chardata.egen`
	`IMS/inbound/gateway/Chardata.java`
	`IMS/inbound/gateway/TopicReceive.java`
	`IMS/inbound/mainframe/source/COMPIMSC`
	`IMS/inbound/mainframe/source/DFS62DTI`
	`IMS/inbound/mainframe/source/IMSINDEF`
	`IMS/inbound/mainframe/source/IMSTOJMS`
	`IMS/outbound/gateway/baseClient.egen`
	`IMS/outbound/gateway/BaseClient.java`
	`IMS/outbound/gateway/build.sh`
	`IMS/outbound/gateway/Client.java`
	`IMS/outbound/gateway/ivtno-in.cpy`
	`IMS/outbound/gateway/ivtno-out.cpy`
	`IMS/outbound/gateway/ivtno.egen`
	`IMS/outbound/gateway/IvtnoInRecord.java`
	`IMS/outbound/gateway/IvtnoOutRecord.java`
	`transactional/CICS/outbound/gateway/baseClient.egen`
	`transactional/CICS/outbound/gateway/BaseClient.java`
	`transactional/CICS/outbound/gateway/build.sh`
	`transactional/CICS/outbound/gateway/Client.java`
	`transactional/CICS/outbound/gateway/EmployeeRecord.java`
	`transactional/CICS/outbound/gateway/emprec.cpy`
	`transactional/CICS/outbound/gateway/emprec.egen`

Directory	Files
	`transactional/CICS/outbound/mainframe/source/BLDVSAM`
	`transactional/CICS/outbound/mainframe/source/CMPPROC`
	`transactional/CICS/outbound/mainframe/source/COMPILEV`
	`transactional/CICS/outbound/mainframe/source/CSDUPDCT`
	`transactional/CICS/outbound/mainframe/source/DPLDEMVC`
	`transactional/CICS/outbound/mainframe/source/DPLDEMVD`
	`transactional/CICS/outbound/mainframe/source/DPLDEMVR`
	`transactional/CICS/outbound/mainframe/source/DPLDEMVU`
	`transactional/CICS/outbound/mainframe/source/EMPREC`
	`transactional/CICS/outbound/mainframe/source/JAMVSAMC`
	`transactional/CICS/outbound/mainframe/source/JVSAMRDO`
	`transactional/IMS/outbound/gateway/baseClient.egen`
	`transactional/IMS/outbound/gateway/BaseClient.java`
	`transactional/IMS/outbound/gateway/build.sh`
	`transactional/IMS/outbound/gateway/Client.java`
	`transactional/IMS/outbound/gateway/ivtno-in.cpy`
	`transactional/IMS/outbound/gateway/ivtno-out.cpy`
	`transactional/IMS/outbound/gateway/ivtno.egen`
	`transactional/IMS/outbound/gateway/IvtnoInRecord.java`
	`transactional/IMS/outbound/gateway/IvtnoOutRecord.java`
	`wlintegration/create.xml`
	`wlintegration/read.xml`
	`wlintegration/readString.xml`
`samples/verify`	`CICS/outbound/source/CHARDATA`
	`CICS/outbound/source/CMPPROC`
	`CICS/outbound/source/COMPILE`
	`CICS/outbound/source/CSDU`
	`CICS/outbound/source/CSDUPD`
	`CICS/outbound/source/TOUPCICS`

Directory	Files
	`gateway/outbound/baseClient.egen`
	`gateway/outbound/BaseClient.java`
	`gateway/outbound/build.sh`
	`gateway/outbound/chardata.cpy`
	`gateway/outbound/chardata.egen`
	`gateway/outbound/Chardata.java`
	`gateway/outbound/Client.java`
	`IMS/outbound/source/CHARDATA`
	`IMS/outbound/source/COMPTOUP`
	`IMS/outbound/source/TOUPIMS`
	`IMS/outbound/source/JAMDEFI`

Index