



BEA WebLogic Java Adapter for Mainframe™

Installation and Configuration Guide for DCL SNAP-IX

Copyright

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

Restricted Rights Legend

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

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

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

Trademarks or Service Marks

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

All other trademarks are the property of their respective companies.

BEA WebLogic Java Adapter for Mainframe Installation and Configuration Guide for DCL SNAP-IX

Part Number	Date	Software Version
N/A	April 2002	5.0

Contents

1. Introduction to Installing BEA WebLogic JAM

About BEA WebLogic Product Distribution	1-1
Web Distribution of WebLogic JAM.....	1-1
CD-ROM Distribution of WebLogic JAM	1-2
About the WebLogic JAM Components.....	1-2
About the BEA WebLogic JAM Installation Program.....	1-3
Installation Methods	1-3
About the BEA Home Directory	1-4
About the BEA WebLogic JAM License	1-4
Evaluation Licenses.....	1-5
Non-Expiring Licenses.....	1-5
Acquiring a WebLogic JAM License.....	1-5
Miscellaneous Notes.....	1-5

2. Preparing to Install the BEA WebLogic Java Adapter for Mainframe for DCL SNAP-IX

Download WebLogic JAM for DCL SNAP-IX from the Web.....	2-2
Determine the Installation Method	2-2
Determine Your Hardware Requirements	2-2
Size Requirements for Installing WebLogic JAM for DCL SNAP-IX.....	2-3
Verifying WebLogic JAM Software Requirements	2-4
Choosing a BEA Home Directory	2-4
Creating More Than One BEA Home Directory	2-5

3. Installing BEA WebLogic Java Adapter for Mainframe for DCL SNAP-IX

Locating the Installation Files	3-1
---------------------------------------	-----

Installing WebLogic JAM	3-2
Installing WebLogic JAM Using Graphical-mode Installation.....	3-2
Installing WebLogic JAM in Console Mode.....	3-7

4. Performing Post-Installation Tasks

Verifying Installation Files and Directories	4-1
WebLogic JAM Gateway Installation.....	4-1
WebLogic JAM CRM Installation	4-2
Using the Installation Verification Sample	4-3
Starting the CRM.....	4-3
Configuring the WebLogic JAM Gateway.....	4-4
Updating Your license.bea File	4-5
Updating Your license.bea File on a Unix System.....	4-5
Deploying WebLogic JAM in the WebLogic Server Environment	4-6
Modifying the WebLogic Server CLASSPATH.....	4-7
Deploying the jam.ear File	4-7
Defining the WebLogic JAM Startup Class	4-9
Uninstalling WebLogic JAM for DCL SNAP-IX	4-11
Uninstalling in Graphical Mode	4-11
Uninstalling in Console Mode.....	4-13

5. Configuring WebLogic JAM Connectivity

Understanding WebLogic JAM Connectivity	5-2
Getting Started with WebLogic JAM Connectivity	5-3
Step 1: Define Where the CRM Will Run	5-4
Step 2: Connecting VTAM and Your Local Stack.....	5-5
Creating a VTAM Switched Major Node Definition	5-6
Establishing a DCL SNAP-IX Stack Configuration	5-7

6. CRM Administration

APPC API Tracing	6-1
Setting APPC API Tracing	6-1
Directing the APPC Trace Output	6-2

A. BEA WebLogic JAM Samples Verification and Installation Files

Installation Verification and Samples Files.....	A-2
--	-----

1 Introduction to Installing BEA WebLogic JAM

Proper installation is essential to successfully operating BEA WebLogic Java Adapter for Mainframe (hereafter referred to as WebLogic JAM). The following sections provide an overview of the installation process for WebLogic JAM for use with the DCL SNAP-IX stack.

About BEA WebLogic Product Distribution

The BEA WebLogic JAM for DCL SNAP-IX is distributed only on the Web. Service Packs for WebLogic JAM are distributed only on the Web.

Web Distribution of WebLogic JAM

An evaluation copy of WebLogic JAM 5.0 is available for download from the BEA corporate Web site at http://commerce.bea.com/downloads/weblogic_server.jsp#jam. When you download WebLogic JAM, you get a built-in 30-day evaluation license. BEA provides an additional 60-day evaluation license that can be downloaded from the same location. At the end of the evaluation period, you can purchase a non-expiring license for your WebLogic JAM product.

WebLogic JAM is distributed as an installer file that contains a copy of the BEA installation program, InstallAnywhere.

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 software CD
 - BEA WebLogic Java Adapter for Mainframe Online Documentation CD
- The following printed documents:
 - BEA WebLogic Java Adapter for Mainframe Installation Guide
 - BEA 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.htm>

Note: The WebLogic JAM for DCL SNAP-IX is not included on the CD-ROM.

About the WebLogic JAM Components

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

- The 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 and mainframe systems, such as CICS and IMS. The

Gateway component also forwards configuration information to the CRM at start-up.

- The 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 mainframe applications may use CICS/ESA with Distributed Program Link (DPL) or Advanced Program-to-Program Communications (APPC), or IMS with implicit or explicit APPC.

About the BEA WebLogic JAM Installation Program

The InstallAnywhere installation program is the BEA standard tool for WebLogic product installation. InstallAnywhere installs the WebLogic JAM Gateway and CRM on the target UNIX system.

Installation Methods

InstallAnywhere supports the following installation methods:

- Graphical user interface (GUI) mode installation
- Console-mode installation (UNIX systems only)

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

About the BEA WebLogic JAM 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.

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 offers an extended 60-day evaluation license that can be downloaded from the 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 email. 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.

Acquiring a WebLogic JAM License

You acquire a WebLogic JAM license from BEA in one of two ways: from the BEA Web site when you download an evaluation copy of the WebLogic JAM product, or through email when you purchase WebLogic JAM. When you download WebLogic JAM from the BEA Web site, an evaluation license is included with the distribution.

Miscellaneous Notes

The following items are specific to running WebLogic JAM with DCL SNAP-IX.

- The Solaris 7 and 8 CRM is not persistent. When it loses its Gateway, it goes down and needs to be restored.
- The multiple Gateway to a single CRM feature is not supported with the Solaris 7 or 8 CRM.

2 Preparing to Install the BEA WebLogic Java Adapter for Mainframe for DCL SNAP-IX

Before you install the BEA WebLogic Java Adapter for Mainframe (WebLogic JAM) software, you must complete the following tasks.

- Download WebLogic JAM for DCL SNAP-IX from the Web
- Determine the installation method
- Determine your hardware requirements
- Verify the WebLogic JAM software requirements
- Choose a BEA Home directory

Download WebLogic JAM for DCL SNAP-IX from the Web

An evaluation copy of WebLogic JAM for DCL SNAP-IX 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 for DCL SNAP-IX, you get a built-in 30-day evaluation license. Because WebLogic JAM for DCL SNAP-IX is included in the WebLogic JAM license, it will be updated when you update the WebLogic JAM license.

Determine the Installation Method

On the Solaris 7 or 8 platform, the installation may be accomplished using either a graphical (InstallAnywhere-based) interface or a console (text mode) interface.

After the installation is complete, the installation may be validated using supplied installation verification samples.

After the WebLogic JAM is installed, it may be removed or uninstalled. Similar to the installation interface, the uninstall interface may be either console or graphical.

Determine 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.

Size Requirements for Installing WebLogic JAM for DCL SNAP-IX

The following requirements must be met for the WebLogic JAM to be successfully installed.

- Platform Requirements

A certified Solaris 7 or 8 platform is required for the CRM (see the *BEA WebLogic Java Adapter for Mainframe Release Notes* for the platform requirements for the WebLogic JAM Gateway).

- WebLogic JAM Gateway

- 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 Gateway.

- Memory Requirements

For a 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 requirements for WebLogic JAM Gateway.

- CRM

- Hard Disk Drive Requirements

Approximately 14 MB of free storage space is required to install the WebLogic JAM CRM.

- Memory Requirements

The Solaris DCL SNAP-IX CRM requires an approximate memory size of 9 MB. This will increase with the number of regions configured for the CRM.

Verifying WebLogic JAM Software Requirements

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

- **JDK**

JDK 1.3 or greater is required by the InstallAnywhere installation product to install the WebLogic JAM software. Version 1.3.1 of the JDK is installed with WebLogic Server 6.1 and Service Pack 1 or 2. If you are installing WebLogic JAM without first installing WebLogic Server, set your PATH environment to the appropriate JDK directory. If you do not have the JDK installed, use the console method of installation.

- **WebLogic Server 6.1 Service Pack 1 or 2**

WebLogic Server 6.1 Service Pack 1 or 2 is required for proper operation of WebLogic JAM if you are installing the WebLogic JAM Gateway. 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 the installation of WebLogic JAM, you are prompted to choose an existing BEA Home directory or to specify a path for a new BEA Home directory convention. For example, you can install only one instance of WebLogic JAM 5.0 in a BEA Home directory, but that BEA Home directory may also contain an instance of WebLogic JAM 4.2.

Creating More Than One BEA Home Directory

Although it is possible to create more than one BEA Home directory, we recommend that you avoid doing so. In almost all situations, a single BEA Home directory is sufficient. There may be circumstances, however, in which you 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 BEA WebLogic Java Adapter for Mainframe for DCL SNAP-IX

After you have reviewed the Preparing to Install WebLogic Java Adapter for Mainframe for DCL SNAP-IX section, you are ready to begin installing WebLogic JAM.

Installing WebLogic JAM includes the following tasks:

- Locating the Installation Files
- Installing WebLogic JAM

Locating the Installation Files

The installation files for installing WebLogic JAM for DCL SNAP-IX are available by download only from the BEA download site. You will need to download `wljam50_So17DCL7.bin` or `wljam50_DCL7.bin` file.

Installing WebLogic JAM

BEA WebLogic JAM can be installed in the following ways:

- In Graphical Mode

Graphical-mode installation is the graphics-based method of executing the BEA installation program.

- In Console Mode

Console-mode installation is the text-based method of executing the BEA installation program.

Installing WebLogic JAM Using Graphical-mode Installation

Perform the following steps to install the WebLogic JAM in Graphics mode.

1. Log on as root to install the WebLogic JAM software.

```
$ su -  
Password:
```

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

```
export JAVA_HOME=/usr/BEA/jdk131  
export PATH=/usr/BEA/jdk131/bin:$PATH
```

3. 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-Window terminal emulator.

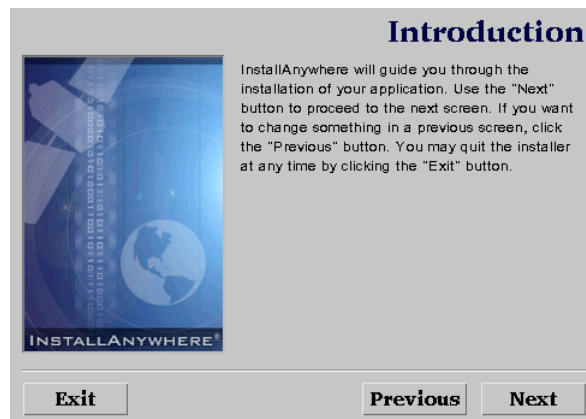
4. Execute the installation program. For example, on Solaris 8 enter:

```
sh wljam50_DCL7.bin
```

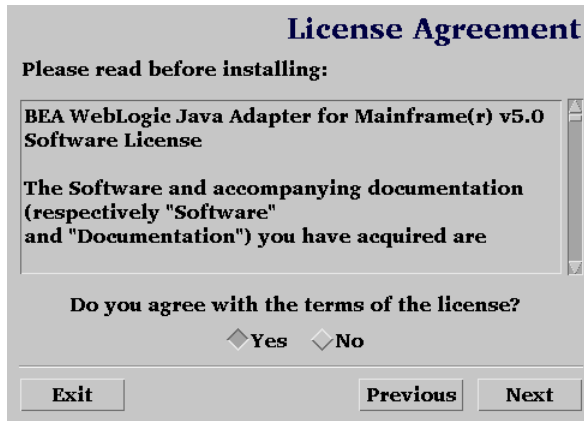
The BEA WebLogic Java Adapter for Mainframe splash screen displays.



The splash screen is followed by the Introduction screen.

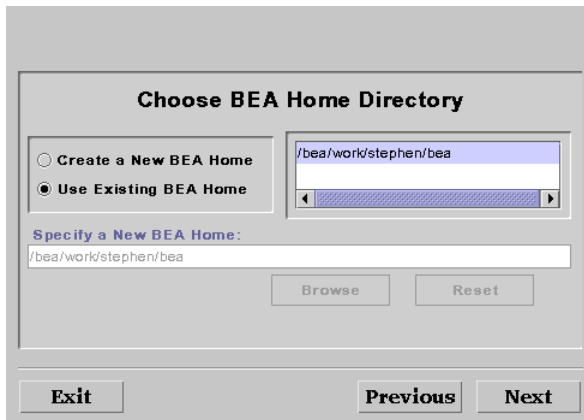


5. Click Next to continue with the installation. The License Agreement screen displays.



After reading the license agreement, click Yes to accept the license agreement and click Next to proceed with the installation. The Choose BEA Home Directory screen displays.

Note: If you click No, you will not be able to continue with the installation. You must either exit the installation or go back to the Introduction screen.



6. Create or select a BEA Home directory (the default is bea). The BEA Home directory is the root directory in which you install all BEA products.

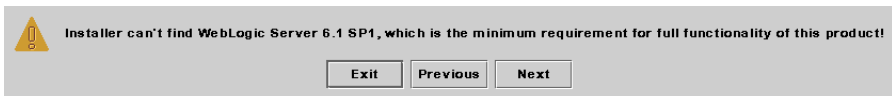
If you do not already have a BEA Home directory, you can create one by following these steps:

- a. Select the Create a New BEA Home option button.

- b. Accept the default location for your BEA Home directory or click Browse to specify a different location. If you accept the default location, your BEA Home directory will be `bea`, and your WebLogic JAM software default installation directory is `bea/work/userid/bea`. If you do not want to install WebLogic JAM in this default location, you will have the opportunity, in Step 11, to choose another directory in which to install the software.
- c. Click Next to proceed with the installation.

If you already have one or more BEA Home directories on your system, you can use one of those directories as follows:

- a. Select the Use Existing BEA Home option button.
 - b. Select a BEA Home directory from the list displayed to the right of the option buttons. All valid BEA Home directories are displayed in this list. Valid BEA Home directories are directories where BEA products have been installed using the standard installation program.
 - c. Click Next to proceed with the installation.
7. If the installation program encounters a previous version of WebLogic JAM in the selected BEA Home directory, the Previous Installation Found message box displays. From this message box, do one of the following:
- Click Continue to install over the previous version.
 - Click Cancel to return to the Choose BEA Home Directory screen and select another directory.
 - Click Exit to exit the installation program.
8. If you do not have a successful installation of WebLogic Server 6.1 Service Pack 1 or 2 in the BEA Home directory, the following warning box displays.



From this warning box, do one of the following:

- Click Exit to exit the installation program.
- Click Previous to return to the Choose BEA Home Directory screen and select another directory.

- Click Next to proceed with the installation process. However, WebLogic JAM will not be functional until you perform a successful installation of the WebLogic Server Service Pack .
9. If this is your first WebLogic JAM installation, once you have selected a BEA Home Directory in which to install WebLogic JAM, the Choose Product Directory screen displays.



Choose Product Directory

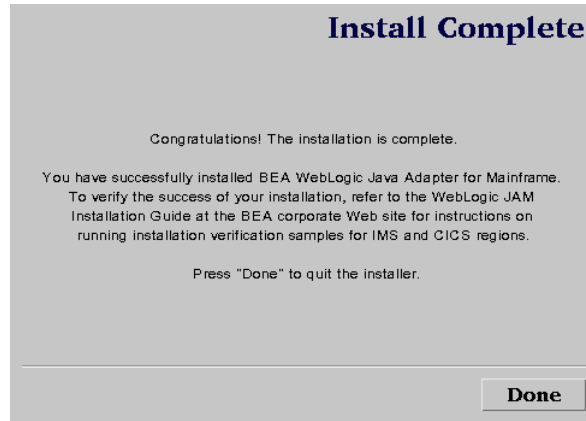
Specify Product Installation Directory:

/bea/work/stephen/bea/NewFolder/wljam5.0

Browse Reset

Exit Previous Install

10. Specify an installation directory for WebLogic JAM by performing one of the following actions:
- Accept the default installation directory.
 - Enter the installation directory directly into the text entry field.
 - Click Browse and navigate to your desired installation root directory, then click OK to make the directory path appear in the Specify Product Installation Directory text entry field.
- Note:** Clicking Reset resets the installation directory to the previous value (before you made any changes).
11. Click Install to accept the product directory you have chosen and proceed with the installation. WebLogic JAM will create a `wljam5.0` folder in the BEA Home directory. The Install Complete screen displays once you click Install.



12. Click Done to exit the installation program.

Installing WebLogic JAM in Console Mode

To install WebLogic JAM in console-mode, run the `wljam50_Sol7DCL7.bin` or `wljam50_DCL7.bin` program.

Perform the following steps to install WebLogic JAM in console mode:

1. Log on as root.

```
$ su -
```

Password:

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

```
export JAVA_HOME=<BEA_HOME>/jdk130
```

```
export PATH=<BEA_HOME>/jdk130/bin:$PATH
```

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

```
export JAVA_HOME=/usr/bea/jdk130
```

```
export PATH=/usr/bea/jdk130/bin:$PATH
```

3. Execute the appropriate installation program for your non-mainframe Unix system in character-based mode. For example, to execute the WebLogic JAM for DCL SNAP-IX installation program for Solaris 8, enter the following:

```
sh wljam50_DCL7.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:
```

```
=====
=====
```

```
License Agreement
```

```
-----
```

```
Please read before installing:
```

```
BEA WebLogic Java Adapter for Mainframe(r) 5.0 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.

For purposes of this agreement:

Client Software means the Software programs provided as part of the Software that operate on an intelligent, single user device, permitting that device access to the shared resources provided by Server Software.

Server Software means the Software programs provided as part of the Software that are designed to operate on a single computing device, providing access to the shared resources attached to that computer. Server Software may contain technical limitations that limit use of Server Software to a specified number of Users.

User Count means the number of users specified on the uniform product code label provided on the outside packaging of the Product.

PRESS <ENTER> TO CONTINUE:

Connection Management Software means the computer programs provided as part of the Software which monitor the number of connections permitted under this License and which are designed to prevent more connections than the number of licensed connections specified by the User Count.

User means a connection to the Server Software permitting a user or a process access to the shared resources provided by Server Software.

Domain means a single unit Server Software (network server) or a collection of Server Software defined as an administratively autonomous System application(s).

The Software contains various Software programs with different license rights. Some of the programs are licensed for use on a single Domain, whereas some are licensed for use on several machines (workstations). The type of license that applies depends on the following definitions and the permitted uses specified in the documentation.

License. Subject to the terms and conditions of this License, BEA grants to you a non-exclusive, non-transferable right to (a) use the Server Software on a single Domain and (b) provided you do not

receive any direct payment, to make, distribute and simultaneously use internally within your organization, PRESS <ENTER> TO CONTINUE:

up to the User Count number of copies of the Client Software. The Client Software may not be used to support other than the Server Software. You may not modify the Connection Management Software or otherwise multiplex workstations to increase the number of connections supported by the Server Software. Further, you may not use any device, process or computer program in conjunction with the Server Software which increases, either directly or indirectly, the number of connections to the Server Software.

Except as permitted by law despite this restriction, you agree, not to reverse compile, disassemble, or otherwise reverse engineer the Software. You also agree not to lease, rent, sublicense or otherwise transfer the Software to any third party, nor to modify or create a derivative work of the Software or otherwise use them except as provided for in this license.

BEA reserves all rights not expressly granted to you.

General Terms

Term. This License will become effective on the date you acquire the Software and will remain in force until terminated. You may terminate this License at any time by destroying the Documentation and Software together with all copies and adaptations. This License shall also automatically terminate if you PRESS <ENTER> TO CONTINUE:

breach any of the terms or conditions. You agree to destroy the original and all copies of the Software and Documentation, or to return them to BEA upon termination of this License.

Governing Law. Except as otherwise restricted by law, this License shall be governed by, and interpreted in accordance with, the laws of the State of California of the United States of America, without regard to laws of California governing conflicts of law. Furthermore, it is understood that this License shall be treated as though it were executed in the State of California. The parties agree that any action relating to this License shall be instituted and prosecuted in the courts of competent jurisdiction of the State of California.

Entire License. This License sets forth the entire understanding and License between you and BEA and may be amended only in writing signed by both parties. NO VENDOR, DISTRIBUTOR, DEALER, RETAILER, SALES PERSON OR OTHER PERSON IS AUTHORIZED TO MODIFY THIS LICENSE OR TO MAKE ANY WARRANTY, REPRESENTATION OR PROMISE WHICH IS DIFFERENT THAN, OR IN ADDITION TO, THE REPRESENTATIONS OR PROMISES OF THIS LICENSE.

Waiver. No waiver of any right under this License shall be effective unless in writing and signed by a duly authorized representative off the party to be PRESS <ENTER> TO CONTINUE:

bound. No waiver of any past or present right arising from any breach or failure to perform shall be deemed to be a waiver of any future right arising under this License.

Severability. If any provision in this License is invalid or unenforceable, that provision shall be construed, limited, modified or, if necessary, severed, to the extent necessary, to eliminate its invalidity or unenforceability, and the other provisions of this License shall remain unaffected.

Export. BEA can only make the Software available pursuant to a valid export license. Consequently, BEA will refund any fees paid by you if it is unable to obtain an export license to ship the Software to you. Your sole remedy to BEA for BEA's failure to obtain an export license is a refund. You hereby agree that you will not knowingly, directly or indirectly, without prior written consent, if required, of the Office of Export Licensing of the U.S. Department of Commerce, Washington D.C. 20230, download, export, transmit or transfer the Software to any group Q, S, W, Y, or Z country specified in the Export Administration Regulations issued by the U.S. Department of Commerce or to any country to which such transmission is restricted by applicable regulations or statutes.

PRESS <ENTER> TO CONTINUE:

U.S. Government Restricted Rights. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in FAR 52.227-14 (June 1987) Alternate III(g)(3) (June 1987), FAR 52.227-19 (June 1987), or DFARS 52.227-7013 (c)(1)(ii) (June 1988), as applicable. Contractor/Manufacturer is BEA Systems, Inc. 2315 North First Street, San Jose, CA 95131

BEA is a registered trademark of BEA Systems, Inc.

BEA Tuxedo is a registered trademark of BEA Systems, Inc. in the US and other countries

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

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

=====
=====

Choose BEA Home Directory

3 *Installing BEA WebLogic Java Adapter for Mainframe for DCL SNAP-IX*

```
-----

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

Enter a number: 2

    1- /bea/loads
    2- /cmhome/wls61

Existing BEA Home: 2

=====
=====

Choose Product Directory

-----

    1- Modify Current Selection (/cmhome/wls61/wljam5.0)
    2- Use Current Selection (/cmhome/wls61/wljam5.0)
Enter a number: 2

=====
=====

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:

4 Performing Post-Installation Tasks

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

- Verifying Installation Files and Directories
- Using the Installation Verification Sample
- Updating Your license.bea File
- Deploying WebLogic JAM in the WebLogic Server Environment
- Uninstalling WebLogic JAM for DCL SNAP-IX

Verifying Installation Files and Directories

The following libraries and executable programs are installed for each of the following installation components. After installing the WebLogic JAM software, verify that these libraries and programs are installed on your system.

WebLogic JAM Gateway Installation

Verify that the WebLogic JAM Gateway files listed by section are installed on your system. All directories are relative to the specified installation directory.

Table 4-1 WebLogic JAM Gateway Installation Files and Directories

Directory	Files
bin	egencobol
lib	jam.jar jam.ear jamdcl.jar
classdocs	jamdoc.jar
config	See Appendix A for a list of files.
samples	See Appendix A for a list of files.

WebLogic JAM CRM Installation

Verify that the WebLogic JAM CRM files listed by section are installed on your system. All directories are relative to the specified installation directory.

Table 4-2 WebLogic JAM CRM Installation Files and Directories

Directory	Files
bin	CRMLOGS CRM crmlkon crmlkoff crmdown solaris.env

Directory	Files
lib	libctxdebugs_12.so libctxos_12.so libctxprim_12.so libctxplat_12.so libcsxappc.so libcsxxfm.so libcsxgpw.so libcsxcrm.so libcsxscrm.so libcsxsp70.so libengine.so.71 libgpnet.so.71

Using the Installation Verification Sample

After you have verified that the files have been installed, run the Installation Verification Sample.

Note: Please refer to the “Using the Installation Verification Samples” topic in the *BEA WebLogic Java Adapter for Mainframe Installation Guide* for information and instructions on running the installation verification samples.

Starting the CRM

Scripts are commonly used when starting the CRM. Two scripts are provided to ease the use of the CRM. The first script, `solaris.env`, is installed with the CRM in the `<CRM_INSTALL_DIR>/bin` directory. This script appends the necessary values to your environment variables. The second script, `startcrm.sh`, is a CRM start-up script that is installed with the Gateway in the `<JAM_INSTALL_DIR>/samples/crm/unix` directory.

1. Copy the `startcrm.sh` script to the directory from which the CRM will be run.

2. Edit `solaris.env`. Supply the correct values for the `APPDIR` and `CRMDIR` variables. `APPDIR` is the directory from which the CRM will be run. `CRMDIR` is the CRM installation directory. Also include the stack library in the `LD_LIBRARY_PATH` variable if this is not already set in your environment.
3. Copy the `solaris.env` to the `crm.env`.
4. Edit `startcrm.sh`. To use a different port, change the port number. However, if you change the port number, make sure to change the port number in the WebLogic Administration Console CRM1 panel. You do not need to change the address because the script will run on the machine where the CRM is installed. BEA recommends that you do not change the CRM name from CRM1. This name for the CRM is pre-configured for all of the samples.
5. Execute the `startcrm.sh` script:

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

The port number is 7101. You can change the port number; however, if the port number is changed, make sure to change the port number in the Gateway configuration CRM1 panel.

Listing 4-1 Value of the `STARTCMD` parameter in the `CRMSTART JCL`

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

Configuring the WebLogic JAM Gateway

Most configuration tasks were pre-configured or were completed during the installation process by the installer program. For additional information about configuring WebLogic JAM, refer to the BEA WebLogic Java Adapter for Mainframe *Configuration and Administration Guide*.

You must make some configuration changes for the Installation Verification Sample for CICS or IMS to run on your system. To configure the CRM to the WebLogic JAM Gateway, use the WebLogic Administration Console.

You must also edit the `startverifyServer.sh` to add the `jamdcl.jar` filename to the `CLASSPATH`. The `jamdcl.jar` must be defined in the `CLASSPATH` before the `jam.jar`.

Note: Please refer to the “Using the Installation Verification Samples” topic in the *BEA WebLogic Java Adapter for Mainframe Installation Guide* for additional information and instructions on configuring the the WebLogic JAM Gateway.

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 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 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=./jdk131/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. 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 `jamdc1.jar` and `jam.jar` to the WebLogic Server `CLASSPATH`
 - Deploying `jam.ear`
 - Defining the WebLogic JAM startup class

Modifying the WebLogic Server CLASSPATH

The `jamdcl.jar` and the `jam.jar` files add classes to the WebLogic Server CLASSPATH. The `.jar` files must be added to the CLASSPATH in the WebLogic Server startup script for each server where it will run.

To manually add the `jamdcl.jar` and the `jam.jar` files to the WebLogic Server CLASSPATH, perform the following steps:

1. Using the text editor of your choice, locate the script you use to start WebLogic Server in your domain. For example, use the `startWebLogic` script in your `config/mydomain` directory *mydomain* is the name of your domain.
 - For example, use your `startWebLogic.sh` script in your `config/mydomain` directory where *mydomain* is the name of your domain and `startWebLogic.sh` is the script to start WebLogic Server.
2. Edit the `startWebLogic` script by adding the `jamdcl.jar` and the `jam.jar` file to the CLASSPATH setting as shown in the following examples:

```
CLASSPATH=$WL_HOME:$WL_HOME/lib/weblogic_sp.jar:$WL_HOME/lib/weblogic.jar:./ext/weblogic-tags.jar:/bea/wljam.50/lib/jamdcl.jar:/bea/wljam.50/lib/jam.jar
```

Note: The `jamdcl.jar` must be defined in the CLASSPATH before the `jam.jar`.

In this example, `/bea/wljam.50` is the install directory for WebLogic JAM.

3. Save the file and close the text editor.

Deploying the jam.ear File

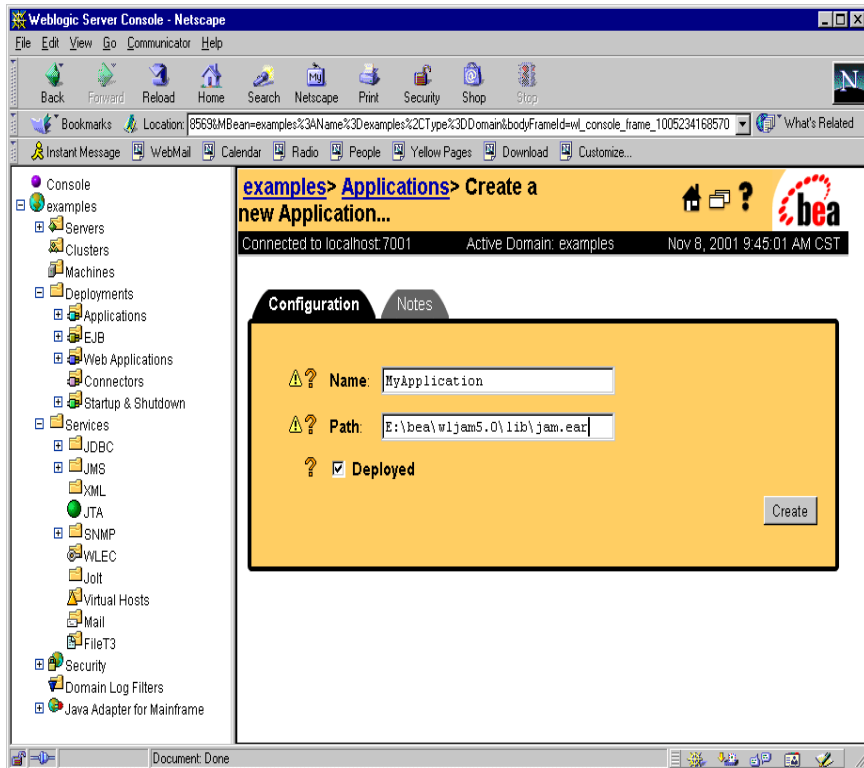
The `jam.ear` file deploys the WebLogic JAM applications and the WebLogic JAM extension to the WebLogic Administration Console. This file contains the following EJBs:

- `jambeans`
- `jamtojms`
- `jam.war` (for the console)

Deploy the `jam.ear` file on the appropriate servers using the WebLogic Administration Console. To manually deploy the `jam.ear` file, perform the following steps:

1. In the WebLogic Administration Console, select **Deployments**→**Applications** to invoke the Applications table.
2. Click on **Install a new Application**. Browse to locate the `jam.ear` file in `wljam5.0\lib` directory. Click **Upload**.
3. Click on **Applications** to return to the **Applications** table.
4. Click on **Configure a new Application** to invoke the **Create a new Application** page.

5. Fill in the fields for this configuration entry.
 - a. Assign a name to the application
 - b. Indicate the path to the application (.ear file)
 - c. Select **Deployed** and click **Create** to create the new entry.

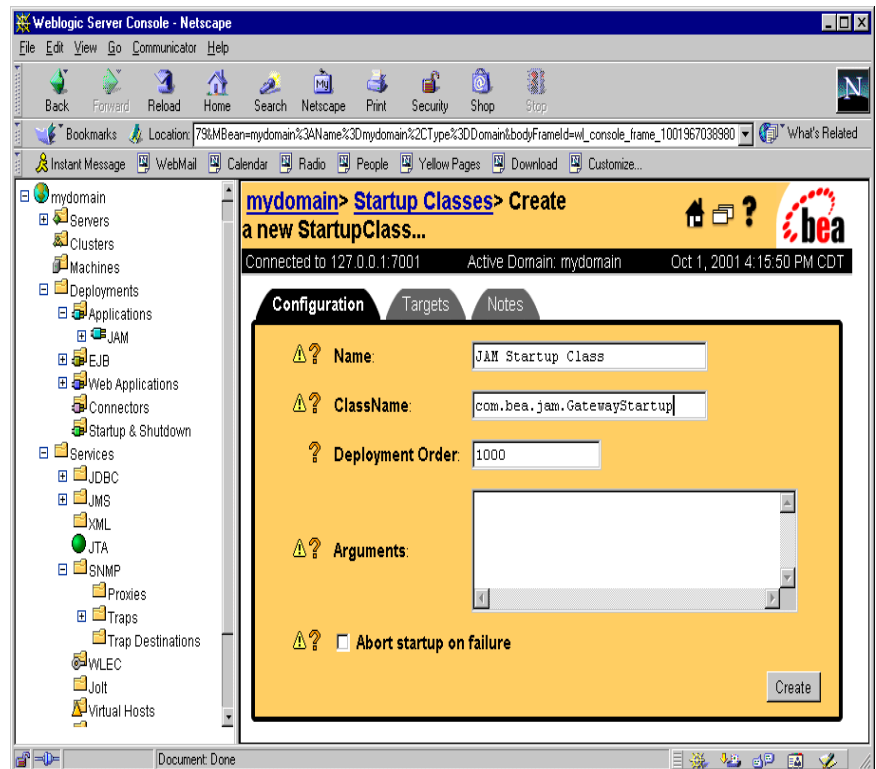


Defining the WebLogic JAM Startup Class

The WebLogic JAM startup class defines a class that is launched at WebLogic Server startup. To manually define the WebLogic JAM startup class using the WebLogic Administration Console, perform the following steps:

4 Performing Post-Installation Tasks

1. In the WebLogic Administration Console, select **Deployments**—>**Startup & Shutdown** to invoke the Startup & Shutdown table.
2. Click on **Configure a new Startup Class** to invoke the **Create a new Startup Class** page.
3. Fill in the fields.
 - a. Assign a name for the **Name** field.
 - b. The **ClassName** must be `com.bea.jam.GatewayStartup`.
4. Click **Create** to create the new startup class.



5. Select the **Targets** tab. Select the target server from the available list and click the arrow to add it to the chosen list. Click **Apply** to apply the new startup class.

After you have completed the post-installation tasks for WebLogic JAM, refer to the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide* for information about configuring WebLogic JAM.

Uninstalling WebLogic JAM for DCL SNAP-IX

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:

- In Graphical Mode

Graphical-mode installation is the graphics-based method of executing the BEA installation program.

- In Console Mode

Console-mode installation is the text-based method of executing the BEA installation program.

Uninstalling in Graphical Mode

To uninstall WebLogic JAM for DCL SNAP-IX using graphical mode, perform the following steps:

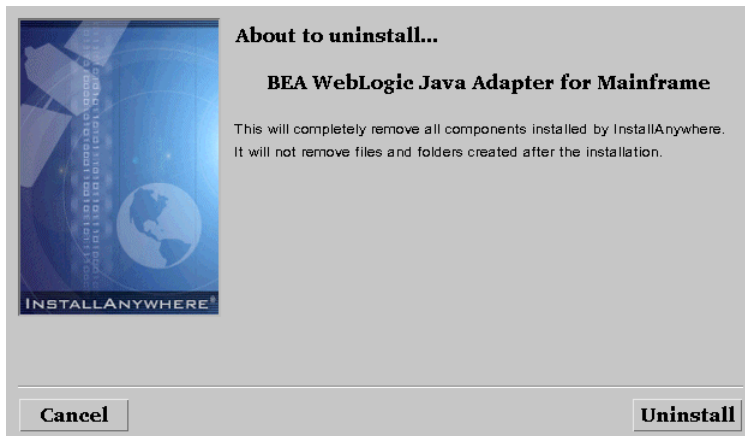
1. Change directories to your uninstaller directory. The following command offers an example:

```
>cd /usr/beam/wljam5.0/uninstaller
```

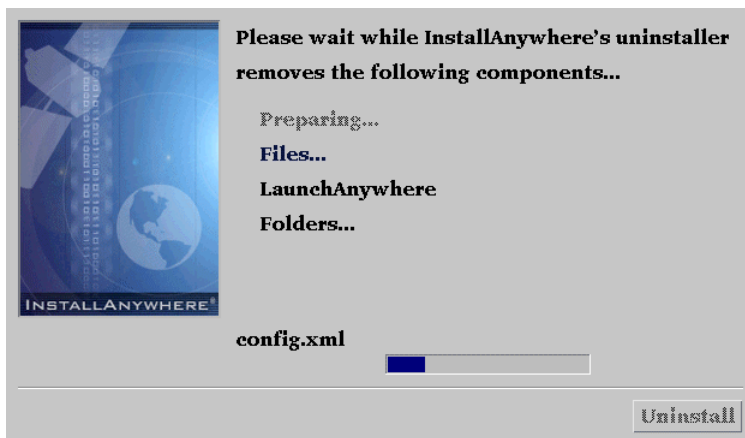
2. Start the uninstallation program by performing the following command:

```
/usr/beam/wljam5.0/uninstaller> sh Uninstall
```

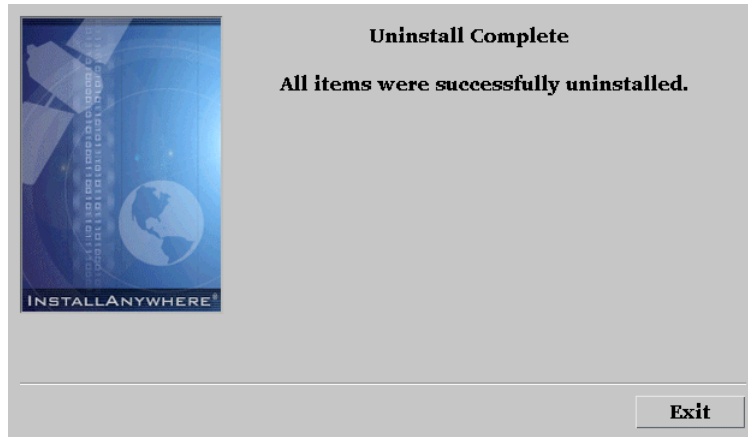
The About to uninstall screen displays.



3. Click Uninstall to start uninstalling BEA WebLogic Java Adapter for Mainframe. The following screen displays.



4. When the following screen displays, uninstallation is complete.



5. Click Exit to complete the uninstallation.

Uninstalling in Console Mode

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

1. Change directories to the uninstaller directory. For example:

```
> cd /usr/bea/wljam5.0/uninstaller
```
2. Start the uninstallation program using the following example command:

```
/usr/bea/wljam5.0/uninstaller> sh Uninstall -i console
```
3. At the prompt, press ENTER to begin the uninstall. The following listing displays a WebLogic JAM uninstallation. If you want to exit the uninstall program, enter QUIT at the prompt.

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

Listing 4-2 Unix Uninstallation Example

```
machine name:/bea/loads/wljam5.0/uninstaller sh Uninstall -i
console

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:

=====
=====

Uninstall...

-----

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

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

=====
=====

Uninstall Complete

-----
```

All items were successfully uninstalled.

PRESS <ENTER> TO EXIT THE UNINSTALLER:

5 Configuring WebLogic JAM Connectivity

The WebLogic Administration Console provides you with the tools you need to configure connectivity between your mainframe and WebLogic Server.

This section provides information on the following subjects:

- Understanding WebLogic JAM Connectivity
- Step 1: Define Where the CRM Will Run
- Step 2: Connecting VTAM and Your Local Stack

Note: Please refer to the “Configuring WebLogic Connectivity” topic in the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide* for instructions on performing steps 3, 4, 5, and 6.

- Step 3: Connect the CRM to Back-End System on the Mainframe
- Step 4: Enter Connectivity Information into WebLogic Administration Console
- Step 5: Define a WebLogic JAM Gateway
- Step 6: Verify Your WebLogic JAM Connectivity Configuration

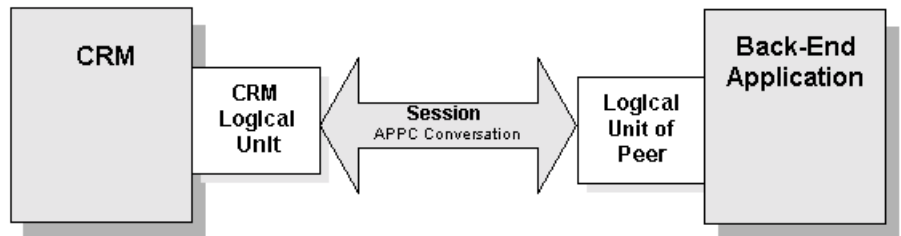
Understanding WebLogic JAM Connectivity

WebLogic JAM uses two distributed software components to connect to your back-end systems: the WebLogic JAM Gateway and the Communications Resource Manager (CRM). The WebLogic JAM Gateway component runs within an instance of WebLogic Server and serves as a proxy to other applications running within WebLogic Server.

The CRM runs as a native operating system process, and it connects to your back-end system using Advanced Program to Program Communications (APPC), also known as LU 6.2, using SNA network connections. The CRM and the WebLogic JAM Gateway communicate with each other using a TCP/IP socket. The CRM connects to your back-end system using an SNA network connection called a logical unit (LU).

A logical unit is an SNA network's way of providing access to the SNA network to end users and software programs. A logical unit is a unique, addressable part of an SNA network that manages data flows between network partners. A logical unit is somewhat like a TCP/IP address and port because software programs can use it to access the network and to communicate with other software programs that are distributed throughout the enterprise. Unlike TCP/IP connections, logical units must be defined prior to use. [Figure 5-1](#) shows the CRM using APPC to establish communication with a back-end application.

Figure 5-1 CRM Using APPC



In order for software programs such as the CRM to communicate via APPC, each peer program must have access to a logical unit. The software programs then allocate a session between the two logical units to communicate and collaborate.

A session is a pipeline between two logical units that manages the exchange of data between the logical units. The two peer programs send data back and forth over the logical unit pair using a session. This exchange of data is called a conversation. The number of sessions that can exist simultaneously over a given logical unit pair is configured in the SNA network. The number of sessions that are configured between an LU-LU pair, determine how many concurrent conversations can occur simultaneously. Establishing WebLogic JAM connectivity involves allocating logical units and sessions in your SNA network, and recording this configuration in WebLogic JAM.

The WebLogic Administration Console allows you to define where WebLogic JAM components will run within your enterprise, and the network connections that they will establish. Once entered into the console, this configuration is persisted and distributed to WebLogic JAM components upon start-up.

Getting Started with WebLogic JAM Connectivity

The overall task of establishing WebLogic JAM connectivity involves asking administration personnel to allocate SNA network resources (logical units, sessions) and then recording these resources in WebLogic JAM configuration via the WebLogic Administration Console. This task has been organized into six primary steps:

- Step 1: Define Where the CRM Will Run
- Step 2: Connecting VTAM and Your Local Stack

Note: Please refer to the “Configuring WebLogic Connectivity” topic in the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide* for instructions on performing steps 3, 4, 5, and 6.

- Step 3: Connect the CRM to Back-End System on the Mainframe
- Step 4: Enter Connectivity Information into WebLogic Administration Console
- Step 5: Define a WebLogic JAM Gateway
- Step 6: Verify Your WebLogic JAM Connectivity Configuration

Your system administrators will configure your mainframe to communicate using SNA and then establish the actual connection to the CRM using the parameter information provided in the steps of this section.

When you have all of the appropriate parameters, you are ready to configure connectivity by entering them into the WebLogic Administration Console. Use the steps in this section to help you and your system administrators connect your systems correctly and to get the correct configuration information into the WebLogic Administration Console. Instructions are also provided to help you verify your configuration.

Note: The Mainframe Connectivity Worksheet is provided to help you prepare to configure connectivity between your mainframe system and your BEA WebLogic Server system. For a copy of the worksheet, refer to the *BEA WebLogic Java Adapter for Mainframe Configuration and Administration Guide*.

Step 1: Define Where the CRM Will Run

The CRM can run on Solaris 7 or 8 with a DCL SNAP-IX stack.

For this step, you need to determine where you want the CRM to run and then gather the configuration information that you will input into the WebLogic Administration Console in a later step. [Table 5-1](#) provides a list of the parameters that you will use to configure connectivity with the CRM in the WebLogic Administration Console.

Table 5-1 CRM Definition Parameter in the WebLogic Administration Console

WebLogic Administration Console Parameters	Description	Parameter Syntax
Name	The arbitrary name of the CRM. The CRM process uses this name to ensure the correct configuration between the WebLogic JAM Gateway and the CRM process.	Alpha-numeric string Ex. CRM1

WebLogic Administration Console Parameters	Description	Parameter Syntax
Listen Address	The host name of the machine where the CRM runs, or the TCP/IP address of the machine where the CRM runs.	Alpha-numeric string or dotted IP address Ex. myhost or 123.4.5.678
Listen Port	The TCP/IP port that is used by the CRM to listen for connecting WebLogic JAM components. Ask your network administrator for an unused TCP/IP port on the machine where the CRM runs.	Numeric value Ex. 8002
Stack Type	The predefined name for the SNA network software. If the CRM is running on Solaris 7 or 8 with the DCL SNAP-IX stack, this entry is the abbreviation spx70.	Supported stack available in drop-down list. Ex. spx70

Step 2: Connecting VTAM and Your Local Stack

Connecting VTAM to your local stack consist of the following two steps:

- Creating a VTAM Switched Major Node Definition
- Establishing a DCL SNAP-IX Stack Configuration

Creating a VTAM Switched Major Node Definition

VTAM Switched Major Node (SWNET) definitions define and link together physical units (PU) and logical units (LU). Each workstation connected to a network must be represented on the network as a PU and each PU can have one or more LUs. The VTAM SWNET definition defines how the PUs and LUs communicate with a server. An example of a SWNET definition is listed below.

Listing 5-1 Sample VTAM SWNET Definition

```
SWNEBEA  VBUILD  TYPE=SWNET, MAXNO=3, MAXGRP=3
SWNEBEA1  PU  ADDR=03,
           IDBLK=021,
           IDNUM=10333,
           PUTYPE=2,
           NETID=BEALAN,
           CPNAME=BEASUN,
           MAXPATH=3,
           DWACT=YES,
           CONNTYPE=APPN,
           DYNLU=YES,
           CPCP=YES
PATHDI1   PATH  DIALNO=3767,
           GRPNM=G1ETH1
LUBEAO1   LU   LOCADDR=0
LUBEAO2   LU   LOCADDR=0
```

The fields that are most relevant to WebLogic JAM and the local stack configuration are listed in the table below.

Table 5-2 VTAM SWNET Field Definitions

Field	Field From Example	Definition and Usage
SWNET Definition Name	SWNEBEA1	The name of the SWNET definition that must be varied active before the stack will connect to your mainframe.

Field	Field From Example	Definition and Usage
PU Name	SWNEBEA1	The PU name of the link station expecting to connect to the mainframe. This field appears as a link station in your local stack configuration.
IDBLK	IDBLK=021	Must be unique.
IDNUM	IDNUM=10333	Must be unique.
DIALNO	DIALNO=1874560398634502	The MAC address of the Solaris 7 or 8 machine where the DCL SNAP-IX stack will run.
NETID	NETID=BEALAN	The network ID which is also the first half of the control point name for your DCL SNAP-IX node.
CPNAME	CPNAME=BEASUN	The control point name which appears as the second half of the control point name for your DCL SNAP-IX node.
LU Name	LUBEAO1	Name of the LU that is defined on your local stack configuration. It is also used as the netname in the session definition within your CICS region.

Establishing a DCL SNAP-IX Stack Configuration

A DCL SNAP-IX stack configuration can be set up using stack-specific configuration utilities or by manually creating a configuration file with any text editor. The following listing shows an example of an SNA node configuration file. The configuration file is divided into sections for various components of the configuration. Each section defines a component using parameters (or keywords) and values. Some of these keywords and values will affect how your WebLogic JAM configuration is defined.

Consult with your VTAM administrator to obtain key parameters in the VTAM SWNET definition that must be included in the SNA stack configuration.

Listing 5-2 Example of a sna_node.cfg File

```
[define_node_config_file]
major_version = 5
minor_version = 1
update_release = 1
revision_level = 57

[define_node]
cp_alias = dalsun8
description = ""
fqcp_name = BEALAN.BEASUN
node_type = LEN_NODE
mode_to_cos_map_supp = YES
mds_supported = YES
node_id = <bea89476>
max_locates = 1500
dir_cache_size = 255
max_dir_entries = 0
locate_timeout = 0
reg_with_nn = YES
reg_with_cds = YES
mds_send_alert_q_size = 100
cos_cache_size = 24
tree_cache_size = 40
tree_cache_use_limit = 40
max_tdm_nodes = 0
max_tdm_tgs = 0
max_isr_sessions = 1000
isr_sessions_upper_threshold = 900
isr_sessions_lower_threshold = 800
isr_max_ru_size = 16384
isr_rcv_pac_window = 8
store_endpt_rscvs = NO
store_isr_rscvs = NO
store_dlur_rscvs = NO
cos_table_version = VERSION_0_COS_TABLES
send_term_self = NO
disable_branch_awareness = NO
cplu_syncpt_support = NO
cplu_attributes = NONE
dlur_support = NO
pu_conc_support = YES
nn_rar = 128
max_ls_exception_events = 0
ptf_flags = NONE
```

```
[define_ethernet_dlc]
dlc_name = ETHER0
description = ""
neg_ls_supp = YES
initially_active = NO
adapter_number = 0
lan_type = 802_3_DIX
```

```
[define_ethernet_port]
port_name = ETSAP0
description = ""
dlc_name = ETHER0
port_type = PORT_SATF
port_number = 0
max_rcv_btu_size = 1033
tot_link_act_lim = 64
inb_link_act_lim = 0
out_link_act_lim = 0
ls_role = LS_NEG
implicit_dspu_services = NONE
implicit_dspu_template = ""
implicit_ls_limit = 0
act_xid_exchange_limit = 9
nonact_xid_exchange_limit = 5
ls_xmit_rcv_cap = LS_TWS
max_ifrm_rcvd = 7
target_pacing_count = 7
max_send_btu_size = 1033
mac_address = <937694760285>
lsap_address = 0x04
implicit_cp_cp_sess_support = NO
implicit_limited_resource = NO
implicit_deact_timer = 30
implicit_hpr_support = NO
implicit_link_lvl_error = NO
implicit_uplink_to_en = NO
effect_cap = 3993600
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
initially_active = YES
```

```
window_inc_threshold = 1
test_timeout = 10
test_timer_retry = 5
xid_timer = 10
xid_timer_retry = 5
ack_timeout = 5000
p_bit_timeout = 5000
t2_timeout = 100
rej_timeout = 10
busy_state_timeout = 30
idle_timeout = 30
max_retry = 3
```

```
[define_ethernet_ls]
ls_name = SWNEBEA1
description = ""
port_name = ETSAP0
adj_cp_name = P390.MVSREGION
adj_cp_type = END_NODE
mac_address = <197459563870>
lsap_address = 0x04
auto_act_supp = NO
tg_number = 0
limited_resource = NO
solicit_sscp_sessions = YES
pu_name = BEASUN1
disable_remote_act = NO
default_nn_server = NO
dspu_services = NONE
dspu_name = <0000000000000000>
dlus_name = <00000000000000000000000000000000>
bkup_dlus_name = <00000000000000000000000000000000>
hpr_supported = NO
hpr_link_lvl_error = NO
link_deact_timer = 30
use_default_tg_chars = YES
ls_attributes = SNA
adj_node_id = <00000000>
local_node_id = <00000000>
cp_cp_sess_support = NO
effect_cap = 3993600
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
```



```
user_def_parm_3 = 128
target_pacing_count = 7
max_send_btu_size = 1033
ls_role = USE_PORT_DEFAULTS
max_ifrm_rcvd = 0
dlus_retry_timeout = 0
dlus_retry_limit = 0
branch_link_type = NONE
adj_brnn_cp_support = ALLOWED
dddlu_offline_supported = NO
initially_active = NO
restart_on_normal_deact = NO
react_timer = 30
react_timer_retry = 65535
test_timeout = 10
test_timer_retry = 5
xid_timer = 10
xid_timer_retry = 5
ack_timeout = 5000
p_bit_timeout = 5000
t2_timeout = 100
rej_timeout = 10
busy_state_timeout = 30
idle_timeout = 30
max_retry = 3
```

```
[define_partner_lu]
plu_alias = CICS1
description = ""
fqplu_name = P390.CICSREGN
plu_un_name = CICSREGN
parallel_sess_supp = YES
max_mc_ll_send_size = 0
conv_security_ver = NO
```

```
[define_local_lu]
lu_alias = LUBEA01
list_name = ""
description = ""
lu_name = LUBEA01
lu_session_limit = 0
pu_name = <0000000000000000>
nau_address = 0
default_pool = NO
syncpt_support = YES
lu_attributes = NONE
```

5 *Configuring WebLogic JAM Connectivity*

```
sscp_id = 0
disable = NO
sys_name = ""
timeout = 60
back_level = NO
```

```
[define_model]
mode_name = SMSNA100
description = ""
max_neg_sess_lim = 32767
plu_mode_session_limit = 12
min_conwin_src = 5
min_conloser_src = 5
auto_act = 5
receive_pacing_win = 4
max_receive_pacing_win = 0
default_ru_size = YES
max_ru_size_upp = 1024
max_ru_size_low = 0
cos_name = #CONNECT
```

```
[define_directory_entry]
resource_name = P390.MVSREGN
resource_type = ENCP_RESOURCE
description = (Auto defined - remote node)
parent_name = <00000000000000000000000000000000>
parent_type = ENCP_RESOURCE
```

```
[define_directory_entry]
resource_name = P390.CICSREGN
resource_type = LU_RESOURCE
description = ""
parent_name = P390.MVSREGN
parent_type = ENCP_RESOURCE
```

```
[define_defaults]
description = ""
mode_name = SMSNA100
implicit_plu_forbidden = NO
specific_security_codes = NO
limited_timeout = 20
```

The fields that are most relevant to WebLogic JAM configuration and the VTAM SWNET definitions are listed in the table below.

Table 5-3 SNA Node Configuration Fields

Component Name	Field Within Component	Description
[define_node]	fqcp_name=BEALAN.BEASUN1	Fully Qualified Control Point name. Must match the NETID and CPNAME as specified in the VTAM SWNET definition.
[define_ethernet_ls]	ls_name=SWNEBEA1	Link Station name. Must match the PU name as specified in the VTAM SWNET definition.
[define_partner_lu]	plu_alias=CICSREGN	Partner Logical Unit name. Name of CICS region or IMS region that will communicate with your applications. This is used in the logical unit field in a Batch, CICS, or IMS region definition in the WebLogic Administration Console.
[define_local_lu]	lu_alias=LUBEA01	Logical Unit name. Name of the logical unit specified in your JAM CRM definition in the WebLogic Administration Console, in your session definitions in your CICS region, and in your VTAM SWNET definition.
[define_local_lu]	syncpt_support=YES	Syncpoint Support. Must be set to Yes to enable transactions to work.

Component Name	Field Within Component	Description
[define_mode]	mode_name_SMSNA100	Mode name. This is used in the mode name field in a CRM link definition in the WebLogic Administration Console.

6 CRM Administration

The following sections describe the types of tracing that can be set and the corresponding functionality.

APPC API Tracing

The DCL SNAP-IX APPC API may be captured by enabling the APPC API tracing. The API trace shows the parameters and values passed and returned to the DCL SNAP-IX stack.

Setting APPC API Tracing

The CRM APPC API trace option can be enabled by using the -s option on the command line. For example, the command:

```
CRM -s //myhost:3838 CRMB
```

Would start the CRM with APPC API tracing enabled. If the -s option is not set or the APPC API trace feature is disabled.

APPC API tracing can be dynamically enabled or disabled after the CRM process is started by using the WebLogic Administration Console. From the CRM configuration page, select 'enabled' or 'disabled' for APPC trace, and click Apply.

Directing the APPC Trace Output

The APPC Trace output can be directed to a file by setting the environment variable `SNATRC` as follows:

```
export SNATRC="filename::"
```

This will direct the APPC trace output to `$APPCDIR/filename`. If the `SNATRC` environment variable is not set and the CRM is being run in console mode, the APPC trace output is output to the CRM console. If the `SNATRC` environment variable is not set and the CRM is being run in the background, the APPC trace output is output to `stdout`.

A BEA WebLogic JAM Samples Verification and Installation Files

The following files for the installation verification sample are installed with WebLogic JAM. All files are relative to the specified installation directory.

Installation Verification and Samples Files

The following files are installed for the Solaris 7 or 8 platform.

Directory	Files
config/verify	setVerifyEnv.sh startVerifyServer.sh config.xml jamconfig_CICS.xml jamconfig_IMS.xml SerializedSystemIni.dat fileRealm.properties ca.pem democert.pem demokey.pem clientclasses/verify/gateway/outbound/BaseClient.class clientclasses/verify/gateway/outbound/Chardata.class clientclasses/verify/gateway/outbound/Client.class applications/jam.ear applications/DefaultWebApp/Web-inf/web.xml
config/examples/	ca.pem democert.pem demokey.pem

Directory	Files
	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
	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 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/CICS/outbound/gateway/ BaseClient.class clientclasses/examples/transactional/CICS/outbound/gateway/ Client.class

Directory	Files
	<code>clientclasses/examples/CICS/outbound/gateway/ EmployeeRecord\$EmpRecord1V\$EmpAddr7V.class</code> <code>clientclasses/examples/CICS/outbound/gateway/ EmployeeRecord\$EmpRecord1V\$EmpName3V.class</code> <code>clientclasses/examples/CICS/outbound/gateway/ EmployeeRecord\$EmpRecord1V.class</code> <code>clientclasses/examples/CICS/outbound/gateway/ EmployeeRecord.class</code> <code>clientclasses/examples/CICS/outbound/gateway/ BaseClient.class</code> <code>clientclasses/examples/CICS/outbound/gateway/Client.class</code>
	<code>clientclasses/dataviews/examples/IMS/inbound/gateway/ Chardata.class</code>
	<code>applications/jam.ear</code> <code>applications/JAM_TradeServer.jar</code> <code>applications/JAM_ToupperServer.jar</code> <code>applications/DefaultWebApp/Web-inf/web.xml</code>
<code>samples/crm/</code>	<code>uss/crm.env</code> <code>uss/startcrm.sh</code>

Directory	Files
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 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

Directory	Files
samples/examples/	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 IMS/inbound/mainframe/source/COMPIMSC IMS/inbound/mainframe/source/DFS62DTI IMS/inbound/mainframe/source/IMSINDEF IMS/inbound/mainframe/source/IMSTOJMS IMS/inbound/gateway/TopicReceive.java IMS/inbound/gateway/chardata.cpy IMS/inbound/gateway/chardata.egen IMS/inbound/gateway/Chardata.java IMS/inbound/gateway/build.sh

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

Directory	Files
	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 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/CICS/outbound/gateway/EmployeeRecord.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/emprec.cpy transactional/CICS/outbound/gateway/emprec.egen

Directory	Files
	CICS/outbound/gateway/emprec.cpy
	CICS/outbound/gateway/emprec.egen
	CICS/outbound/gateway/EmployeeRecord.java
	CICS/outbound/gateway/baseClient.egen
	CICS/outbound/gateway/BaseClient.java
	CICS/outbound/gateway/Client.java
	CICS/outbound/gateway/build.sh
	CICS/outbound/mainframe/source/CMPPROC
	CICS/outbound/mainframe/source/COMPCRUND
	CICS/outbound/mainframe/source/CSDUCRUND
	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
	CICS/inbound/gateway/ejb-jar.xml
	CICS/inbound/gateway/weblogic-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.java
	CICS/inbound/gateway/TradeServerBean.java
	CICS/inbound/gateway/TradeServerHome.java
	CICS/inbound/gateway/build.sh
	CICS/inbound/gateway/tradeserver.egen
	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
	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

Directory	Files
	wlintegration/create.xml
	wlintegration/read.xml
	wlintegration/readString.xml

