# BEA WebLogic Adapter for CORBA®

## Installation and Configuration Guide

Version 8.1.1
Document Revised: Februrary 2004

# Copyright

Copyright © 2004 BEA Systems, Inc.. All Rights Reserved.

Portions Copyright © 2004 iWay Software. 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 Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

# Contents

## About This Document

## 1. Installing the BEA WebLogic Adapter for CORBA

# A. Using CORBA Implementations With the Adapter

# Index

# About This Document

This document explains how to install and configure the BEA WebLogic Adapter for CORBA. This document is organized as follows:

- Chapter 1, "Installing the BEA WebLogic Adapter for CORBA," explains how to install the adapter.
- Appendix A, "Using CORBA Implementations With the Adapter," provides details about using the BEA WebLogic Adapter for CORBA with JacORB, Orbix2000, and VisiBroker for Java.

## Who Should Read This Documentation

This document is intended for the following members of an integration team:

- Integration Specialists—Lead the integration design effort. Integration specialists have expertise in defining the business and technical requirements of integration projects, and in designing integration solutions that implement specific features of WebLogic Integration. The skills of integration specialists include business and technical analysis, architecture design, project management, and WebLogic Integration product knowledge.
- Technical Analysts—Provide expertise in an organization's information technology infrastructure, including telecommunications, operating systems, applications, data repositories, future technologies, and IT organizations. The skills of technical analysts include technical analysis, application design, and information systems knowledge.

- Enterprise Information System (EIS) Specialists—Provide domain expertise in the systems that are being integrated using WebLogic Integration adapters. The skills of EIS specialists include technical analysis and application integration design.
- System Administrators—Provide in-depth technical and operational knowledge about databases and applications deployed in an organization. The skills of system administrators include capacity and load analysis, performance analysis and tuning, deployment topologies, and support planning.

## What You Need to Know

This document assumes that you have an understanding of:

- Web technologies
- WebLogic Integration
- CORBA software, system, and environment. This includes understanding of CORBA tools and configuration as well as processes and data models.
- Your specific CORBA business needs and applications.

## Product Documentation on the dev2dev Web Site

BEA product documentation, along with other information about BEA software, is available from the BEA dev2dev Web site:

`http://dev2dev.bea.com`

To view the documentation for a particular product, select that product from the list on the dev2dev page; the home page for the specified product is displayed. From the menu on the left side of the screen, select Documentation for the appropriate release. The home page for the complete documentation set for the product and release you have selected is displayed.

## Related Information

Readers of this document may find the following documentation and resources especially useful:

- *BEA WebLogic Adapter for CORBA Release Notes*

  `http://edocs.bea.com/wladapters/corba/docs811/pdf/relnotes.pdf`
- *BEA Application Explorer Installation and Configuration Guide*

  `http://edocs.bea.com/wladapters/bae/docs811/pdf/install.pdf`

- *Introduction to the BEA WebLogic Adapters*

  `http://edocs.bea.com/wladapters/docs81/pdf/intro.pdf`

- BEA WebLogic Adapters 8.1 Dev2Dev Product Documentation

  `http://dev2dev.bea.com/products/wladapters/index.jsp`

- Application Integration documentation

  `http://edocs.bea.com/wli/docs81/aiover/index.html`

  `http://edocs.bea.com/wli/docs81/aiuser/index.html`

- BEA WebLogic Integration documentation

  `http://edocs.bea.com/wli/docs81/index.html`

- BEA WebLogic Platform documentation

  `http://edocs.bea.com/platform/docs81/index.html`

# Contact Us!

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

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Adapter for CORBA and the version of the documentation.

If you have any questions about this version of BEA WebLogic Adapter for CORBA, or if you have problems using the BEA WebLogic Adapter for CORBA, contact BEA Customer Support through BEA WebSUPPORT at www.bea.com. You can also contact Customer Support by using the contact information provided on the Customer Support Card which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

CHAPTER 1

# Installing the BEA WebLogic Adapter for CORBA

This section explains how to install the BEA WebLogic Adapter for CORBA with WebLogic Integration on both Windows and UNIX systems.

This section is organized as follows:

- Preparing to Install the Adapter
- Installing the Adapter
- Next Steps

# Preparing to Install the Adapter

Before you install the BEA WebLogic Adapter for CORBA, be sure to complete the following tasks:

- Review the Release Notes
- Setting up Your ORB Environment
- Understanding the Representation of Paths

## Review the Release Notes

The *BEA WebLogic Adapter for CORBA Release Notes* contain important information about the software you must install prior to installing the BEA WebLogic Adapter for CORBA. Also, be sure to check the release notes for information about any required patches for your system. The *BEA WebLogic Adapter for CORBA Release Notes* are available at the following URL:

`http://edocs.bea.com/wladapters/corba/docs811/pdf/relnotes.pdf`

## Setting up Your ORB Environment

Before you install and configure the BEA WebLogic Adapter for CORBA, ensure that your ORB environment is configured to work with the adapter.

Your ORB environment is properly configured when:

- Your ORB infrastructure is properly configured
- Your server is reguistered in the Naming Service (NS), or you have an Interoperable Object reference (IOR) file for your server
- Your interface repository (IFR) is running and populated

To learn more about setting up your ORB environment, see Appendix A, "Using CORBA Implementations With the Adapter."

## Understanding the Representation of Paths

When you install WebLogic Integration, you specify the locations for files. Some of these files are required by the adapter. This document uses the following conventions to represent the locations of these files.

- *BEA_HOME* represents the BEA Home directory of your WebLogic installation. For example:
  - If you install the product in the default location on a Windows system, *BEA_HOME* represents `c:\bea`.
  - If you install the product in the default location on a UNIX system, *BEA_HOME* represents `/bea`.
- *WLI_HOME* represents the root of your WebLogic Integration installation. For example:
  - If you install WebLogic Integration in the default location on a Windows system, *WLI_HOME* represents `c:\bea\weblogic81\integration`.
  - If you install WebLogic Integration in the default location on a UNIX system, *WLI_HOME* represents `/bea/weblogic81/integration`.
- *domain* is used to indicate the name of a domain.

  You use the Configuration Wizard to create custom user domains. When you set up the domain configuration with the Configuration Wizard, you must specify a domain name, *domain*. You must also indicate where the directory associated with this domain is created. This directory contains files required for that domain. To learn more about the Configuration Wizard, see *Creating WebLogic Configurations Using the Configuration Wizard*, which is available at the following URL:

  http://edocs.bea.com/platform/docs81/confgwiz/index.html

  For example, if you accept the Configuration Wizard default location on a Windows system, *BEA_HOME*`\user_projects`, the files required by the administration server are installed in the following directory:

  *BEA_HOME*`\user_projects\`*domain*

  If you accept the Configuration Wizard default location on a UNIX system, *BEA_HOME*`/user_projects`, the files required by the administration server are installed in the following directory:

  *BEA_HOME*`/user_projects/`*domain*

  *DOMAIN_HOME* represents the complete path to the root of a domain.

  For example, if you use the Configuration Wizard to create a domain in the default location on a Windows system, *DOMAIN_HOME* represents `c:\bea\weblogic81\user_projects\`*domain*.

  If you use the Configuration Wizard to create a domain in the default location on a UNIX system, *DOMAIN_HOME* represents `/bea/weblogic81/user_projects/`*domain*.

**Note:** *WLI_HOME* and *BEA_HOME* (italicized) also represent the corresponding Windows and UNIX environment variables. For example, the literal interpretation of *WLI_HOME* is %WLI_HOME% for Windows and $WLI_HOME for UNIX.

Unlike *WLI_HOME* and *BEA_HOME*, *DOMAIN_HOME* is not an environment variable that is set by default in the WebLogic Integration environment.

# Installing the Adapter

This section explains how to install the BEA WebLogic Adapter for CORBA with WebLogic Integration. It includes the following steps:

- Step 1. Obtain the BEA WebLogic Adapter for CORBA
- Step 2. Configure the Domain
- Step 3. Extract the Adapter Files and Change the WebLogic Script
- Step 4. Update the BEA License
- Step 5. Deploy the Adapter
- Step 6. Create an Adapter Administrative User

## Step 1. Obtain the BEA WebLogic Adapter for CORBA

You need to obtain two files for the BEA WebLogic Adapter for CORBA, the EAR file for the adapter and a zip file containing samples.

### Obtaining the EAR file

To obtain the EAR file containing the BEA WebLogic Adapter for CORBA software (BEA_CORBA_8_1.ear), do one of the following:

- Download the file from the following URL:

  http://commerce.bea.com/products/weblogicadapters/wl_adapter_home.jsp

- Obtain the software on CD.

### Obtaining the Samples

You also must obtain the samples file for the CORBA adapter. This samples file includes JacORB components and sample service schemas.

To obtain the file containing the BEA WebLogic Adapter for CORBA samples (`BEA_CORBA_SAMPLES.zip`), do one of the following:

- Download the file from the following URL:

  `http://commerce.bea.com/products/weblogicadapters/wl_adapter_home.jsp`

- Obtain the software on CD.

You can use the samples after the adapter is installed. To learn more about using the samples, see the *BEA WebLogic Adapter for CORBA User Guide*.

## Step 2. Configure the Domain

You must deploy the BEA WebLogic Adapter for CORBA in a domain that supports application integration functionality. You can create one of your own, or use the sample integration domain. The sample integration domain is:

- On Windows: *`BEA_HOME`*`\weblogic81\samples\domains\integration`
- On UNIX: *`BEA_HOME`*`/weblogic81/samples/domains/integration`

If you have not already done so, use the Configuration Wizard to create the domain using the Integration domain template.

To learn more about the Configuration Wizard, see *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL:

`http://edocs.bea.com/platform/docs81/confgwiz/tempref.html`

## Step 3. Extract the Adapter Files and Change the WebLogic Script

This section explains how to extract the BEA WebLogic Adapter for CORBA files and edit your WebLogic script to add jar files to the CLASSPATH, and adjust the PATH.

**Note:** For WebLogic Integration 8.1 SP2, the script file you edit is `setDomainEnv.cmd` or `setDomainEnv.sh`. For versions of WebLogic Integration, earlier than 8.1 SP2, the file name is `startWeblogic.cmd` or `startWeblogic.sh`.

Set the classpath using the procedure appropriate for your system:

- Extracting Files and Adjusting the Classpath for Windows Using Orbix and WebLogic Integration 8.1 SP2

## Extracting Files and Adjusting the Classpath for Windows Using Orbix and WebLogic Integration 8.1 SP2

To extract the BEA WebLogic Adapter for CORBA files and edit the WebLogic script:

1. Use WinZip (or another similar extracting product) to extract the `BEA_CORBA_8_1.ear` file to a directory of your choice (for example, *`BEA_HOME`*`\adapters\corba`).

2. Go to the root directory for your domain:

   ```
   cd DOMAIN_HOME
   ```

   **Note:** You must choose a domain that supports application integration functionality.

3. Open the `setDomainEnv.cmd` script file with an ASCII editor.

4. Add the `ORBImpl` and `ORBSingleton` classes. In the `setDomainEnv.cmd` file, find the following command:

   ```
   JAVA_OPTIONS="${JAVA_OPTIONS} ${JAVA_PROPERTIES}
   -Dweblogic.security.SSL.ignoreHostnameVerify=false
   -Dwlw.iterativeDev=true
   -Dwlw.testConsole=true
   -Dwlw.logErrorsToConsole=${logErrorsToConsoleFlag}"
   ```

5. Append the following lines to the `JAVA_OPTIONS` command:

```
-Dorg.omg.CORBA.ORBCLASS=com.iona.corba.art.artimpl.ORBImpl
-Dorg.omg.CORBA.ORBSingletonClass=com.iona.corba.art.artimpl.ORBSingleton
```

6. Create a variable called `IONAPATH` that contains the path to your `asp-corba.jar`, the path to your orbix domain, and the classpath:

```
set
IONAPATH=ORBIX_HOME\asp\6.0\lib\asp-corba.jar;ORBIX_HOME\etc\domains\
orb-domain-home;%CLASSPATH%
```

**Note:** In these instructions, *`ORBIX_HOME`* is home directory of your Orbix installation and *`orb-domain-home`* is the Orbix domain home. This set command is all one line, there are no carriage returns.

7. Add the `art.jar` file and the `IONAPATH` variable to your CLASSPATH:

```
CLASSPATH=%PRE_CLASSPATH%;ORBIX_HOME\lib\art\art\5\art.jar;%WEBLOGIC_CL
ASSPATH%;%IONAPATH%;%POST_CLASSPATH%;%WLP_POST_CLASSPATH%
```

8. Save your changes and close the script file.

9. Go to the Orbix domain home. Copy the Orbix domain configuration file `neworb-domain.cfg` to a file named `bea.cfg` in the same directory.

   The adapter requires that this file be named `bea.cfg`.

## Extracting Files and Adjusting the Classpath for Windows Using Orbix and WebLogic Integration 8.1

These instructions apply to all versions of WebLogic Integration earlier than 8.1 SP2 that are supported by this adapter.

To extract the BEA WebLogic Adapter for CORBA files and edit the WebLogic script:

1. Use WinZip (or another similar extracting product) to extract the `BEA_CORBA_8_1.ear` file to a directory of your choice (for example, *`BEA_HOME`*`\adapters\corba`).

2. Go to the root directory for your domain:

```
cd DOMAIN_HOME
```

   **Note:** You must choose a domain that supports application integration functionality.

3. Open the `startWebLogic.cmd` script file with an ASCII editor.

4. Find the following command in the script file:

```
set CLASSPATH=%ARDIR%\ant\ant.jar;%JAVA_HOME%\jre\lib\rt.jar
```

5. Directly after this line, add commands to specify the location of the ORB application and interface repository file.

**So, what is the location for the Orbix app and interface repository file?**

   For JacORB, add the following commands:

```
set JACORB_HOME=JacORB_installation_directory_path
```

```
set CLASSPATH=%JACORB_HOME%\CorbaAppDirectory;%JACORB_HOME%
\jacorb.properties;%JACORB_HOME%\lib\jacorb.jar;%JACORB_HOME%
\lib\idl.jar;%CLASSPATH%
```

```
set PATH=%JACORB_HOME%\CorbaAppDirectory;%PATH%
```

For VisiBroker, add the following commands:

```
set VBORB_HOME=VisiBroker_installation_directory_path
```

```
set CLASSPATH=%CLASSPATH%;%VBORB_HOME%\lib\vbjorb.jar;%VBORB_HOME%
\lib\idl;%VBORB_HOME%\CorbaAppDirectory;
```

**Note:** For path requirements for other ORBs, see your ORB documentation.

6. Find the following command in the script file:

```
set CLASSPATH=%ARDIR%\ant\ant.jar;%JAVA_HOME%\jre\lib\rt.jar
```

```
%JAVA_HOME%\bin\java %JAVA_VM%
```

7. Immediately *after* this command line, insert the following command lines so that the `CLASSPATH` and `PATH` point to the files you extracted in Step 1:

```
rem ==== Adding corba win32 directory to PATH =====
```

```
set PATH=%PATH%;BEA_HOME\adapters\corba\win32
```

8. Add the following option to the command to specify the location of the ORB jar file:

```
-Xbootclasspath/p:ORBIX_HOME\asp\6.0\lib\asp-corba.jar
```

**Note:** Here, *`ORBIX_HOME`* is home directory of your Orbix installation.

9. Save your changes and close the script file.

10. Go to the Orbix domain home. Copy the Orbix domain configuration file `neworb-domain.cfg` to a file named `bea.cfg` in the same directory.

The adapter requires that this file be named `bea.cfg`.

## Extracting Files and Adjusting the Classpath for Windows Using Other ORBs

These instructions apply to all versions of WebLogic Integration supported by this adapter.

To extract the BEA WebLogic Adapter for CORBA files and edit the WebLogic script:

1. Use WinZip (or another similar extracting product) to extract the `BEA_CORBA_8_1.ear` file to a directory of your choice (for example, *`BEA_HOME`*`\adapters\corba`).

2. Go to the root directory for your domain:

```
cd DOMAIN_HOME
```

**Note:** You must choose a domain that supports application integration functionality.

3. Open the script file with an ASCII editor. For WebLogic Integration 8.1 SP2, the file is `setDomainEnv.cmd`. For earlier versions of WebLogic Integration, the file is `startWebLogic.cmd`.

4. Find the following command in the script file:

```
set CLASSPATH=%ARDIR%\ant\ant.jar;%JAVA_HOME%\jre\lib\rt.jar
```

5. Directly after this line, add commands to specify the location of the ORB application and interface repository file.

   For JacORB, add the following commands:

```
set JACORB_HOME=JacORB_installation_directory_path

set CLASSPATH=%JACORB_HOME%\CorbaAppDirectory;%JACORB_HOME%
\jacorb.properties;%JACORB_HOME%\lib\jacorb.jar;%JACORB_HOME%
\lib\idl.jar;%CLASSPATH%

set PATH=%JACORB_HOME%\CorbaAppDirectory;%PATH%
```

   For VisiBroker, add the following commands:

```
set VBORB_HOME=VisiBroker_installation_directory_path

set CLASSPATH=%CLASSPATH%;%VBORB_HOME%\lib\vbjorb.jar;%VBORB_HOME%
\lib\idl;%VBORB_HOME%\CorbaAppDirectory;
```

   **Note:** For path requirements for other ORBs, see your ORB documentation.

6. Find the following command in the script file:

```
set CLASSPATH=%ARDIR%\ant\ant.jar;%JAVA_HOME%\jre\lib\rt.jar

%JAVA_HOME%\bin\java %JAVA_VM%
```

7. Immediately *after* this command line, insert the following command lines so that the `CLASSPATH` and `PATH` point to the files you extracted in Step 1:

```
rem ==== Adding corba win32 directory to PATH =====

set PATH=%PATH%;BEA_HOME\adapters\corba\win32
```

8. Add the following option to the command to specify the location of the ORB jar file:

   For Visibroker:

```
-Xbootclasspath/a:x:\Path\Filename.jar
```

   For other ORBs:

```
-Xbootclasspath/p:x:\Path\Filename.jar
```

For example, to identify the location of a JacORB jar file the command should resemble the following:

```
%JAVA_HOME%\bin\java %JAVA_VM%
-Xbootclasspath/p:x:\JacORB1_4_1\lib\jacorb.jar %JAVA_OPTIONS% -Xmx256m
-classpath %SVRCP% -Dbea.home=%BEA_HOME% -Dweblogic.home=%WL_HOME%
-Dweblogic.system.home=D:\bea\user_projects\eaidomain
-Dwli.bpm.server.evaluator.supportsNull=false
-Dweblogic.management.username=
-Dweblogic.management.password= -Dweblogic.Name=myserver
-Dweblogic.RootDirectory=D:\bea\user_projects\eaidomain
-Djava.security.policy=%WL_HOME%\lib\weblogic.policy
-Dweblogic.management.discover=true
-Dweblogic.servlet.ClasspathServlet.disableStrictCheck=true
weblogic.Server
```

**Note:** These instructions are for access to JacORB services. For configuration details for other ORBs, see your ORB documentation.

9. Save your changes and close the script file.

## Extracting Files and Adjusting the Classpath for UNIX Using Orbix and WebLogic Integration 8.1 SP2

To extract the BEA WebLogic Adapter for CORBA files and edit the WebLogic script:

1. Use jar (or another similar extracting product) to extract `BEA_CORBA_8_1.ear` to a directory of your choice (for example, *`BEA_HOME`*`/adapters/corba`).

2. Go to the root directory for your domain:

   ```
   cd DOMAIN_HOME
   ```

   **Note:** You must choose a WebLogic Integration domain that supports application integration functionality.

3. Open the `setDomainEnv.sh` script file with an editor.

4. Add the `ORBImpl` and `ORBSingleton` classes. In the `setDomainEnv.sh` file, find the following command:

   ```
   JAVA_OPTIONS="${JAVA_OPTIONS} ${JAVA_PROPERTIES}
   -Dweblogic.security.SSL.ignoreHostnameVerify=false
   -Dwlw.iterativeDev=true
   -Dwlw.testConsole=true
   -Dwlw.logErrorsToConsole=${logErrorsToConsoleFlag}"
   ```

5. Append the following lines to the `JAVA_OPTIONS` command:

```
-Dorg.omg.CORBA.ORBCLASS=com.iona.corba.art.artimpl.ORBImpl
-Dorg.omg.CORBA.ORBSingletonClass=com.iona.corba.art.artimpl.ORBSingleton
```

6. Create a variable called `IONAPATH` that contains the path to your `asp-corba.jar`, the path to your orbix domain, and the classpath:

   ```
   IONAPATH=ORBIX_HOME/asp/6.0/lib/asp-corba.jar:ORBIX_HOME/domains/
   orb-domain-home:$CLASSPATH
   ```

   **Note:** In these instructions, *`ORBIX_HOME`* is home directory of your Orbix installation and *`orb-domain-home`* is the Orbix domain home. This command is all one line.

7. Add the `art.jar` file and the `IONAPATH` variable to your CLASSPATH:

   ```
   CLASSPATH="${PRE_CLASSPATH}:ORBIX_HOME/lib/art/art/5/art.jar:${WEBLOGIC
   _CLASSPATH}:${IONAPATH}:${POST_CLASSPATH}:${WLP_POST_CLASSPATH}"
   ```

8. Save your changes and close the script file.

9. Go to the Orbix domain home. Copy the Orbix domain configuration file `neworb-domain.cfg` to a file named `bea.cfg` in the same directory.

   The adapter requires that this file be named `bea.cfg`.

10. Go to the Orbix domain home. Copy the Orbix domain configuration file `neworb-domain.cfg` to a file named `bea.cfg` in the same directory.

    The adapter requires that this file be named `bea.cfg`.

## Extracting Files and Adjusting the Classpath for UNIX Using Orbix and WebLogic Integration 8.1

These instructions apply to all versions of WebLogic Integration earlier than 8.1 SP2 that are supported by this adapter.

To extract the BEA WebLogic Adapter for CORBA files and edit the WebLogic script:

1. Use jar (or another similar extracting product) to extract `BEA_CORBA_8_1.ear` to a directory of your choice (for example, *`BEA_HOME`*`/adapters/corba`).

2. Go to the root directory for your domain:

   ```
   cd DOMAIN_HOME
   ```

   **Note:** You must choose a WebLogic Integration domain that supports application integration functionality.

3. Open the `startWebLogic.sh` script file with an editor.
4. Find the following command in the script file:

   ```
   CLASSPATH="${ARDIR}/ant/ant.jar:${JAVA_HOME}/jre/lib/rt.jar"
   ```

5. Directly after this line, add commands to specify the location of the ORB application and interface repository file.

**So, what is the location for the Orbix app and interface repository file?**

   For JacORB, add the following commands:

   ```
   JACORB_HOME=JacORB_installation_directory_path

   CLASSPATH="$JACORB_HOME/jacorb.properties:$JACORB_HOME/
   CorbaAppDirectory:$JACORB_HOME/lib/jacorb.jar:$JACORB_HOME/
   classes:$CLASSPATH"

   PATH="$JACORB_HOME/bea:$PATH"LD_LIBRARY_PATH="/opt/JacORB_1_4_1/
   jacorb.properties"

   export LD_LIBRARY_PATH
   ```

   For VisiBroker, add the following commands:

   ```
   VBORB_HOME=VisiBroker_installation_directory_path

   CLASSPATH="$VBORB_HOME/CorbaAppDirectory:$VBORB_HOME:$VBORB_HOME/lib/vb
   jorb.jar:$CLASSPATH"

   PATH="$VBORB_HOME/CorbaAppDirectory:$PATH"
   ```

   **Note:** For path requirements for other ORBs, see your ORB documentation.

6. Locate the following Java command in the script file:

   ```
   ${JAVA_HOME}/bin/java ${JAVA_VM}
   ```

7. Add the following option to the command to specify the location of the ORB jar file:

   ```
   -Xbootclasspath/p:ORBIX_HOME/asp/6.0/lib/asp-corba.jar ${MEM_ARGS}
   ```

   **Note:** In these instructions, *`ORBIX_HOME`* is home directory of your Orbix installation and *`orb-domain-home`* is the Orbix domain home. This command is all one line.

   For example, to identify the location of a JacORB jar file the command should resemble the following:

   ```
   JAVA_HOME/bin/java $JAVA_VM -Xbootclasspath/p:$AE_HOME/lib/jacorb.jar
   $JAVA_OPTIONS -classpath $SVRCP -Dbea.home=$BEA_HOME
   -Dweblogic.home=$WL_HOME -Dweblogic.system.home=/qa/edantk/bea/user/abc
   -Dwli.bpm.server.evaluator.supportsNull=false
   -Dweblogic.management.username=
   ```

```
-Dweblogic.management.password= -Dweblogic.Name=myserver
-Dweblogic.RootDirectory=/qa/edantk/bea/user/abc
-Djava.security.policy=$WL_HOME/lib/weblogic.policy
-Dweblogic.management.discover=true
-Dweblogic.servlet.ClasspathServlet.disableStrictCheck=true
weblogic.Server
```

**Note:** These instructions are for configuring access to JacORB services. For details for other ORBs, see your ORB documentation.

8. Save your changes and close the script file.
9. Go to the Orbix domain home. Copy the Orbix domain configuration file `neworb-domain.cfg` to a file named `bea.cfg` in the same directory.

   The adapter requires that this file be named `bea.cfg`.

## Extracting Files and Adjusting the Classpath for UNIX Using Other ORBs

These instructions apply to all versions of WebLogic Integration supported by this adapter.

To extract the BEA WebLogic Adapter for CORBA files and edit the WebLogic script:

1. Use jar (or another similar extracting product) to extract `BEA_CORBA_8_1.ear` to a directory of your choice (for example, *`BEA_HOME`*`/adapters/corba`).
2. Go to the root directory for your domain:

   ```
   cd DOMAIN_HOME
   ```

   **Note:** You must choose a WebLogic Integration domain that supports application integration functionality.
3. Open the WebLogic script file with an editor. For WebLogic Integration 8.1 SP2, the file is `setDomainEnv.sh`. For earlier versions of WebLogic Integration, the file is `startWebLogic.sh`.
4. Find the following command in the script file:

   ```
   CLASSPATH="${ARDIR}/ant/ant.jar:${JAVA_HOME}/jre/lib/rt.jar"
   ```
5. Directly after this line, add commands to specify the location of the ORB application and interface repository file.

   For JacORB, add the following commands:

   ```
   JACORB_HOME=JacORB_installation_directory_path
   ```

```
CLASSPATH="$JACORB_HOME/jacorb.properties:$JACORB_HOME/
CorbaAppDirectory:$JACORB_HOME/lib/jacorb.jar:$JACORB_HOME/
classes:$CLASSPATH"
```

```
PATH="$JACORB_HOME/bea:$PATH"LD_LIBRARY_PATH="/opt/JacORB_1_4_1/
jacorb.properties"
```

```
export LD_LIBRARY_PATH
```

For VisiBroker, add the following commands:

```
VBORB_HOME=VisiBroker_installation_directory_path
```

```
CLASSPATH="$VBORB_HOME/CorbaAppDirectory:$VBORB_HOME:$VBORB_HOME/lib/vb
jorb.jar:$CLASSPATH"
```

```
PATH="$VBORB_HOME/CorbaAppDirectory:$PATH"
```

**Note:** For path requirements for other ORBs, see your ORB documentation.

6. Locate the following Java command in the script file:

```
JAVA_HOME/bin/java $JAVA_VM
```

7. Add the following option to the command to specify the location of the ORB jar file:

For Visibroker:

```
-Xbootclasspath/a:$AE_HOME/Path/Filename.jar
```

For other ORBs:

```
-Xbootclasspath/p:$AE_HOME/Path/Filename.jar
```

For example, to identify the location of a JacORB jar file the command should resemble the following:

```
JAVA_HOME/bin/java $JAVA_VM -Xbootclasspath/p:$AE_HOME/lib/jacorb.jar
$JAVA_OPTIONS -classpath $SVRCP -Dbea.home=$BEA_HOME
-Dweblogic.home=$WL_HOME -Dweblogic.system.home=/qa/edantk/bea/user/abc
-Dwli.bpm.server.evaluator.supportsNull=false
-Dweblogic.management.username=
-Dweblogic.management.password= -Dweblogic.Name=myserver
-Dweblogic.RootDirectory=/qa/edantk/bea/user/abc
-Djava.security.policy=$WL_HOME/lib/weblogic.policy
-Dweblogic.management.discover=true
-Dweblogic.servlet.ClasspathServlet.disableStrictCheck=true
weblogic.Server
```

**Note:** These instructions are for configuring access to JacORB services. For details for other ORBs, see your ORB documentation.

8. Save your changes and close the script file.

## Step 4. Update the BEA License

In order to use the BEA WebLogic Adapter for CORBA you must have a valid software license. If you have downloaded the adapter for evaluation, see the instructions on the adapter download page to obtain an evaluation license. If you have purchased a license for the adapter, you should receive the license file as an e-mail attachment. Once you have the license file for the adapter, you must update your `license.bea` file to include the new information for the adapter.

To update your `license.bea` file:

1. Save the adapter license file in the *`BEA_HOME`* directory. To avoid overwriting your `license.bea` file, use a name other than `license.bea`. For example, save the file as `corba_adapter_license.bea`. The adapter license file is the *`license_update_file`* referred to in step 4 of this procedure.

   **Warning:** Do not overwrite or change the name of the existing `license.bea` file.

2. Go to the *`BEA_HOME`* directory:
   - On a Windows system, open an MS-DOS session and go to the *`BEA_HOME`* directory.
   - On a UNIX system, go to the *`BEA_HOME`* directory.

3. Add the JDK to your `PATH` variable. If it is already included, skip to step 4.
   - On a Windows system:

     ```
     set PATH=BEA_HOME\jdk141_03\bin;%PATH%
     ```

   - On a UNIX system:

     ```
     PATH=BEA_HOME/jdk141_03/bin:$PATH
     export PATH
     ```

4. Merge the adapter license file into your existing license:
   - On a Windows system:

     ```
     UpdateLicense license_update_file
     ```

   - On a UNIX system:

     ```
     sh UpdateLicense.sh license_update_file
     ```

   Here, *`license_update_file`* is the name of the adapter license file you saved in step 1.

5. Save a backup copy of your updated `license.bea` file. This backup location should be a safe place that is neither the WebLogic Integration nor the application installation directories.

# Step 5. Deploy the Adapter

After you have installed the BEA WebLogic Adapter for CORBA, you must deploy it to your domain.

To deploy the adapter:

1. Start WebLogic Server in your domain.
2. Start the WebLogic Server Administration Console in a browser using the following URL:

   `http://`*`host:port`*`/console/`

   Where,

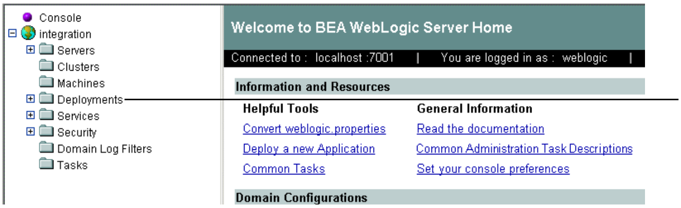
   – *`host`* represents the machine on which WebLogic Server is running

   – *`port`* represents the listening port.

   For example, `http://localhost:7001/console/`

3. Enter the user name and password for the server.

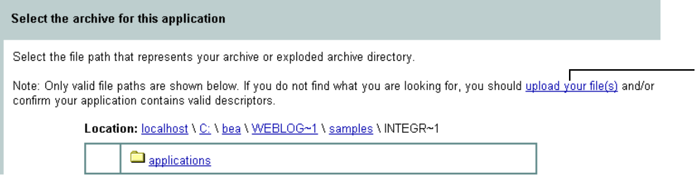
   The WebLogic Server Administration Console appears.


Console
integration
Servers
Clusters
Machines
Deployments
Services
Security
Domain Log Filters
Tasks
Welcome to BEA WebLogic Server Home
Connected to : localhost :7001 | You are logged in as : weblogic |
Information and Resources
Helpful Tools
General Information
Convert weblogic.properties
Read the documentation
Deploy a new Application
Common Administration Task Descriptions
Common Tasks
Set your console preferences
Domain Configurations

Click to expand the Deployments node.

4. In the left pane, expand the Deployments node.
5. Under the Deployments node, right-click Applications, and then select Deploy a New Application.

   The Administration Console displays the Deployment Assistant in the right panel.



Select the file path that represents your archive or exploded archive directory.

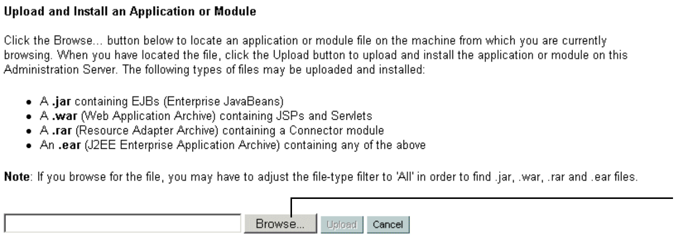
Note: Only valid file paths are shown below. If you do not find what you are looking for, you should upload your file(s) and/or confirm your application contains valid descriptors.

**Location:** localhost \ C: \ bea \ WEBLOG~1 \ samples \ INTEGR~1

Click to upload your files

6. Click the upload your files(s) link.

The Administration Console displays the Install or Update an Application window.



**Upload and Install an Application or Module**

Click the Browse... button below to locate an application or module file on the machine from which you are currently browsing. When you have located the file, click the Upload button to upload and install the application or module on this Administration Server. The following types of files may be uploaded and installed:

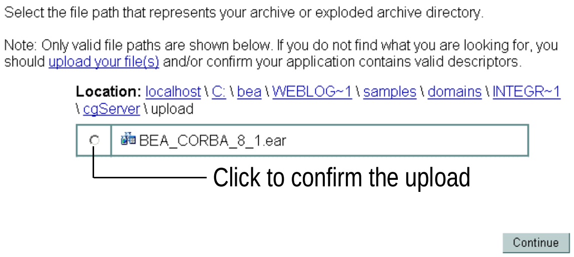
- A **.jar** containing EJBs (Enterprise JavaBeans)
- A **.war** (Web Application Archive) containing JSPs and Servlets
- A **.rar** (Resource Adapter Archive) containing a Connector module
- An **.ear** (J2EE Enterprise Application Archive) containing any of the above

**Note**: If you browse for the file, you may have to adjust the file-type filter to 'All' in order to find .jar, .war, .rar and .ear files.

Click to browse to the ear file's location.

7. Click Browse, navigate to the directory in which the `BEA_CORBA_8_1.ear` file resides, and then click Upload.

   The Administration Console prompts you to confirm the uploaded application.



Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should upload your file(s) and/or confirm your application contains valid descriptors.

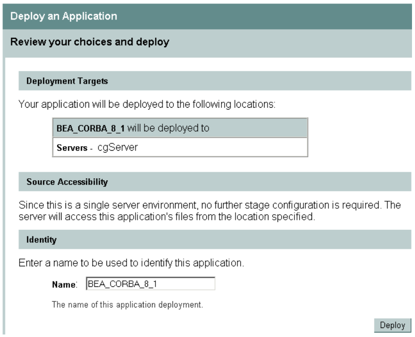
**Location:** localhost \ C: \ bea \ WEBLOG~1 \ samples \ domains \ INTEGR~1 \ cgServer \ upload

○ BEA_CORBA_8_1.ear

Click to confirm the upload

8. Click the radio button next to the application file and then click Continue.

   The Administration Console uploads the file and displays the Deploy an Application window (specifying the default target server).



Deploy an Application

**Review your choices and deploy**

**Deployment Targets**

Your application will be deployed to the following locations:

**BEA_CORBA_8_1** will be deployed to

**Servers** - cgServer

**Source Accessibility**

Since this is a single server environment, no further stage configuration is required. The server will access this application's files from the location specified.

**Identity**

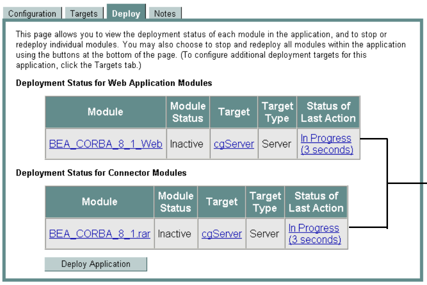
Enter a name to be used to identify this application.

**Name**: BEA_CORBA_8_1

The name of this application deployment.

9. Click Deploy.

   The Administration Console deploys the application and displays its deployment status.



This page allows you to view the deployment status of each module in the application, and to stop or redeploy individual modules. You may also choose to stop and redeploy all modules within the application using the buttons at the bottom of the page. (To configure additional deployment targets for this application, click the Targets tab.)

**Deployment Status for Web Application Modules**

| Module | Module Status | Target | Target Type | Status of Last Action |
|---|---|---|---|---|
| BEA_CORBA_8_1_Web | Inactive | cgServer | Server | In Progress (3 seconds) |

**Deployment Status for Connector Modules**

| Module | Module Status | Target | Target Type | Status of Last Action |
|---|---|---|---|---|
| BEA_CORBA_8_1.rar | Inactive | cgServer | Server | In Progress (3 seconds) |

When this status is success, you can go to the next step.

10. Click Deploy Application to activate the application, or wait until the Administration Console does so automatically.

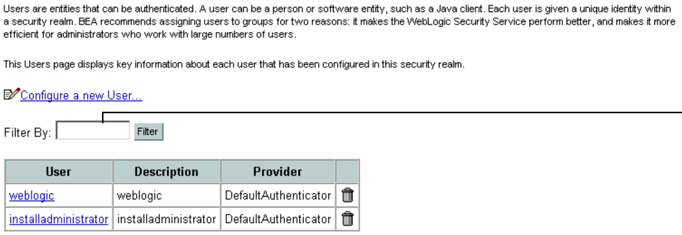
# Step 6. Create an Adapter Administrative User

If you want to manage security for the BEA WebLogic Adapter for CORBA, you can create an administrative user (such as `corbaAdapterAdmin`) who is authorized to log in to the Application View Console, create application views, configure services and events, deploy, and test. This adapter administrator also needs to be added to the `Administrators` group.

To create a new adapter administrative user:

1. In the left pane of theWebLogic Server Administration Console, click the Security node.
2. In the left pane, click the Realms node.
3. In the left pane, click the name of the realm for which you want to configure security.
4. In the left pane, click Users.

   The Users page appears.



Users are entities that can be authenticated. A user can be a person or software entity, such as a Java client. Each user is given a unique identity within a security realm. BEA recommends assigning users to groups for two reasons: it makes the WebLogic Security Service perform better, and makes it more efficient for administrators who work with large numbers of users.

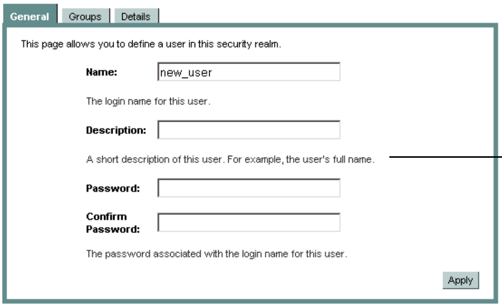
This Users page displays key information about each user that has been configured in this security realm.

| User | Description | Provider | |
|---|---|---|---|
| weblogic | weblogic | DefaultAuthenticator | 🗑 |
| installadministrator | installadministrator | DefaultAuthenticator | 🗑 |

Click to configure a new user.

5. Click the Configure a New User link.

   The Create User page appears.



This page allows you to define a user in this security realm.

**Name:** new_user

The login name for this user.

**Description:**

A short description of this user. For example, the user's full name.

**Password:**

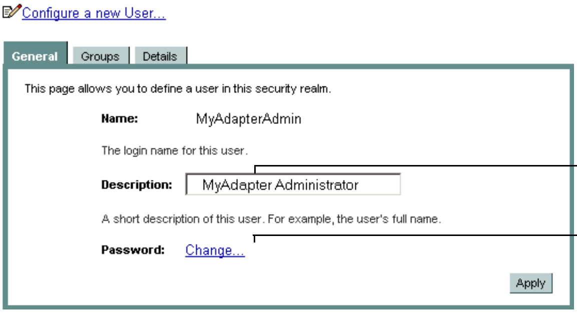
**Confirm Password:**

The password associated with the login name for this user.

Enter the user information.

6. Enter the user name, description and password, and then click Apply.

   The User page appears.



This page allows you to define a user in this security realm.

**Name:** MyAdapterAdmin

The login name for this user.

**Description:** MyAdapter Administrator

A short description of this user. For example, the user's full name.

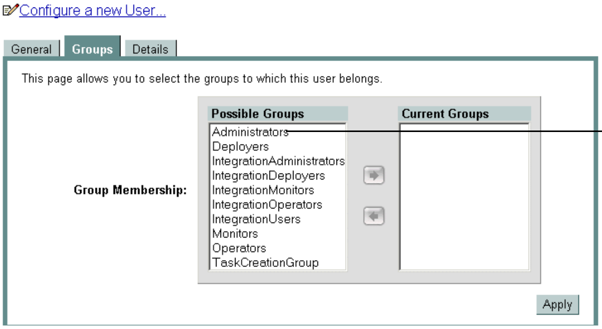
**Password:** Change...

Enter a description for this user.

Click to change the password.

7. Click the Groups tab.

   The Groups page appears.



This page allows you to select the groups to which this user belongs.

**Group Membership:**

| Possible Groups | Current Groups |
|---|---|
| Administrators<br>Deployers<br>IntegrationAdministrators<br>IntegrationDeployers<br>IntegrationMonitors<br>IntegrationOperators<br>IntegrationUsers<br>Monitors<br>Operators<br>TaskCreationGroup | |

Select a group for the user.

8. In the Possible Groups list, select Administrators and then click the right arrow to add the Administrators group to the list of current groups.

9. Click Apply.

10. In the left pane, click the Users node and confirm that the user you created appears in the list of users.

# Next Steps

After you have finished installing the BEA WebLogic Adapter for CORBA, you can proceed to the following tasks:

- Installing the BEA Application Explorer
- Configuring the BEA Application Explorer for Your ORB
- Starting Integration with CORBA

## Installing the BEA Application Explorer

To proceed, you must install the BEA Application Explorer. If you have not already done so, install it now. To learn more about installing the BEA Application Explorer, see the *BEA Application Explorer Installation and Configuration Guide* at the following URL:

`http://edocs.bea.com/wladapters/bae/docs811/pdf/install.pdf`

## Configuring the BEA Application Explorer for Your ORB

Once the BEA Application Explorer is installed, you must configure the BEA Application Explorer so that it can generate schemas for your ORB.

To configure the BEA Application Explorer:

1. Open the BEA Application Explorer command file:
   - On a Windows system, the default location of this `.bat` file is:

     `C:\Program Files\BEA Systems\BEA Application Explorer\bin\ae.bat`
   - On a UNIX system, the location of this shell script is `/`*`installation_directory`*`/bin/ae`

     For example: `/home/apps/bea/bae/bin/ae`
2. Copy the CORBA jar files for your ORB into the `C:\Program Files\BEA Systems\BEA Application Explorer\lib` directory.

**Table 1-1**

| Operating System and ORB | Files to Copy |
|---|---|
| Visibroker on Windows | `\Inprise\vbroker\lib\vbjorb.jar` |
| Visibroker on UNIX | `/Inprise/vbroker/lib/vbjorb.jar` |
| JacORB on Windows | `\JacORB1_4_beta4\lib\jacorb.jar` |
| JacORB on UNIX | `/JacORB1_4_beta4/lib/jacorb.jar` |
| ORBacus 4.1.1.2 on Windows | `\ORBacus4.1.1.2\OOC\lib\OB.jar` and<br>`\ORBacus4.1.1.2\OOC\lib\OBNaming.jar` |
| ORBacus 4.1.1.2 on UNIX | `/ORBacus4.1.1.2/OOC/lib/OB.jar` and<br>`/ORBacus4.1.1.2/OOC/lib/OBNaming.jar` |
| IONA ASP on Windows | `\IONA\lib\art\art\5\art.jar` and<br>`\IONA\asp\6.0\lib\asp-corba.jar` |
| IONA ASP on UNIX | `/IONA/lib/art/art/5/art.jar` and<br>`/IONA/asp/6.0/lib/asp-corba.jar` |

3. Uncomment the following statement:

   – On a Windows system using JacORB:

   ```
   REM "%JAVACMD%" -Xbootclasspath/p:"%AE_HOME%\lib\jacorb.jar"
   -classpath "%LOCALCLASSPATH%" com.ibi.bse.gui.BseFlashScreen
   ```

   – On a UNIX system using JacORB:

   ```
   #$JAVACMD -Xbootclasspath/p:$AE_HOME/lib/jacorb.jar -classpath
   $LOCALCLASSPATH com.ibi.bse.gui.BseFlashScreen $@
   ```

   – On a Windows system using Visibroker:

   ```
   REM "-Xbootclasspath/a:x:\VISIBROKER_HOME\vbjorb.jar" -classpath
   "%LOCALCLASSPATH%" com.ibi.bse.gui.BseFlashScreen
   ```

   – On a UNIX system using Visibroker:

   ```
   #$JAVACMD "-Xbootclasspath/a:x:/VISIBROKER_HOME/vbjorb.jar"
   -classpath $LOCALCLASSPATH com.ibi.bse.gui.BseFlashScreen $@
   ```

**Note:** *`VISIBROKER_HOME`* is the home directory for your Visibroker installation.

4. Comment the following statement:

   – On a Windows system:

   ```
   "%JAVACMD%" -classpath "%LOCALCLASSPATH%"
   com.ibi.bse.gui.BseFlashScreen
   ```

   – On a UNIX system:

   ```
   $JAVACMD -classpath $LOCALCLASSPATH com.ibi.bse.guil.BseFlashScreen
   $@
   ```

5. Add the following line to the BEA Application Explorer command file:

**Table 1-2**

| ORB and Operating System | Line to Add to the BEA Application Explorer Command File |
|---|---|
| Visibroker on Windows | `-Xbootclasspath/a:x:`*`drive`*`:\Inprise\vbroker\lib\vbjorb.jar` |
| Visibroker on UNIX | `-Xbootclasspath/a:x:/Inprise/vbroker/lib/vbjorb.jar` |
| JacORB on Windows | `-Xbootclasspath/a:x:`*`drive`*`:\JacORB1_4_beta4\lib\jacorb.jar` |
| JacORB on UNIX | `-Xbootclasspath/a:x:/JacORB1_4_beta4/lib/jacorb.jar` |
| ORBacus on Windows | `-Xbootclasspath/a:x:`*`drive`*`:\ORBacus4.1.1.2\OOC\OB.jar` |
| ORBacus on UNIX | `-Xbootclasspath/a:x:/ORBacus4.1.1.2/OOC/OB.jar` |
| IONA ASP on Windows | `-Xbootclasspath/a:x:`*`drive`*`:/IONA/lib/art/art/5/art.jar;`*`drive`*`:\IONA\asp\6.0\lib\asp-corba.jar` |
| IONA ASP on UNIX | `-Xbootclasspath/a:x:/IONA/lib/art/art/5/art.jar;/IONA/asp/6.0/lib/asp-corba.jar` |

6. Save your changes to the BEA Application Explorer command file.

7. Copy the ORB jar files to the `lib` directory under the BEA Application Explorer directory.

   – On a Windows system using Visibroker:

   copy *VISIBROKER_HOME*\lib\vbjorb.jar BAEHOME\lib

   – On a UNIX system using Visibroker:

```
cp VISIBROKER_HOME/lib/vbjorb.jar BAEHOME/lib
```

**Note:** *VISIBROKER_HOME* is the home directory for your Visibroker installation.

- On a Windows system using JacORB:

```
copy JACORB_HOME\lib\jacorb.jar BAEHOME\lib
```

- On a UNIX system using JacORB:

```
cp JACORB_HOME/lib/jacorb.jar BAEHOME/lib
```

**Note:** *JACORB_HOME* is the home directory for your JacORB installation.

- On a Windows system using Orbix:

```
copy ORBIX_HOME\lib\art\art\5\art.jar BAEHOME\lib
copy ORBIX_HOME\asp\6.0\lib\asp-corba.jar BAEHOME\lib
```

- On a UNIX system using Orbix:

```
cp ORBIX_HOME/lib/art/art/5/art.jar BAEHOME/lib
cp ORBIX_HOME/asp/60/lib/5/asp-corba.jar BAEHOME/lib
```

**Note:** *ORBIX_HOME* is the home directory for your Orbix installation.

You can now generate schemas for your ORB using the BEA Application Explorer.

## Starting Integration with CORBA

After you have successfully installed and deployed the BEA WebLogic Adapter for CORBA and the BEA Application Explorer, you can begin integrating with your CORBA system using the adapter and BEA WebLogic Integration. To learn more about integrating with CORBA, see the *BEA WebLogic Integration Adapter for CORBA User Guide* at the following URL:

http://edocs.bea.com/wladapters/corba/docs811/pdf/user.pdf

# Using CORBA Implementations With the Adapter

This section provides details about using the BEA WebLogic Adapter for CORBA with JacORB, Orbix2000, and VisiBroker for Java. It includes the following topics:

- Using JacORB
- Using Orbix2000
- Using VisiBroker for Java

## Using JacORB

The BEA WebLogic Adapter for CORBA includes a sample ORB called JacORB. JacORB is an open source Java implementation of the Object Management Group's CORBA specification. It is supplied with your adapter software to enable you to test the adapter.

JacORB is designed to comply with CORBA 2.3 Java language mapping, and supports commonly used CORBA services. It runs on all platforms that implement the Java Virtual Machine (JVM). JacORB is made available under the terms of the GNU Library General Public License (LGPL). Commercial support is provided by Object Computing Inc., a Sun Authorized Java Center and member of the OMG; for more information, go to http://www.ociweb.com.

JacORB operates with any CORBA-compliant ORB over IIOP. In practice, JacORB has been used successfully with at least the following ORBs: MICO, TAO, Orbacus, Iona Orbix, Borland VisiBroker, ORBit, omniORB, Vitria C++, and Java. ORB interoperability is made simple by using a foreign name service IOR in the file where the BEA WebLogic Adapter for CORBA looks up the name server OAR. This can be configured in the `jacorb.properties` file.

This section presents the following information about using JacORB with the Weblogic Adapter for CORBA:

- JacORB Name Service
- JacORB Interface Repository
- Building and Running the JacORB Request Broker

# JacORB Name Service

Name servers are used to locate objects using a human–readable reference (a name) rather than a machine or network address. If objects providing a certain service are looked up using the service name, clients are separated from the actual locations of the objects that provide the service. The binding from name to service can be changed without the client's knowledge.

JacORB provides an implementation of the OMG's Interoperable Naming Service (INS), which supports the binding of names to object references (and looking up object references using these names). It also allows clients to easily convert names to strings and vice versa. The JacORB name service comprises two components: the name server program, and a set of interfaces and classes used to access the service.

# JacORB Interface Repository

Run-time type information in CORBA is managed by the ORB's Interface Repository (IFR) component. It allows applications to request, inspect, and modify IDL type information dynamically. For example, the IFR enables applications to find out which operations an object supports. Some ORBs may also need the IFR to find out whether a given object's type is a subtype of another, but most ORBs can operate without the IFR by encoding this kind of type information in the helper classes generated by the IDL compiler.

In essence, the IFR is just another remotely accessible CORBA object that offers operations to retrieve (and, theoretically, to modify) type information. The IFR manages type information in a hierarchical containment structure that corresponds to constructs within IDL specifications: modules contain definitions of interfaces, structures, constants, and so on. Interfaces in turn contain definitions of exceptions, operations, attributes, and constants.

# Building and Running the JacORB Request Broker

To build and run the JacORB Request Broker, perform the following steps. (You will need ANT 1.4.1, a Java-based build tool.)

1. Unzip JacORB1_4_1-full.zip.

   This file is found in BEA_CORBA_SAMPLES.zip in the adapter's installation directory. It creates the directory JacORB1_4_1 on the selected drive.

   For example, if you unzip to drive D, the result is D:\JacORB1_4_1.

2. Unzip beacorba.zip in the directory JacORB1_4_1 that you created in Step 1.

   This file is found in BEA_CORBA_SAMPLES.zip in the adapter's installation directory.

3. Copy JacORB1_4_1\jacorb_properties.template and rename the copy JacORB1_4_1\jacorb.properties.

4. Edit the jacorb.properties file:

   – In the file's Initial References Configuration section, uncomment this line:

     ```
     #ORBInitRef.NameService=file:/d:/JacORB1_4_1/bea/ns_ref.ior
     ```

   – In the same line, replace d: with the drive and path into which you unzipped your JacORB files.

   – If the next line—#ORBInitRef.NameService=file...—is uncommented, comment it out.

5. Copy JacORB1_4_1\bea\jaco.bat into the JacORB\bin directory, replacing JacORB1_4_1\bin\jaco.bat.

6. Edit JacORB1_4_1\bea\setenv-sample.bat to specify directories in the following three statements:

   ```
   set JAVA_HOME=jdk_directory
   set JACORB_HOME=JacORB1_4_1_directory
   set ANT_HOME=ant_tool_directory
   ```

   Here, *jdk_directory* is the directory where your JDK resides, *JacORB1_4_1_directory* is the directory where JacORB resides, and *ant_tool_directory* is the directory where your ANT tool resides.

   For example:

   ```
   set JAVA_HOME=d:\apps\jdk1.3.1_04
   set JACORB_HOME=d:\apps\JacORB1_4_1
   set ANT_HOME=d:\apps\ant-1.5.2
   ```

7. Rename JacORB1_4_1\bea\club\ClubServer.java.jackorb to JacORB1_4_1\bea\club\ClubServer.java

**Note:** To compile for Visibroker, rename `JacORB1_4_1\bea\club\ClubServer.java.vb` to `JacORB1_4_1\bea\club\ClubServer.java`

8. Build the JacORB application by executing the following commands in a new DOS command window:

   ```
   d:JacORB1_4_1\bea\setenv-sample.bat
   ```

   ```
   d:JacORB1_4_1\bea\club\ant
   ```

   Here, *d:* is the drive and path into which you unzipped your JacORB files.

9. Start the Interface Repository service by executing the following commands in a new DOS command window:

   ```
   JacORB1_4_1\bea\setenv-sample.bat
   ```

   ```
   ir repository_class_path IOR_filename
   ```

   Here, *repository_class_path* is the path to your repository class files, and *IOR_filename* is the name of the Interface Object Repository file.

   For example:

   ```
   JacORB1_4_1\bea\ir ..\classes ir_ref.txt
   ```

10. Start the name service by executing the following commands in a new DOS command window:

    ```
    d:\JacORB1_4_1\bea\setenv-sample.bat
    ```

    ```
    ns [ins_filename][-p port] [-t timeout]
    ```

    Here, *d:* is the drive and path into which you unzipped your JacORB files, *ins_filename* is the name of the Naming Service file specified in the `jacorb.properties` file, *port* is the number of the port on which the service is listening, and *timeout* is the server timeout.

    The following is an example of the name service command line:

    ```
    JacORB1_4_1\bea\ns ns_ref.ior
    ```

11. Start JacORB's NameManager using the `nmg` command.

12. Start the Java interpreter explicitly by executing the following command:

    ```
    d:\JacORB1_4_1\bea\setenv-sample.bat
    ```

    Here, *d:* is the drive and path into which you unzipped your JacORB files.

13. Execute the following JacORB command:

```
jaco jacorb.naming.NameServer [filename][-p port][-t timeout]
```

Here, *jacorb.naming.NameServer* is the name of the Name Server, *filename* is the name of the Naming Service file specified in the jacorb.properties file, *port* is the number of the port on which the service is listening, and *timeout* is the server timeout.

For example:

```
JacORB1_4_1\bea\jaco bea.club.ClubServer
```

# Using Orbix2000

The BEA WebLogic Adapter for CORBA supports Orbix 2000 Versions 1.2 and communicates using IIOP version 1.1.

Orbix is a software environment for building and integrating distributed object-oriented applications. Orbix is a full implementation of the Common Object Request Broker Architecture (CORBA) from the Object Management Group (OMG). Orbix fully supports CORBA version 2.3. For more information, see the *Orbix 2000 Administrator's Guide.*

Orbix includes a CORBA IDL compiler, which is used by programmers to compile interface definitions along with the client and server code. A client application compiled in this way contains internal information about server objects. Clients use this information to invoke the remote objects. Orbix provides an interface repository, which enables clients to call operations on Interface Definition Language (IDL) interfaces that are unknown at compile time. The interface repository (IFR) provides centralized persistent storage of IDL interfaces. Orbix programs can query the interface repository at runtime to obtain information about IDL definitions.

To verify that your Orbix server is registered in the Naming Service, enter the following command at the command prompt:

```
itadmin ns list
```

Your server name should appear within a list of all the registered servers. If it is not in the list, you must register your server.

To ensure your IFR is running and populated, you can view its contents using the following command:

```
itadmin ifr list
```

This command lists all the currently scoped names, such as interfaces and types.

You can use the itadmin command to view the IDL definition of one of the current scoped names, as follows:

```
itadmin ifr show current_scoped_name
```

In order to populate the interface repository with IDL definitions, run the IDL compiler with the `-R` option. For example, the following command populates the interface repository with the IDL definitions in `bank.idl`:

```
idl -I. -I$(ART_IDL_DIR)\omg -R= bank.idl
```

For more information on the Orbix utilities and command line, see Section IV of the *Orbix 2000 Administrators Guide.*

# Using VisiBroker for Java

VisiBroker is a complete CORBA 2.3 Object Request Broker (ORB) that supports the development, deployment, and management of distributed object applications across a variety of hardware platforms and operating systems. In addition to VisiBroker (the ORB), three other components are available with VisiBroker:

- Naming Service, which allows you to associate one or more logical names with an object implementation and to store those names in a namespace. It also lets client applications use this service to obtain an object reference using the logical name assigned to that object.
- Event Service, which provides a facility that separates the communication between objects. It provides a *supplier-consumer* communications model that allows multiple *supplier objects* to send data asynchronously to multiple *consumer objects* through an event channel.
- Gatekeeper, which runs on a Web server and enables client programs to locate and use objects that do not reside on the Web server and to receive callbacks, even when firewalls are being used. The Gatekeeper can also be used as an HTTP daemon, thereby eliminating the requirement for a separate HTTP server during the application development phase.

The BEA WebLogic Adapter for CORBA supports VisiBroker for Java Version 4.5 and communicates using IIOP version 1.1. Applications created with VisiBroker for Java can communicate with object implementations developed with VisiBroker for C++

VisiBroker requires the Java Development Kit (JDK) or the Java Runtime Environment (JRE). You can obtain these tools from the Sun Microsystems Web site ( http://java.sun.com/). JRE version 1.2.2 or higher is required to run the VisiBroker Console. You must install the JRE before you install VisiBroker. However, VisiBroker supports any current version of Java for your applications.

The BEA WebLogic Adapter for CORBA requires that an IOR file be available to locate the reference to the Interface Repository. However, VisiBroker's IOR file is not automatically output to a file. Modify your IFR startup procedure to automatically output the startup IOR reference to a file. For example, use the following command to automatically output the IOR reference to `ir.ior:`

```
irep myIr >ir.ior
```

VisiBroker configuration and run-time requirements for using the BEA WebLogic Adapter for CORBA include:

- The `PATH` environment variable must point to the VisiBroker libraries used by BEA WebLogic Adapter for CORBA.
- The VisiBroker jar file must be in the class path.
- The IFR must be populated with the IDL of the objects for which you want to create Web services.
- The VisiBroker Naming Service and IFR must be running.
- The CORBA servers you are going to use must be running or be set up to start on demand.

For example, with VisiBroker you start the Interface Repository using the following command:

```
irep myIr >ir.ior
```

Here, *`myIR`* is the startup IOR reference.

With VisiBroker you start the naming service using the following command:

```
$start nameserv NS_name
```

Here, *`NS_name`* is the name of the naming service.

To verify that your server is registered in the Naming Service and your IFR is loaded:

1. Choose Start→Programs→VisiBroker→VisiBroker Console.
2. Expand VisiBroker ORB Services.
3. Expand the Naming Services folder or the IFR folder.

   A list of the Naming Service or IFR objects should appear in the right pane.

Alternately, you can enter the `osfind` command at a command prompt. The `osfind` command finds the name of the server running the VisiBroker Naming Service. It is usually the machine where VisiBroker is installed.

For more information, refer to *VisiBroker for Java Installation Guide*, available at: http://info.borland.com/techpubs/books/vbj/vbj45/framesetindex.html.

# Index