



# BEA WebLogic Adapter for CORBA™

## Installation and Configuration Guide for WebLogic Integration 2.1

# Copyright

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

Copyright © 2002 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 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.

### BEA WebLogic Adapter for CORBA Installation and Configuration Guide for WebLogic Integration 2.1

Part Number	Date
N/A	October 2002

---

# Table of Contents

## About This Document

Audience.....	vi
Related Information.....	vi
Contact Us!.....	vii
Documentation Conventions .....	viii

## 1. Installing the Adapter for WebLogic Integration 2.1

Before Installing the Adapter .....	1-2
Understanding the Representation of Paths.....	1-2
Step 1. Obtaining the BEA WebLogic Adapter for CORBA .....	1-3
Step 2. Setting Up Your ORB Environment .....	1-4
Step 3. Extracting JARs and Adjusting Classpath.....	1-4
Extracting JARs and Adjusting Classpath for Windows .....	1-4
Extracting JARs and Adjusting Classpath for UNIX.....	1-6
Step 4. Configuring the WebLogic Integration Database for the Domain .....	1-8
Step 5. Replacing the xmltoolkit.jar File.....	1-9
Step 6. Updating the BEA License.....	1-11
Step 7. Deploying the Adapter Using the WebLogic Server Console .....	1-12
Step 8. Adding the Administrative Server User Name to the Adapter Group .....	1-16
Step 9. Initializing BEA Application Explorer for JacORB.....	1-17
Next Steps.....	1-18

## A. Installing the JacORB Object Request Broker

Overview .....	A-1
JacORB Name Service.....	A-2
JacORB Interface Repository.....	A-2
Building and Running the JacORB Request Broker .....	A-3



---

# About This Document

This document explains how to install the BEA WebLogic Adapter for CORBA, which is used to develop client-server interfaces between CORBA and other applications. It describes how to install the BEA WebLogic Adapter for CORBA with WebLogic Integration 2.1 and use adapter tools to develop online connections to CORBA applications.

This document is organized as follows:

- [Chapter 1, “Installing the Adapter for WebLogic Integration 2.1,”](#) describes how to install the BEA WebLogic Adapter for CORBA.
- [Appendix A, “Installing the JacORB Object Request Broker,”](#) provides instructions for optionally installing the JacORB Object Request Broker in order to test the adapter using the samples provided.

# Audience

This document is written for system integrators who develop client interfaces between CORBA and other applications. It describes how to install and deploy the BEA WebLogic Adapter for CORBA and how to use it with WebLogic Integration 2.1 and adapter tools to develop online connections to CORBA applications. It is assumed that readers know Web technologies and have a general understanding of Microsoft Windows and UNIX systems as well as:

- General knowledge of CORBA concepts.
- Knowledge of CORBA processes and data model for the required application area.
- Knowledge of WebLogic Integration architecture.
- General knowledge of client-server concepts.

# Related Information

The following documents provide additional information for the associated software components:

- *BEA WebLogic Adapter for CORBA User Guide*
- *BEA WebLogic Adapter for CORBA Release Notes*
- *BEA Application Explorer Installation and Configuration Guide*
- BEA WebLogic Server installation and user documentation, which is available at the following URL:  
[http://edocs.bea.com/more\\_wls.html](http://edocs.bea.com/more_wls.html)
- BEA WebLogic Integration installation and user documentation, which is available at the following URL:  
[http://edocs.bea.com/more\\_wli.html](http://edocs.bea.com/more_wli.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](mailto: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 which version of the BEA WebLogic Adapter for CORBA documentation you are using.

If you have any questions about this version of the BEA WebLogic Adapter for CORBA, or if you have problems installing and running the BEA WebLogic Adapter for CORBA, contact BEA Customer Support through BEA WebSupport at [www.bea.com](http://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

# Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
<b>boldface text</b>	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	Indicates emphasis or book titles.
monospace text	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard. <i>Examples:</i> #include <iostream.h> void main ( ) the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float
<b>monospace boldface text</b>	Identifies significant words in code. <i>Example:</i> void <b>commit</b> ( )
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> String <i>expr</i>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON OR



Convention	Item
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed. <i>Example:</i> <code>buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...</code>
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	Indicates one of the following in a command line: <ul style="list-style-type: none"><li>■ That an argument can be repeated several times in a command line</li><li>■ That the statement omits additional optional arguments</li><li>■ That you can enter additional parameters, values, or other information</li></ul> The ellipsis itself should never be typed. <i>Example:</i> <code>buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...</code>
.	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.



# 1 Installing the Adapter for WebLogic Integration 2.1

This section provides instructions for installing the BEA WebLogic Adapter for CORBA. It includes the following topics:

- [Before Installing the Adapter](#)
- [Understanding the Representation of Paths](#)
- [Step 1. Obtaining the BEA WebLogic Adapter for CORBA](#)
- [Step 2. Setting Up Your ORB Environment](#)
- [Step 3. Extracting JARs and Adjusting Classpath](#)
- [Step 4. Configuring the WebLogic Integration Database for the Domain](#)
- [Step 5. Replacing the xmltoolkit.jar File](#)
- [Step 6. Updating the BEA License](#)
- [Step 7. Deploying the Adapter Using the WebLogic Server Console](#)
- [Step 8. Adding the Administrative Server User Name to the Adapter Group](#)
- [Step 9. Initializing BEA Application Explorer for JacORB](#)
- [Next Steps](#)

# Before Installing the Adapter

Before you install the BEA WebLogic Adapter for CORBA, you must review the *BEA WebLogic Adapter for CORBA Release Notes* to ensure that you have the required prerequisite software installed. The *BEA WebLogic Adapter for CORBA Release Notes* is available at the following URL:

<http://edocs.bea.com/wlapters/doc70/index.html>

# Understanding the Representation of Paths

Because the location of files in the WebLogic Integration environment depends on options selected during installation and configuration, the conventions that follow are used throughout to represent paths.

- *BEA\_HOME* represents the BEA Home directory specified for your WebLogic installation.

For example, if you install the product in the default location on a Windows system, *BEA\_HOME* represents `c:\bea`.

- *WLI\_HOME* represents the root of your WebLogic Integration installation.

For example:

- If you install WebLogic Integration 2.1 in the default location on a Windows system, *WLI\_HOME* represents `c:\bea\wlintegration2.1`.

- *domain* is used to indicate the name of a domain.

- In WebLogic Integration 2.1, preconfigured domains (`bpmdomain`, `eidomain`, `wlidomain`, and `samples`) are created as subdirectories of the *WLI\_HOME\config* directory. Therefore, *domain* may be used to represent the root of a preconfigured WebLogic Integration 2.1 domain as follows:

`WLI_HOME\config\domain`

- *DOMAIN\_HOME* represents the complete path to the root of a domain.

For example:

- If you install WebLogic Integration 2.1 in the default location on a Windows system, *DOMAIN\_HOME* represents  
`c:\bea\wlintegration2.1\config\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.

# Step 1. Obtaining the BEA WebLogic Adapter for CORBA

Obtain the following BEA WebLogic Adapter for CORBA software components on CD or download them from [www.bea.com](http://www.bea.com):

- The BEA WebLogic Adapter for CORBA, which is packaged as an EAR file (`BEA_CORBA_1_0.ear`).
- The `BEA_CORBA_SAMPLES.zip` file, which includes JacORB components and sample service schemas.

**Note:** At the time of publication of this document, WebLogic Server 6.1 is unable to explode RAR files (BEA Support CASE number 333672). Use an extraction tool (such as WinZip) to extract the contents of the adapter EAR file, `BEA_CORBA_1_0.ear`, and add the location of the unpackaged objects to the server's classpath (see [Step 3. Extracting JARs and Adjusting Classpath](#)).

# Step 2. Setting Up Your ORB Environment

Before installing and configuring the BEA WebLogic Adapter for CORBA, you should set up your ORB environment and configure it to work with the adapter, as described in Appendix A, “Using CORBA Implementations with the Adapter” in the *BEA WebLogic Adapter for CORBA User Guide*. Verify that your ORB infrastructure is properly configured, your server is registered in the Naming Service, and your interface repository (IFR) is running and populated.

# Step 3. Extracting JARs and Adjusting Classpath

Set the classpath using the procedure appropriate for your system:

- [Extracting JARs and Adjusting Classpath for Windows](#)
- [Extracting JARs and Adjusting Classpath for UNIX](#)

## Extracting JARs and Adjusting Classpath for Windows

To unzip the adapter JAR files and adjust the classpath on Windows, complete the following steps:

1. Use WinZip (or another similar extracting product) to extract the `BEA_CORBA_1_0.ear` file to a directory of your choice (for example, `BEA_HOME\AdapterEars`).
2. Extract `JacORB1_4_beta4.zip` from `BEA_CORBA_SAMPLES.zip`, then extract `jacorb.jar` from `JacORB1_4_beta4.zip`. Accept the default extraction locations.

3. Go to the root directory for your domain:

```
cd DOMAIN_HOME
```

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

4. Find the `SetDomainTypeData.cmd` file.

Here, *DomainType* is the type of the domain. For example, if your domain is configured to support the development of solutions that employ the full range of WebLogic Integration functionality, it contains the `SetwliDomainData.cmd` file.

5. Update the following SVRCP environment variable settings to the `SetDomainTypeData.cmd` file for the domain to include all the JAR files included in the EAR file.

**Note:** The SVRCP environment variable is used in the `SetDomainTypeData` script to set the classpath for the java executable.

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

After the following line:

```
set SVRCP=%SVRCP%;%WLI_DOMAIN_HOME%\wli
```

add the following JAR files, which are listed in the order required for the classpath:

```
SET SVRCP=%SVRCP%;BEA_HOME\AdapterEars\ibi-edaqm.jar
SET SVRCP=%SVRCP%;BEA_HOME\AdapterEars\xercesImpl.jar
SET SVRCP=%SVRCP%;BEA_HOME\AdapterEars\xmlParserAPIs.jar
SET SVRCP=%SVRCP%;BEA_HOME\AdapterEars\xmltoolkit.jar
SET SVRCP=%SVRCP%;BEA_HOME\AdapterEars
SET SVRCP=%SVRCP%;BEA_HOME\AdapterEars\BEA_CORBA_1_0.jar

REM Set the CLASSPATH for use with JacORB Object Request Broker
SET SVRCP=%SVRCP%;E:\JacORB1_4_beta4
```

Here, `BEA_HOME\AdapterEars` is the directory specified in step 1 and `E:\JacORB1_4_beta4` points to the directory where `jacorb.properties` resides.

6. Find the `StartWeblogic.cmd` file, which is in your `DOMAIN_HOME` directory along with the `SetDomainTypeData.cmd` file mentioned above.

## 1 Installing the Adapter for WebLogic Integration 2.1

---

7. Update the Java command line to add `-Xbootclasspath` after `%DB_JVMARGS%`.

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

```
%JAVA_HOME%\bin\java %DB_JVMARGS%
-Xbootclasspath/p:d:\JacORB1_4_beta4\lib\jacorb.jar -Xmx256m
-classpath
%SVRCP% -Dbea.home=%BEA_HOME% -Dweblogic.home=%WL_HOME%
-Dweblogic.system.home=%WLI_HOME%
-Dwli.bpm.server.evaluator.supportsNull=false
-Dweblogic.Domain=wliDomain
-Dweblogic.management.password=security
-Dweblogic.Name=myserver
-Dweblogic.RootDirectory=%WLI_HOME%
-Djava.security.policy=%WL_HOME%\lib\weblogic.policy
-Dweblogic.management.discover=false weblogic.Server
```

## Extracting JARs and Adjusting Classpath for UNIX

To extract the adapter JAR files and adjust the classpath on UNIX, complete the following steps:

1. Use `jar` (or another similar extracting product) to extract `BEA_CORBA_1_0.ear` to a directory of your choice (for example, `BEA_HOME/AdapterEars`).
2. Extract `JacORB1_4_beta4.zip` from `BEA_CORBA_SAMPLES.zip`, then extract `jacorb.jar` from `JacORB1_4_beta4.zip`. Accept the default extraction locations.
3. Go to the root directory for your domain:

```
cd DOMAIN_HOME
```

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

4. Find the `SetDomainTypeData.cmd` file.

Here, *DomainType* is the type of the domain. For example, if your domain is configured to support the development of solutions that employ the full range of WebLogic Integration functionality, it contains the `SetwliDomainData.cmd` file.



5. Update the following SVRCP environment variable settings to the `SetDomainTypeData.cmd` file for the domain to include all the JAR files included in the EAR file.

The SVRCP environment variable is used in the `SetDomainTypeData` script to set the classpath for the java executable.

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

After the following line:

```
SVRCP=$SVRCP;$WLI_DOMAIN_HOME/wlai
```

add the following JAR files, which are listed in the order required for the classpath:

```
SVRCP=$SVRCP:BEA_HOME/AdapterEars/ibi-edaqm.jar
SVRCP=$SVRCP:BEA_HOME/AdapterEars/xercesImpl.jar
SVRCP=$SVRCP:BEA_HOME/AdapterEars/xmlParserAPIs.jar
SVRCP=$SVRCP:BEA_HOME/AdapterEars/xmltoolkit.jar
SVRCP=$SVRCP:BEA_HOME/AdapterEars
SVRCP=$SVRCP:BEA_HOME/AdapterEars/BEA_CORBA_1_0.jar

# Set the CLASSPATH for use with JacORB Object Request Broker
SVRCP=$SVRCP:/usr/JacORB1_4_beta4
```

Here, `BEA_HOME/AdapterEars` is the directory specified in step 1 and `/usr/JacORB1_4_beta4` points to the directory where `jacorb.properties` resides.

6. Find the `StartWeblogic.cmd` file, which is in your `DOMAIN_HOME` directory along with the `SetDomainTypeData.cmd` file mentioned above.

7. Update the java command line to add -Xbootclasspath after \$DB\_JVMARGS.

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

```
$JAVA_HOME/bin/java $DB_JVMARGS  
-Xbootclasspath/p:/opt/JacORB1_4_beta4/lib/jacorb.jar -Xmx256m -classpath  
$SVRCP -Dbea.home=$BEA_HOME -Dweblogic.home=$WLI_HOME  
-Dweblogic.system.home=$WLI_HOME  
-Dwli.bpm.server.evaluator.supportsNull=false  
-Dweblogic.Domain=wlidomain  
-Dweblogic.management.password=security  
-Dweblogic.Name=myserver  
-Dweblogic.RootDirectory=$WLI_HOME  
-Djava.security.policy=$WLI_HOME/lib/weblogic.policy  
-Dweblogic.management.discover=false weblogic.Server
```

## Step 4. Configuring the WebLogic Integration Database for the Domain

If you have not already done so, you must create the WebLogic Integration database tables for your domain. For detailed instructions, see “Configuring the Database for a Domain” in *Starting, Stopping and Customizing BEA WebLogic Integration*, which is available at the following URL:

[http://edocs.bea.com/wlintegration/v2\\_1sp/config/index.htm](http://edocs.bea.com/wlintegration/v2_1sp/config/index.htm)

## Step 5. Replacing the xmltoolkit.jar File

The BEA WebLogic Adapters include a new `xmltoolkit.jar` file. You must replace your existing WebLogic Integration `xmltoolkit.jar` with the new JAR file.

To configure the new `xmltoolkit.jar` file, follow these steps:

1. Rename your original `xmltoolkit.jar` to `xmltoolkit.jar.old` by entering the commands appropriate for your operating system:

- On a Windows system:

```
cd WLI_HOME\lib
rename xmltoolkit.jar xmltoolkit.jar.old
```

- On a UNIX system:

```
cd WLI_HOME/lib
mv xmltoolkit.jar xmltoolkit.jar.old
```

2. Extract the `xmltoolkit.jar` from the adapter EAR file into a temporary directory.
3. Copy the new `xmltoolkit.jar` file (extracted in step 2) to the `WLI_HOME\lib` directory for Windows or the `WLI_HOME/lib` directory for UNIX.

**Caution:** Simply replacing the `xmltoolkit.jar` file is not sufficient; you must also make changes to the `setenv` and `SetDomainTypeData` scripts as described in the following steps.

4. Edit the top-level `setenv` script and make the appropriate changes for your operating system:

- On a Windows system, edit the `WLI_HOME\setenv.cmd` script.

Replace the line:

```
set WLICOMMONCP=
```

with

```
set WLICOMMONCP=%WLI_HOME%\lib\xmltoolkit.jar
```

## 1 Installing the Adapter for WebLogic Integration 2.1

---

- On a UNIX system, edit the `WLI_HOME/setenv.sh` script.

Replace the line:

```
WLICOMMONCP=$WLI_LIB/wlicommon.jar
```

with

```
WLICOMMONCP=$WLI_LIB/wlicommon.jar:$WLI_HOME/lib/xmltoolkit.jar
```

5. Edit the `SetDomainTypeData` script.

Here, *DomainType* is the type of the domain. For example, depending on the configuration of your domain, locate and edit the `SetwliDomainData.cmd` or `SeteaiDomainData.cmd` file.

- On a Windows system:

For example, edit the `DOMAIN_HOME\SetwliDomainData.cmd` script.

Replace the line:

```
set SVRCP=%WLISERVERCP%;%CMNCP%
```

with

```
set  
SVRCP=%WLI_HOME%\lib\xmltoolkit.jar;%WLISERVERCP%;%CMNCP%
```

- On a UNIX system:

For example, edit the `DOMAIN_HOME/SetwliDomainData` script.

Replace the line:

```
SVRCP=$WLISERVERCP:$CMNCP
```

with

```
SVRCP=$WLI_HOME/lib/xmltoolkit.jar:$WLISERVERCP:$CMNCP
```

## Step 6. Updating the BEA License

The BEA WebLogic Adapter for CORBA cannot be used without a valid software license. If you have downloaded the adapter for evaluation, you must obtain an evaluation license as described on the adapter download page. If you have purchased a license for the adapter, the license file is typically sent to you as an e-mail attachment.

When you have obtained a valid license for the adapter, update your `license.bea` file by completing the following steps:

1. Save the license file that you obtained with a name other than `license.bea`, in the `BEA_HOME` directory. For example, save the file as `corba_adapter_license.bea`. Use this file as the *license\_update\_file* in step 4 of this procedure.

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

2. Perform the step appropriate for your platform:
  - 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. If it is not already included, add the JDK to your `PATH` variable by executing the command appropriate to your system:
  - On a Windows system:

```
set PATH=BEA_HOME\jdk131_03\bin;%PATH%
```
  - On a UNIX system:

```
PATH=BEA_HOME/jdk131_03/bin:$PATH
export PATH
```

4. Merge the license update file into your existing license by executing the command appropriate to your system:

- 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 to which you saved the license update file in step 1.

5. Save a copy of your updated `license.bea` file in a safe place outside the WebLogic Integration and application installation directories.

## Step 7. Deploying the Adapter Using the WebLogic Server Console

After the BEA WebLogic Adapter for CORBA is installed, it must be deployed to WebLogic Server for your domain (for example, `wl1domain`). To configure and deploy an adapter using the WebLogic Server Administration Console, complete the following steps:

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

```
http://localhost:port/console/
```

Here, *localhost* represents the machine on which WebLogic Server is running and *port* represents the listening port.

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

3. When prompted, enter the user name and password for the server.

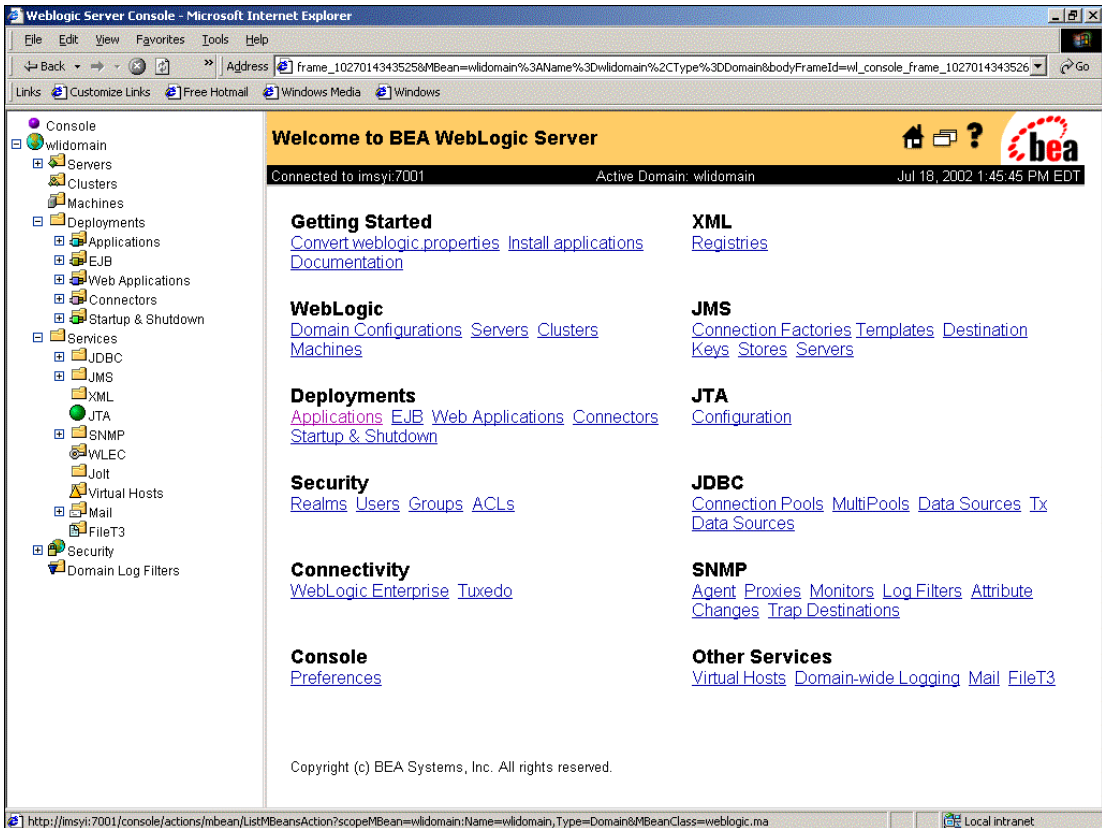
**Note:** If you have not updated the default login, see “WebLogic Integration Users and Passwords” in *Starting, Stopping, and Customizing WebLogic Integration* at the following URL:

## Step 7. Deploying the Adapter Using the WebLogic Server Console

[http://edocs.bea.com/wlintegration/v2\\_1sp/config/getstart.htm](http://edocs.bea.com/wlintegration/v2_1sp/config/getstart.htm)

The WebLogic Server Administration Console opens.

**Figure 1-1 WebLogic Server Console**

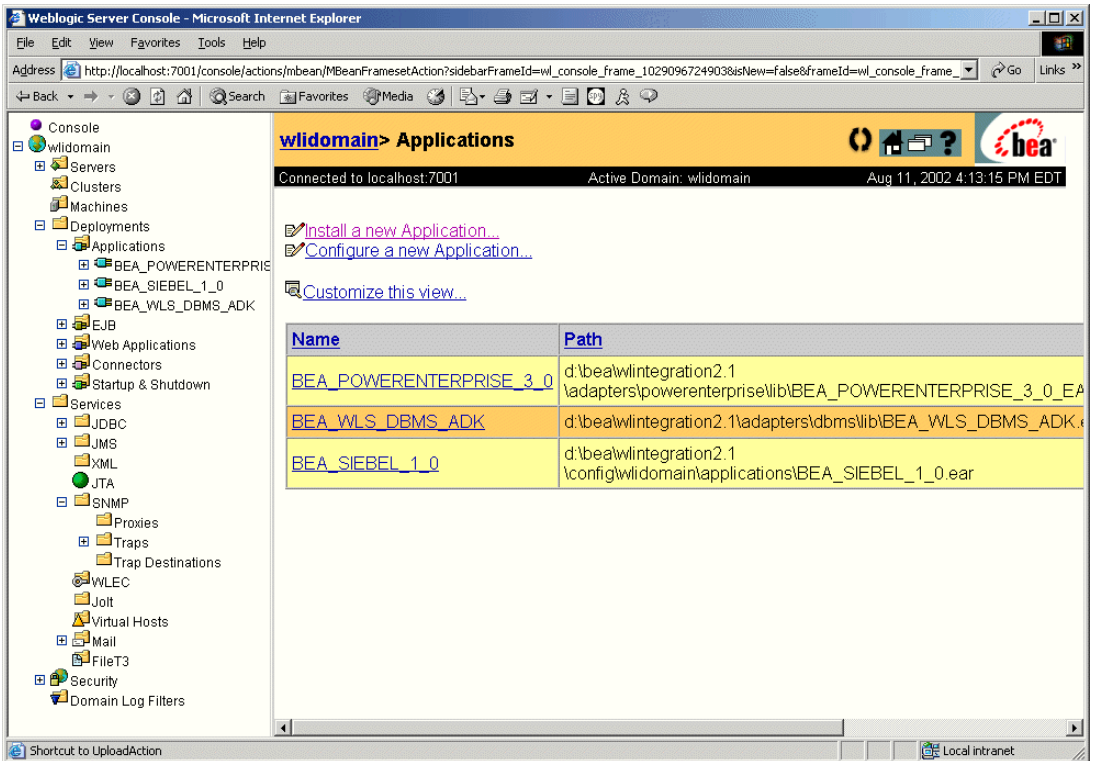


4. In the left pane, choose Deployments and then Applications from the navigation tree.

The console displays the Applications window.

# 1 *Installing the Adapter for WebLogic Integration 2.1*

**Figure 1-2 Applications Window**

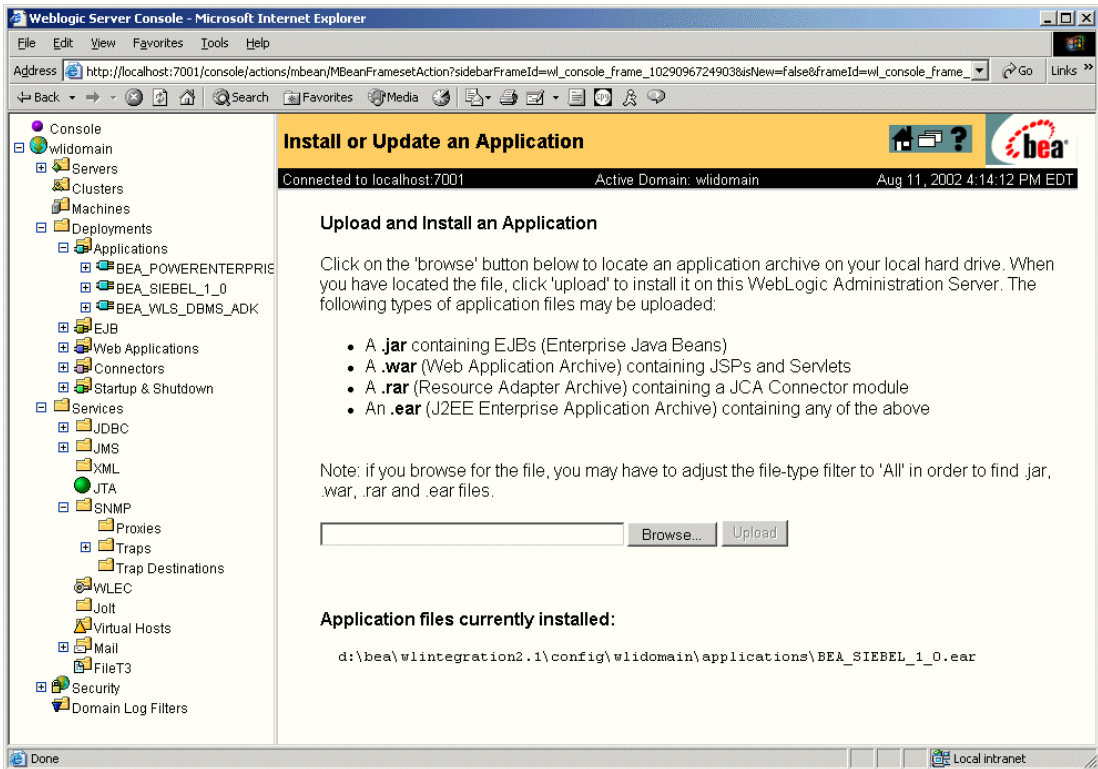


5. Click the Install a new Application link.

The console displays the Install or Update an Application window.



**Figure 1-3 Install or Update an Application Window**



6. Click Browse to locate the application archive you selected during installation (BEA\_CORBA\_1\_0.ear).
7. Click Upload to upload the BEA\_CORBA\_1\_0.ear file.

The console displays the application files currently installed to indicate that the upload is complete and the adapter file is deployed to WebLogic Server.
8. You can verify deployment by viewing the adapter configuration, as follows:
  - a. Choose Deployments and then Applications from the navigation tree.
  - b. Click the BEA\_CORBA\_1\_0.ear file link.

## Step 8. Adding the Administrative Server User Name to the Adapter Group

A user group, `adapter`, is defined in each domain that supports application integration functionality. Before you create an application view that employs the events or services supported by an adapter, you must add the user name defined for the administrative server to the `adapter` group.

**Note:** By default, the `adapter` group includes the user `system`. If the user name defined for the administrative server is `system`, skip this step. For example, if you are starting the server in a preconfigured domain, and you have not modified the default administrative server login, you can skip this step.

To add the administrative server user name to the `adapter` group, complete the following steps:

1. In the left pane of the WebLogic Server Administration Console, choose Security and then Groups from the navigation tree.

The console displays groups currently defined for the domain.

2. Locate and click the link for the `adapter` group to display the group definition.

**Figure 1-4 Group Definition**

The screenshot shows the WebLogic Server Administration Console interface. The breadcrumb navigation at the top reads "eaidomain> Realms> myRealm> Groups". The status bar indicates "Connected to localhost:7001", "Active Domain: eaidomain", and the date/time "Sep 10, 2002 7:18:19 PM EDT". The BEA logo is in the top right corner. The main content area is titled "Group" and displays the definition for the "adapter" group. It includes a "Name:" field with the value "adapter". Below it, the "Members:" section shows a list of users with checkboxes: ☐ joe, ☐ system, ☐ guest, ☐ hub, ☐ wlpisystem, ☐ mary, ☐ wlicsystem, ☐ admin, and ☐ wlicSamplesUser. A note "(select to remove)" is present. At the bottom left, there are two input fields labeled "Add Users:" and "Add Groups:". An "Apply" button is located at the bottom right of the form.

3. If the administrative server user name is not included in the Members list, enter the user name in the Add Users field.
4. Click Apply to add the user name to the group.  
The name is added to the Members list.

## Step 9. Initializing BEA Application Explorer for JacORB

To use the BEA Application Explorer to generate schemas for JacORB:

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. Uncomment the following statement:

- On a Windows system:

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

- On a UNIX system:

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

3. 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 $@
```

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

## Next Steps

If you have not already installed the BEA Application Explorer, install it now. See the *BEA Application Explorer Installation and Configuration Guide*.

When you have successfully deployed the adapter and installed the BEA Application Explorer, you can log on to the WebLogic Integration Application View Console to create application views that employ events and services supported by the BEA WebLogic Adapter for CORBA. For more information, see the *BEA WebLogic Adapter for CORBA User Guide*.

# A Installing the JacORB Object Request Broker

This section provides instructions for optionally installing the JacORB Object Request Broker in order to test the adapter using the samples provided. It contains the following topics:

- [Overview](#)
- [Building and Running the JacORB Request Broker](#)

## Overview

The BEA WebLogic Adapter for CORBA includes a sample ORB called JacORB, which is distributed as an open source sample ORB under the GNU license. The adapter also supports commercial ORBs, such as ORBIX and VisiBroker. For more information on these ORBs, see the corresponding ORB documentation.

JacORB is an open source implementation of the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) specification, targeted at developers using the Java language. It is being supplied with your software to enable you to test your adapter using the samples provided.

JacORB is designed to be compliant with the 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 using the GNU Library General Public License (LGPL) terms. Commercial support is provided by Object Computing Inc (OCI). OCI is a Sun Authorized Java Center and member of the OMG. For more information, see the Web site <http://www.ocweb.com>.

JacORB interoperates 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.

## **JacORB Name Service**

Name servers are used to locate objects using a human-readable reference (their name) rather than a machine or network address. If objects providing a certain service are looked up using the service name, their clients are decoupled from the actual locations of the objects that provide this service. The binding from name to service can be changed without the clients needing to know.

JacORB provides an implementation of the OMG's Interoperable Naming Service (INS) which supports binding names to object references and to lookup 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 (IR) component. It allows applications to request, inspect, and modify IDL type information dynamically, for example, to find out which operations an object supports. Some ORBs may also need the IR to find out whether a given object's type is a subtype of another. Most ORBs can do without the IR by encoding this kind of type information in the helper classes generated by the IDL compiler.

In essence, the IR is a remotely accessible CORBA object that offers operations to retrieve (and in theory also modify) type information. The IR manages type information in a hierarchical containment structure that corresponds to the structure of

scoping constructs in 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 the ANT 1.4.1, a Java-based build tool.)

1. Unzip `JacORB1_4_beta4-full.zip`. This will create a directory called `JacORB1_4_beta4` on the selected drive. For example, if you selected to unzip to drive D, the result will be `D:\JacORB1_4_beta4`.
2. Unzip `beacorba.zip` under the `JacORB1_4_beta4` directory created in step 1.
3. Copy the `JacORB1_4_beta4\jacorb_properties.template` and rename it `JacORB1_4_beta4\jacorb.properties`.
4. Edit the `jacorb.properties` file as follows:  

```
ORBInitRef.NameService=file:/d:/JacORB1_4_beta4/bea/ns_ref.txt
```

Here, *d* is the drive where you unzipped your JacORB files.
5. Copy `JacORB1_4_beta4\bea\jaco.bat` into the `JacORB\bin` directory, replacing `JacORB1_4_beta4\bin\jaco.bat`.

6. Edit `JacORB1_4_beta4\bea\setenv-sample.bat`.

```
set JAVA_HOME=jdk_directory
set JACORB_HOME=JacORB1_4_beta4_directory
set ANT_HOME=ant_tool_directory
```

Here, *jdk\_directory* is the directory where your JDK resides, *JacORB1\_4\_beta4\_directory* is the directory where JacORB resides, and *ant\_tool\_directory* is the directory where your ANT tool resides. For example,

```
set JAVA_HOME=c:\jdk1.3
set JACORB_HOME=d:\JacORB1_4_beta4
set ANT_HOME=c:\jakarta-ant-1.4
```

7. Rename `JacORB1_4_beta4\bea\club\ClubServer.java.jackorb` to `JacORB1_4_beta4\bea\club\ClubServer.java`

**Note:** To compile for Visibroker, rename

```
JacORB1_4_beta4\bea\club\ClubServer.java.vb to
JacORB1_4_beta4\bea\club\ClubServer.java
```

8. Build the JacORB application. In a new DOS command window:
  - a. Execute `JacORB1_4_beta4\bea\setenv-sample.bat`
  - b. Execute `JacORB1_4_beta4\bea\club\ant`
9. Start the Interface Repository service. In a new DOS command window:
  - a. Execute `JacORB1_4_beta4\bea\ setenv-sample.bat`
  - b. Execute the following command:

```
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_beta4\bea\ir ..\classes ir_ref.txt.
```



10. Start the name service. The JacORB NameManager, a GUI for the name service, can be started using the `nmg` command. The NameManager then tries to connect to an existing name service. The JacORB name server is a process that needs to be started before the name service can be accessed by programs. In a new DOS command window:

- a. Execute `D:\JacORB1_4_beta4\bea\ setenv-sample.bat`
- b. Execute the following command:

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

Here, *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. For example,

```
JacORB1_4_beta4\bea\ns ns_ref.txt
```

11. Start the Java interpreter explicitly by typing:

- a. Execute `D:\JacORB1_4_beta4\bea\ setenv-sample.bat`
- b. Execute the following command:

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

Here, *jacob.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_beta4\bea\jaco bea.club.ClubServer
```

