



BEA WebLogic Platform™

ISV Partners' Guide

Copyright

Copyright © 2004-2006 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

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

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

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

Trademarks or Service Marks

BEA, BEA JRockit, BEA Liquid Data for WebLogic, BEA WebLogic Server, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA MessageQ, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic JRockit, BEA WebLogic Log Central, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server Process Edition, BEA WebLogic WorkGroup Edition, BEA WebLogic Workshop, and Liquid Computing are trademarks of BEA Systems, Inc. BEA Mission Critical Support is a service mark of BEA Systems, Inc. All other company and product names may be the subject of intellectual property rights reserved by third parties.

All other trademarks are the property of their respective companies.

Contents

About This Document

What You Need to Know	viii
Product Documentation on the dev2dev Web Site.	viii
Related Information	viii
Contact Us!	viii
Documentation Conventions	ix

1. Overview

Types of ISVs.	1-1
ISV Resources, Services, and Tools	1-2
WebLogic Platform ISV Edition.	1-3
BEA Partner Program.	1-5
Software Bundle	1-5
ISV Redistribution License.	1-6
WebLogic Platform Installation and Configuration Tools	1-6
Installation Program.	1-7
Configuration Wizard and Template Builder.	1-7
Silent Installation and Configuration.	1-7
Control Development Kit	1-8
Partner Support	1-8
ISV Tasks and Documentation Links	1-8

2. Developing Applications

Before You Develop	2-1
Applications You Can Develop on WebLogic Platform.	2-2
Using WebLogic Platform ISV Edition	2-5
Product Description	2-5
WebLogic Platform Components Included in ISV Edition	2-7
Installing WebLogic Platform ISV Edition	2-8
WebLogic Platform ISV Edition Licenses.	2-9
Building Portal Applications for Use with ISV Edition	2-9
File-Based Portals vs. Streaming Portals	2-10
Creating and Configuring Your Portal Development Environment	2-12
Testing Your Portal Development Environment.	2-17
Building Your Portals	2-18
Administering Users of a File-Based Portal	2-18
Deploying the Portal Application	2-19
Building Process Edition Applications for Use with ISV Edition.	2-20
Using the Partner Software Bundle.	2-21
Software Bundle Contents	2-21
Installing the Software Bundle.	2-22
Downloading BEA Software	2-22
Software Products Not Included in the Software Bundle	2-22
Next Steps	2-22

3. Configuring and Installing WebLogic Platform Applications

Configuring WebLogic Platform for Your Application	3-1
Configuration Wizard	3-2
Creating Domains for WebLogic Platform ISV Edition	3-2
For More Information	3-3

Template Builder	3-3
WebLogic Platform Administration Tools	3-4
WebLogic Server Administration Console	3-4
WebLogic Integration Administration Console	3-5
WebLogic Administration Portal	3-5
Startup Scripts	3-6
Silent-Mode Installation and Configuration	3-7
Silent-Mode Installation	3-7
Silent-Mode Domain Configuration	3-7
Running Silent-Mode Installation and Configuration from a Script	3-8

4. Distributing WebLogic Platform

Using an ISV Redistribution License	4-1
About the ISV Redistribution License	4-1
ISV Redistribution License Technical Enforcement	4-2
Creating an ISV Redistribution License and Linking It to the WebLogic Platform Files 4-3	
Step 1: Prepare to Create an ISV Redistribution License	4-4
Step 2: Extract the License Data and Link the WebLogic Platform Files	4-4
Validating the ISV Redistribution License File	4-5
Distributing Files	4-6
Service Packs and License Files	4-6

5. Creating and Packaging WebLogic Workshop Java Controls

Java Controls Overview	5-1
Custom Java Controls and Workshop's Extensibility Model	5-2
Ways You Can Extend Workshop's Environment	5-2
Extensibility Portal	5-2
Key Benefits of Java Controls to ISVs	5-3

How to Make Custom Java Controls Available to End Users	5-3
For More Information About Java Controls	5-4
Building Advanced Controls	5-4
Creating Online Help	5-5
Creating a Control Archive	5-6
Control Deliverables	5-6
Control Stubs	5-7

6. Supporting and Distributing WebLogic JRockit

About WebLogic JRockit	6-1
Obtaining WebLogic JRockit	6-2
Support for WebLogic JRockit in WebLogic Platform Applications	6-2
Distributing the JRockit JVM as a Standalone Product	6-3

7. Preparing Customer Documentation

General Advice on Preparing Customer Documentation	7-1
Documentation for Silent-Mode Installation and Configuration	7-2
Documentation for Interactive Installation and Configuration	7-2

8. About BEA Partner Support

Partner Support Resources	8-1
Partner Enablement Resources	8-2
Software Development Resources	8-2
Sales and Marketing Resources	8-3
Support Resources	8-3
Upgrading WebLogic Platform	8-4

A. Configuration Files Required for File-Based Portal Applications

Index

About This Document

This document describes the services and tools provided by BEA to facilitate your job, as a BEA partner, of making your applications easy to use and delivering them to your customers. It is organized as follows:

- [Chapter 1, “Overview,”](#) describes the resources, services, and tools that BEA provides for Independent Software Vendors (ISVs) who distribute WebLogic Platform or build software products, value-added solutions, or on-demand products to be used with WebLogic Platform.
- [Chapter 2, “Developing Applications,”](#) summarizes the types of applications you can build with WebLogic Platform, and also with WebLogic Platform ISV Edition, and provides links to more information about creating, testing, and deploying such applications.
- [Chapter 3, “Configuring and Installing WebLogic Platform Applications,”](#) explains how partners can help their own customers with the WebLogic Platform configuration work they must do before running the partner’s application.
- [Chapter 4, “Distributing WebLogic Platform,”](#) explains how to create the ISV redistribution license and link it to the WebLogic Platform software that you distribute with your value-added solution.
- [Chapter 5, “Creating and Packaging WebLogic Workshop Java Controls,”](#) gives an overview of the WebLogic Workshop Java controls and explains how to package controls for distribution to customers.

- [Chapter 6, “Supporting and Distributing WebLogic JRockit,”](#) describes the BEA contract for redistribution of WebLogic JRockit and different types of customer support that are available for partners’ customers.
- [Chapter 7, “Preparing Customer Documentation,”](#) provides a list of minimum requirements for customer documentation provided by BEA partners, and descriptions of the instructions needed for both silent and interactive modes of installation and configuration.
- [Chapter 8, “About BEA Partner Support,”](#) describes the resources provided by BEA to its partners, and provides suggestions for handling customer updates to WebLogic Platform.
- [Appendix A, “Configuration Files Required for File-Based Portal Applications,”](#) provides listings of four configuration files that need to be copied into the project directory of each portal application that you create for use with WebLogic Platform ISV Edition.

What You Need to Know

This document is intended for new or existing BEA partners who want to distribute products that include or interoperate with WebLogic Platform.

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

For more information in general about WebLogic Platform and its components, see the WebLogic Platform Documentation at <http://e-docs.bea.com/platform/docs81/index.html>.

Contact Us!

Your feedback on the BEA WebLogic Platform 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 WebLogic Platform documentation.

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Platform.

If you have any questions about this version of BEA WebLogic Platform, or if you have problems installing and running BEA WebLogic Platform, contact BEA Customer Support at <http://support.bea.com>. You can also contact Customer Support by using the contact information provided on the quick reference sheet titled “BEA Customer Support,” 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
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	Indicates emphasis or book titles.
monospace text	<div>Indicates <i>user input</i>, as shown in the following examples:</div> <ul style="list-style-type: none">• Filenames: <code>config.xml</code>• Pathnames: <code>BEAHOME/config/examples</code>• Commands: <code>java -Dbea.home=BEA_HOME</code>• Code: <code>public TextMsg createTextMsg(</code> <div>Indicates <i>computer output</i>, such as error messages, as shown in the following example:</div> <pre>Exception occurred during event dispatching:java.lang.ArrayIndexOutOfBoundsException: No such child: 0</pre>

Convention	Item
monospace boldface text	Identifies significant words in code. <i>Example:</i> void commit ()
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> String <i>expr</i>
{ }	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> java utils.MulticastTest -n <i>name</i> [-p <i>portnumber</i>]
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed. <i>Example:</i> java weblogic.deploy [<i>list deploy update</i>]
...	Indicates one of the following in a command line: <ul style="list-style-type: none"> • That an argument can be repeated several times in a command line • That the statement omits additional optional arguments • That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. <i>Example:</i> buildobjclient [-v] [-o <i>name</i>] [-f "file1.cpp file2.cpp file3.cpp . . ."]
.	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Overview

This document describes the services and tools provided by BEA to facilitate your job, as a BEA Independent Software Vendor (ISV) partner, of making your applications easy to use and delivering them to your customers. It begins, in this chapter, with the following topics:

- [Types of ISVs](#)—For the purposes of this document, defines the classes of BEA partners addressed by this document
- [ISV Resources, Services, and Tools](#)—Describes the resources that BEA provides to partners who distribute either WebLogic Platform or their own custom applications that interoperate with WebLogic Platform
- [ISV Tasks and Documentation Links](#)—Shows which tasks should be performed by each type of BEA partner and provides links to instructions for those tasks

Types of ISVs

To improve its ability to distribute WebLogic Platform and enhancements to it, BEA works in partnership with many ISVs. The following table defines the various types of ISVs that work with BEA in this way.

Table 1-1 Types of ISVs that Partner with BEA

ISV partners who deliver this type of product . . .	Distribute . . .
Bundled	WebLogic Platform, or one or more components of WebLogic Platform, packaged with their value-added solution.
Layered	Prepackaged WebLogic applications that run on WebLogic Platform, but that do not include WebLogic Platform. It is assumed that customers who buy layered products have valid WebLogic Platform installations.
Component	Custom WebLogic Workshop controls and IDE extensions that they have designed and built
JRockit	WebLogic JRockit, either as a standalone product or with another product
Complementary	Products that interoperate with WebLogic Platform, but that do not necessarily run as WebLogic Platform applications, such as development tools, security providers, content management providers, management tools, and monitoring tools

ISV Resources, Services, and Tools

BEA provides the following resources and tools for its partners:

- [WebLogic Platform ISV Edition](#)
- [BEA Partner Program](#)
- [ISV Redistribution License](#)
- [WebLogic Platform Installation and Configuration Tools](#)
- [Control Development Kit](#)
- [Partner Support](#)

This section describes each of these resources in detail.

WebLogic Platform ISV Edition

WebLogic Platform ISV Edition is a special software package tailored for ISVs who have a current agreement with BEA to build value-added solutions on WebLogic. WebLogic Platform ISV Edition comprises a set of WebLogic Platform components specifically packaged to help ISVs jumpstart their service-oriented architecture (SOA) initiatives.

WebLogic Platform ISV Edition leverages the capabilities of WebLogic Platform to provide a means to rapidly develop applications that perform the following:

- Service orchestration via processes
- Visual data mapping
- Process monitoring and management

WebLogic Platform ISV also includes WebLogic Portal's run-time framework to develop and run rich-featured portals and Web front ends to applications.

The following WebLogic Platform components are included with ISV Edition:

- WebLogic Server Premium Edition—Enables full support for WebLogic Server functionality, including the core Java 2 Enterprise Edition (J2EE) features and WebLogic Workshop Web applications, Web services, and controls. WebLogic Server Premium Edition provides premium clustering, caching, and messaging capabilities.
- WebLogic Workshop—A unified, integrated development framework that makes it easy for all developers—not only J2EE experts—to build powerful, standards-based J2EE applications across the entire WebLogic Platform. The WebLogic Workshop architecture includes:
 - Workshop Integrated Development Environment (IDE)

A graphical environment that you can use for building, testing, and debugging WebLogic Platform applications. The Workshop IDE's intuitive user interface lets you design your application visually and hides the complexity of J2EE programming.
 - Workshop Run-Time Framework

A program that compiles the annotated code created in the IDE and automatically implements the appropriate J2EE components, such as EJBs, JSPs, and JDBC connections, that are required to build and run the specified application.
- WebLogic JRockit—A high-performance JVM optimized for server-side performance and scalability. WebLogic JRockit is supported by all WebLogic Platform components and is now fully integrated into the WebLogic Platform package.

- WebLogic Portal—ISV Edition includes a subset of WebLogic Portal functionality, called the portal framework, that enables the rapid development of portal interfaces to Web applications independently of application logic or Web pages. The portal framework includes single file mode run-time support for rendering portal applications to end users.

The portal framework provides several key features for application development that partners can leverage to do the following:

- Easily and seamlessly surface information and applications—including JSPs, business processes, Web services, and portlets—into well-defined portal interfaces.
 - Develop all pieces of a portal, from pages and portlets to a portal’s user interface elements, as individual components that can be built quickly and independently, and can be reused.
 - Develop and use federated “plug & play” portlets that are compliant with the Web Services for Remote Portlets (WSRP) standard. (Note that the portal framework also provides a standardized Inter-Portlet Communication (IPC) package for all portlet types— JSP, JPF, and Struts, in addition to WSRP.)
- WebLogic Server Process Edition—ISV Edition includes all of WebLogic Server Process Edition functionality, which:
 - Enables the development and deployment of process driven, service-oriented applications.
 - Enables you to translate between XML, non-XML, and Java data formats, allowing you to rapidly integrate heterogeneous applications regardless of the format used to represent data.

WebLogic Platform ISV Edition is intended for use with ISV applications and is not intended for general-purpose use or development of other custom applications. For more information about ISV Edition and developing applications for it, see [“Using WebLogic Platform ISV Edition” on page 2-5](#). For important information about distributing ISV Edition applications, see [Chapter 4, “Distributing WebLogic Platform.”](#)

Note: WebLogic Platform ISV Edition is an entry level packaging of the full WebLogic Platform product. If there are features outside the scope of ISV Edition that you need for your value-added solution, your BEA partner representative will work with you to choose the set of components that you need.

WebLogic Platform ISV Edition is available for download from the following site:

<http://commerce.bea.com/showallproducts.jsp>

BEA Partner Program

The BEA Partner Program is a forum for jointly developing business opportunities with technology companies that redistribute BEA products, either as stand-alone software or as part of a package that includes their own software. The goal of this program is to provide partners with resources that can accelerate time-to-market profitability, such as technical services, education, and marketing resources.

Partners worldwide benefit from services such as discounts on instructor-led classes, technical Webcasts, and Web-based product training. The program also offers comprehensive technical services and support to software developers

This document, *WebLogic Platform ISV Partners' Guide*, is written for partners who enroll in the BEA Partner Program as a Software Partner. BEA Software Partners include Independent Software Vendors (ISVs) developing technology that complements and extends the functionality of BEA WebLogic Platform. BEA Software Partners also include Application Service Providers: vendors who provide full solutions, including hardware, hosting, software, and services.

If you are not already enrolled in the BEA Partner Program and you would like to enroll, complete the following steps:

1. Verify that your target hardware-software configurations are supported by WebLogic Platform. For a complete list of supported configurations, see [Supported Configurations for WebLogic Platform 8.1](#).
2. Learn more about the program at the BEA Partner Program Web site:
<http://www.bea.com/partners>
3. Enroll in the BEA Partner Program at the same Web site.

Software Bundle

As a benefit of the BEA Partner Program, partners who enroll receive a set of software licenses and media, which includes BEA Tuxedo with Jolt and the following offerings:

- WebLogic SDK, Professional Edition 8.1
Enables full development and scale-limited commercial usage rights of WebLogic Platform, and includes support.
- WebLogic Platform test server licenses
Grants partners the ability to load the WebLogic Platform software onto test servers for the purposes of QA and pre-production testing, creating demonstrations, and creating and

testing proof-of-concept applications and other prototypes in configurations that potentially require a great deal of scalability.

For comprehensive information about the software and licenses, enrolled partners can access the *BEA Partner Program Guide* at the following URL:

ftp://pdownload:BUY_ME@ftpna2.bea.com/pub/program/BEA_Partner_Program_Guide.pdf

ISV Redistribution License

The WebLogic Platform package available from BEA normally includes a license that prohibits the buyer from redistributing the software. However, for partners who are in the business of redistributing WebLogic Platform or its components, BEA provides the ability for ISVs to generate a license for their customers so that they can run the WebLogic Platform software that is bundled or embedded with the ISV's value-added solution. This generated license is called the *ISV redistribution license*.

Although the function of all ISV redistribution licenses is the same—to provide customers of ISVs with the ability to run the version of the WebLogic Platform software distributed by the ISV—each license is unique, specific to each individual partner, that controls the specific set of WebLogic Platform components that ISV customers can use at run time.

ISV partners who have current agreement to distribute a value-added solution with WebLogic Platform ISV Edition receive software that allows ISVs to generate both an ISV redistribution license and a version of the WebLogic Platform software that is associated with that license. The ISV redistribution license allows ISV customers to run the specific WebLogic Platform components associated with that license. (For a complete list of components included in WebLogic Platform ISV Edition, see “[WebLogic Platform Components Included in ISV Edition](#)” on page 2-7.)

To find out how to generate an ISV redistribution license and package it with the WebLogic Platform software developed by your company, see [Chapter 4, “Distributing WebLogic Platform.”](#)

WebLogic Platform Installation and Configuration Tools

To simplify the tasks of installing and configuring WebLogic Platform, BEA provides the following tools:

- [Installation Program](#)

- [Configuration Wizard and Template Builder](#)
- [Silent Installation and Configuration](#)

For each tool, BEA provides a simple and intuitive graphical user interface (GUI). The tools are designed, specifically, to help you create ISV-ready WebLogic Platform configurations that your customers will be able to install easily.

Installation Program

BEA's installation program makes it easy to install either the entire WebLogic Platform product or individual components of it—whichever you prefer. The program offers an easy-to-use tool called Smart Update, which enables you to obtain and install product upgrades. In addition, the installation program helps you launch installers for third-party products, such as XMLSPY and Stylesheet Designer.

When run in noninteractive or *silent* mode, the installation program makes it even easier for your customers to install WebLogic Platform.

Configuration Wizard and Template Builder

If you are planning to distribute WebLogic applications to customers, you may want to take advantage of two tools that can help you simplify the configuration of your customer's environment: the Configuration Wizard and the Template Builder. Using the Template Builder, you can create custom templates of the WebLogic domains required to run your applications. After your customers install your product, they can run the Configuration Wizard to set up their environments based on these custom templates.

Like the installation program, the Configuration Wizard can be run in silent mode. In other words, after your product is installed, you can run the Configuration Wizard to set up your customer's environment without any customer input.

Silent Installation and Configuration

By creating a simple shell script or command file that sequentially runs silent mode installation and silent mode configuration, you can simplify even further your customers' work of installing and configuring WebLogic Platform. This gives you customers a streamlined process, right out of the box, so that they can start running your applications more quickly and easily.

Control Development Kit

WebLogic Workshop Version 8.1 offers a new feature for Workshop applications: the ability to develop Java controls. Java controls are server-side components that encapsulate external resources and business logic for use in Workshop applications. They are integrated with the Workshop IDE and managed by the Workshop run-time framework.

To enable users to configure the run-time behavior of the Java controls in your Workshop application, you can also add a custom, design-time user interface to your Java controls.

For application programmers using the Workshop IDE, Java controls are a set of methods and callbacks that expose simple Java method interfaces. These interfaces, in turn, are customized to access a specific resource. By providing easy defaults and extensive examples, Workshop makes it easy for both partners and end users to build controls.

As a partner, you can use controls to create simple wrappers for potentially complex application software. Such a wrapper, exposed in WebLogic Workshop, can serve as a user-friendly software development interface to your final product software. To help you build your own controls, WebLogic Workshop provides a package called the Control Development Kit (CDK): a set of APIs, samples, and documentation, that you can package for distribution to your customers, along with your own software. The CDK is part of the Extension Development Kit (EDK).

Partner Support

After you enroll in the BEA Partner Program, you receive access to a comprehensive set of tools and resources that support the entire scope of partner enablement, training, software development, marketing and sales assistance, and support. For more information about partner support provided by the BEA Partner Program, see [Chapter 8, “About BEA Partner Support.”](#)

ISV Tasks and Documentation Links

BEA collaborates with Independent Software Vendors (ISVs) who sell the following types of products:

- Bundled
- Layered
- Component
- Standalone JRockit
- Complementary

[Table 1-2](#) describes the job required to sell and deliver each type of product, recommends a procedure for performing that job, and provides links to instructions, in this document, for each procedure.

Table 1-2 ISV Tasks and Related Documentation

If you are an ISV selling this type of product . . .	Your job consists of . . .	Complete the following procedure . . .
Bundled	Redistributing part or all of BEA WebLogic Platform	<ol style="list-style-type: none"> 1. Enroll in the Partner Program. See “BEA Partner Program” on page 1-5. 2. Back up your <code>license.bea</code> file. See “Using the Partner Software Bundle” on page 2-21. 3. Install the partner software bundle. See “Using the Partner Software Bundle” on page 2-21. 4. Create the ISV redistribution license and link it to the WebLogic Platform software you will redistribute. 5. Create a package to distribute to your customers, ensuring that the ISV redistribution license and the WebLogic Platform software to which it is linked is installed correctly. See Chapter 4, “Distributing WebLogic Platform.” <p>Note: For instructions for distributing custom software along with any or all of WebLogic Platform, see steps 3 through 6 in the procedure for ISVs of layered products, later in this table.</p>

Table 1-2 ISV Tasks and Related Documentation (Continued)

If you are an ISV selling this type of product . . .	Your job consists of . . .	Complete the following procedure . . .
Layered	Distributing a prepackaged WebLogic application that runs on WebLogic Platform or its components	<ol style="list-style-type: none">1. Enroll in the Partner Program. See “BEA Partner Program” on page 1-5.2. Install the partner software bundle. See “Using the Partner Software Bundle” on page 2-21.3. Build, test, and deploy the application. See “Applications You Can Develop on WebLogic Platform” on page 2-2.4. Configure the WebLogic domain in which the application is meant to be run. Then create a template or template extension of that domain. For general information about configuring WebLogic Platform, see Chapter 3, “Configuring and Installing WebLogic Platform Applications.” For information about the Configuration Wizard, see “Configuration Wizard” on page 3-2. For information about the Template Builder, see “Template Builder” on page 3-3.5. Create an appropriate distribution mechanism (such as an installation program or a Web download) for your application. If you want to deliver your product with a script that runs the Configuration Wizard in silent mode, write that script now. (See “Template Builder” on page 3-3.)6. Create user documentation for your application. If you want your customers to run the Configuration Wizard to create a domain based on a custom template or a template extension you have provided, write instructions for doing so. See Chapter 7, “Preparing Customer Documentation.”

Table 1-2 ISV Tasks and Related Documentation (Continued)

If you are an ISV selling this type of product . . .	Your job consists of . . .	Complete the following procedure . . .
Component	Creating and distributing WebLogic Workshop controls and IDE extensions	<ol style="list-style-type: none"> 1. Enroll in the Partner Program. See “BEA Partner Program” on page 1-5. 2. Install WebLogic Platform. See Installing BEA WebLogic Platform. 3. Build and test the component. See Chapter 5, “Creating and Packaging WebLogic Workshop Java Controls.” 4. Create online help for the component. See “Creating Online Help” on page 5-5. 5. Create an archive of the component files, along with documentation. For information about creating a control archive, see “Creating a Control Archive” on page 5-6. <p>For more information about WebLogic Workshop extensibility, visit the following Web site: http://dev2dev.bea.com/products/wlworkshop81/ext_overview.jsp</p>
JRockit	Redistributing the WebLogic JRockit JVM	<ol style="list-style-type: none"> 1. Enroll in the Partner Program. See “BEA Partner Program” on page 1-5. 2. Familiarize yourself with the WebLogic JRockit JVM, and with BEA’s policies for providing and supporting it. 3. Learn about how BEA distributes WebLogic JRockit and provides support, for this product, to partners and their customers. See Chapter 6, “Supporting and Distributing WebLogic JRockit.”
Complementary	Creating non-BEA software for use in a WebLogic Platform environment	<ol style="list-style-type: none"> 1. Enroll in the Partner Program. See “BEA Partner Program” on page 1-5. 2. If you are creating management or monitoring tools, see Programming WebLogic Management Services with JMX for detailed information and code samples for working with WebLogic Server MBeans.

Developing Applications

This chapter provides links to documentation that explains how to build, test, and deploy WebLogic Platform applications. This chapter also describes:

- Applications you can build with and run on WebLogic Platform ISV Edition
- The software bundle that is provided to partners who enroll in the Partner Program. The software bundle contains the entire WebLogic Platform software suite, and includes developer support and special licensing geared specifically to ISVs.

The following topics are included:

- [Before You Develop](#)
- [Applications You Can Develop on WebLogic Platform](#)
- [Using WebLogic Platform ISV Edition](#)
- [Using the Partner Software Bundle](#)
- [Next Steps](#)

Before You Develop

Before you begin developing applications with WebLogic Platform ISV Edition or any other WebLogic Platform product you have purchased from BEA, note the following:

- If you are building an application that will be distributed along with WebLogic Platform software, be certain that the application uses only the components of WebLogic Platform that you and your customers are licensed to run. For example, if you are building an

application that runs on WebLogic Platform ISV Edition, make sure that the application does not use features or components of WebLogic Integration or WebLogic Portal that are not included with ISV Edition. (The specific set of features and components included with ISV Edition are identified in “[Using WebLogic Platform ISV Edition](#)” on page 2-5.)

- For an introduction to developing WebLogic Platform applications, see the *BEA WebLogic Platform 8.1 Evaluation Guide*. This document provides a set of comprehensive, hands-on tutorials, supplemented with WebLogic Workshop project files, that walk you through the process of building a variety of WebLogic Platform applications, including Web applications, Web services, and integration and portal solutions. You can download this document and sample application files from the following URL:

<http://dev2dev.bea.com/products/wlplatform81/technicalguides/evalguide81.jsp>

Applications You Can Develop on WebLogic Platform

WebLogic Platform provides the infrastructure you need for creating, testing, and deploying the following types of applications:

- Web services
- Web applications
- Enterprise JavaBeans
- Portals¹
- Enterprise integration applications, including:
 - Business processes
 - Data transformations
 - B2B applications²
 - Adapters to enterprise information system (EIS) applications, including mainframe and other legacy applications
- WebLogic Server applications using the standard J2EE programming model

[Table 2-1](#) provides links to the documentation that explains how to create each type of application.

Table 2-1 Where to Find Information About Building WebLogic Applications

Application	Documentation
Web services	<p>For information about creating Web services, see the following documents:</p> <ul style="list-style-type: none"> • Building Web Services in the <i>WebLogic Workshop Help</i>—Explains how to build Web services using WebLogic Workshop • Programming WebLogic Web Services—Explains how to build Web services using the WebLogic Server programming environment
Web applications	See Developing Web Applications in the <i>WebLogic Workshop Help</i> .
Enterprise JavaBeans	See Developing Enterprise JavaBeans in the <i>WebLogic Workshop Help</i> .
Portal applications ¹	<p>For more information about creating portal applications, see the following topics in the <i>WebLogic Workshop Help</i>:</p> <ul style="list-style-type: none"> • Building Portal Applications • Developing Portal Applications • Creating Portlets • Designer Interface Reference
Enterprise integration applications ²	<p>For more information about creating enterprise integration applications, see the following documents in the <i>WebLogic Workshop Help</i>:</p> <ul style="list-style-type: none"> • Building Integration Applications • Guide to Building Business Processes • Programming Transformations • Using Integration Controls • Tutorial: Building Your First Business Process • Tutorial: Building Your First Data Transformation <p>See also the following WebLogic Integration documents:</p> <ul style="list-style-type: none"> • Introducing Trading Partner Integration • Introducing Application Integration • Developing Adapters • WebLogic Integration Solutions Best Practices FAQ • Deploying WebLogic Integration Solutions

Table 2-1 Where to Find Information About Building WebLogic Applications (Continued)

Application	Documentation
WebLogic Server applications using the J2EE programming model	<ul style="list-style-type: none">• Developing WebLogic Server Applications• Programming WebLogic Enterprise JavaBeans• Programming WebLogic Server J2EE Connectors• Programming WebLogic HTTP Servlets• Programming WebLogic JSP• Programming WebLogic Web Services• Using Applets with WebLogic Server• Programming WebLogic JMS• Programming WebLogic jCOM• Programming WebLogic JDBC• Programming WebLogic JNDI• Programming WebLogic JSP Tag Extensions• Programming WebLogic JTA• Programming WebLogic RMI• Programming WebLogic RMI Over IIOP• Programming WebLogic XML• Programming WebLogic Server for Wireless Services• Programming WebLogic Security• Developing Security Providers for WebLogic Server• Using WebLogic Logging Services• Programming WebLogic Management Services with JMX

1. Not all WebLogic Portal product features are available in WebLogic Platform ISV Edition. For a complete list of features that are available, see “[WebLogic Platform Components Included in ISV Edition](#)” on page 2-7.

2. Not all WebLogic Integration applications can be built using WebLogic Platform ISV Edition. For a complete list of applications you can create, see “[WebLogic Platform Components Included in ISV Edition](#)” on page 2-7.

Using WebLogic Platform ISV Edition

The following topics describe WebLogic Platform ISV Edition, and provide important information regarding installation, licenses, and application development:

- [Product Description](#)
- [WebLogic Platform Components Included in ISV Edition](#)
- [Installing WebLogic Platform ISV Edition](#)
- [WebLogic Platform ISV Edition Licenses](#)
- [Building Portal Applications for Use with ISV Edition](#)
- [Building Process Edition Applications for Use with ISV Edition](#)

Product Description

WebLogic Platform ISV Edition includes the following components of WebLogic Platform:

- WebLogic Server
 - WebLogic Workshop
 - WebLogic JRockit
 - WebLogic Portal's portal framework—Enables rapid development of portal interfaces to Web applications. The portal framework is tightly integrated with the WebLogic Workshop IDE and run-time framework, making it easy to create well-defined portal interfaces from which you can surface other applications and Web services that you develop in WebLogic Workshop.
- Note:** The portal framework included in ISV Edition includes only single file mode support for rendering portal applications to end users. Streaming mode, which enables features such as user customization and entitlements, is not available in ISV Edition.

For more information about using the portal framework, see the following topics:

- Developing Portal User Interfaces
<http://e-docs.bea.com/workshop/docs81/doc/en/portal/buildportals/navUI.html>
- White Paper: WebLogic Portal Framework
<http://e-docs.bea.com/wlp/docs81/whitepapers/netix/index.html>

- WebLogic Server Process Edition—ISV Edition includes all of WebLogic Server Process Edition functionality, which enables the development and deployment of process-driven, service-oriented applications. This package includes the following:
 - Business Process Management (BPM)—Allows you to model and execute business processes that span multiple internal systems, external resources, and users. These business processes can orchestrate the execution of business logic and the exchange of business documents among such systems and users in a loosely-coupled fashion.
 - Data Transformation—Enables you to translate between XML, non-XML, and Java data formats allowing you to rapidly integrate heterogeneous applications regardless of the format used to represent data. The data transformation functionality is available through a control, and data transformations can be packaged as controls and re-used across multiple business processes and applications.
 - Application integration—Enables access to enterprise applications provided via iWay 5.5 Adapters.
 - Connectivity to enterprise resources—Enables out-of-the-box connectivity in to enterprise resources databases, EJBs, JMS, Web services, MQ Series, .NET, Tuxedo, file systems, and e-mail. Includes J2CA-based pre-built adapters to leading enterprise applications and technologies.

This connectivity is enabled by a set of out-of-the box controls that enable you to start integration projects with a portfolio of resources. Integration controls provide easy access to enterprise resources from within your application, and handle the work of connecting to the enterprise resource for you so that you can focus on the logic of your business process.

For more information about the WebLogic Server Process Edition capabilities provided in WebLogic Platform ISV Edition, see the following topics:

- WebLogic Server Process Edition 8.1 Documentation
<http://e-docs.bea.com/wli/docs81/wlspe.html>
- Controls: Service Enablement
<http://e-docs.bea.com/wli/docs81/peoverview/controls.html>
- Business Process Management: Process Driven Services
<http://e-docs.bea.com/wli/docs81/peoverview/bpm.html>
- Data Transformation
http://e-docs.bea.com/wli/docs81/peoverview/data_transformation.html

- Process Monitoring and Management

http://e-docs.bea.com/wli/docs81/peoverview/process_monitoring.html

Note: As mentioned in “Before You Develop” on page 2-1, make sure your application does not use components not included with ISV Edition. If the ISV redistribution license you generate is for WebLogic Platform ISV Edition only, your customers will not be licensed to use WebLogic Platform components that are unavailable in ISV Edition. Table 2-2 identifies the specific components and features in WebLogic Platform that are not available in ISV Edition. For more information about the ISV redistribution license, see “Using an ISV Redistribution License” on page 4-1.

WebLogic Platform Components Included in ISV Edition

Table 2-2 identifies the WebLogic Platform components included with WebLogic Platform ISV Edition, as well as features in WebLogic Integration and WebLogic Portal that are not included.

Table 2-2 Summary of Features in WebLogic Platform ISV Edition

WebLogic Platform Component	Features Included in ISV Edition	Features Not Included in ISV Edition
WebLogic Server	All	Not applicable
WebLogic Workshop	All	Not applicable
WebLogic JRockit	All	Not applicable
WebLogic Portal	<ul style="list-style-type: none"> • Portal user interface framework—Full single file mode run-time support for rendering portal applications to end users, including the support for desktops, shells, books, pages, layout, portlets, skins and skeletons. • Development tools—Support for creating the preceding in the WebLogic Workshop IDE, and includes samples and JSP tags. 	<ul style="list-style-type: none"> • Streaming mode support for rendering portal applications • Administration framework • Content management • Personalization • Interaction management • Behavior tracking • Commerce functionality • Collaboration • Search

Table 2-2 Summary of Features in WebLogic Platform ISV Edition (Continued)

WebLogic Platform Component	Features Included in ISV Edition	Features Not Included in ISV Edition
WebLogic Integration	<p>All features of WebLogic Server Process Edition, including:</p> <ul style="list-style-type: none"> • Stateless processes • Stateful processes • Data transformations • Integration controls: <ul style="list-style-type: none"> - File control - E-mail control - WLI JMS control - Service Broker control - HTTP control - MQSeries Control <p>See “Introduction to WebLogic Server Process Edition” in the <i>WebLogic Server Process Edition Overview</i> at the following URL for a complete list: http://e-docs.bea.com/wli/docs81/peoverview/introduction.html</p>	<p>The following WebLogic Integration features:</p> <ul style="list-style-type: none"> • Message Broker, including: <ul style="list-style-type: none"> - Publish control - Subscribe control • B2B-specific components: <ul style="list-style-type: none"> - ebXML control - RosettaNet control - TPM control - TPM lookup function GUI • Worklist-specific components: <ul style="list-style-type: none"> - Task control - TaskManager control • Application Integration-specific components, including the Application View control

Installing WebLogic Platform ISV Edition

When you receive the WebLogic Platform ISV Edition software, either from CD or the download center, you can follow the instructions provided in *Installing WebLogic Platform* at the following URL:

<http://e-docs.bea.com/platform/docs81/install/index.html>

Caution: If you are installing ISV Edition for the first time, we recommend installing it into a new BEA Home directory separate from the home directory for other BEA software products you might have.

If the home directory into which you plan to install ISV Edition already contains other BEA products, back up the `BEA_HOME\license.bea` file that exists in that directory. (If you plan to install the software bundle into a *different* directory, you do not need to back up that existing `license.bea` file.) For more information about the BEA

Home directory and the `license.bea` file, see “Selecting Directories for the WebLogic Platform Installation” in [“Preparing to Install WebLogic Platform”](#) in *Installing WebLogic Platform*.

WebLogic Platform ISV Edition Licenses

When you enroll in the Partner Program, you receive software and a set of licenses that enable you to build and test the full set of WebLogic Platform applications. These licenses do not limit you to the usage of only those components supported in WebLogic Platform ISV Edition, and there is no specific ISV Edition license that you need to install to develop and run applications for use with ISV Edition.

However, when you sign a BEA standard Channel License Agreement (CLA) to distribute a value-added solution with WebLogic Platform ISV Edition, you obtain the rights to create and distribute licenses that *enable the run-time usage of the components supported in ISV Edition*. It is only via the ISV redistribution license that usage is restricted to only those supported components.

Therefore, it is very important that when you create applications for use with WebLogic Platform ISV Edition that you verify that only the components supported in ISV Edition are used at run-time. For information about how to perform this verification, see [“Using an ISV Redistribution License” on page 4-1](#).

Building Portal Applications for Use with ISV Edition

This section explains how to configure a domain in which you can build and run WebLogic Platform ISV Edition applications that use WebLogic Portal’s portal framework. The following topics are included:

- [File-Based Portals vs. Streaming Portals](#)
- [Creating and Configuring Your Portal Development Environment](#)
- [Testing Your Portal Development Environment](#)
- [Building Your Portals](#)
- [Administering Users of a File-Based Portal](#)
- [Deploying the Portal Application](#)

File-Based Portals vs. Streaming Portals

WebLogic Portal provides two deployment options for portals:

- File-based
- Full-featured, or *streaming*

A file-based portal is a scaled down version of a streaming portal. Unlike a streaming portal, a file-based portal does not deploy any EJBs and does not use a database to store configuration information. Because a file-based portal does not engage a database, it does not support the customization features of a streaming portal, such as the Administration Portal and Visitor Tools.

Note: *Only file-based portals are supported in WebLogic Platform ISV Edition.*

When you view a file-based portal in a browser, the portal is rendered in *single file mode*. In single file mode, the portal is rendered from files on the file system, rather than from a database. The file-based portal's XML file, `.portal`, is parsed and the rendered portal is returned to the browser. Typically, file-based portals are used for development purposes and for static portals that cannot be customized.

File-based portals support all of the functionality offered by the Portal User Interface Framework. This framework provides components for creating and managing the look and feel of a portal and for rendering those components in a browser. For example, the framework supports features such as desktops, shells, books, pages, layouts, skins, skeletons, and portlets. Because a file-based portal uses this framework, developers with an ISV Edition license can use WebLogic Workshop to rapidly develop and render portal interfaces.

Typically, WebLogic Portal administrators and users customize full-featured “streaming” portals using the Administration Portal, Visitor Tools, or the portal framework's customization API. These methods of customization are not available for file-based portals. For example, after a file-based portal is developed using WebLogic Workshop, an administrator cannot use the Administration Portal to add pages or books, or rearrange the layout of the portal. A file-based portal's configuration is fixed and can only be changed by directly editing portal files or by using WebLogic Workshop directly.

The following table compares the features that are supported in full-featured portals with those supported in file-based portals. As noted earlier, only file-based portals are supported in WebLogic Platform ISV Edition.

Table 2-3 Feature Comparison Between Streaming and File-Based Portals

Feature	Supported in Streaming Portals	Supported in File-Based Portals
Supports Portal User Interface Framework (desktops, shells, books, pages, layouts, skins, skeletons, portlets)	yes	yes
Web Services for Remote Portlets (WSRP) support	yes	yes
Interportlet communication	yes	yes
Customization	yes	no
Interaction management	yes	no
Behavior tracking	yes	no
Commerce features	yes	no
Collaboration	yes	no
Searching	yes	no
Content Management	yes	no
Personalization	yes	no

For more information about the Portal User Interface Framework, including details about file-based and streamed rendering, see the following documents:

- The *Portal User Interface Framework Guide* at the following URL:
<http://e-docs.bea.com/wlp/docs81/lookandfeel/index.html>
- *White Paper: WebLogic Portal Framework* at the following URL:
<http://e-docs.bea.com/wlp/docs81/whitepapers/netix/index.html>

Creating and Configuring Your Portal Development Environment

This section explains how to configure your development environment to build file-based portals using WebLogic Workshop.

1. Make sure you have the Portal Examples installed on your machine. The Portal Examples include several JAR and JSP files that need to be copied into the domain you create for your portal development environment. For more information about installing the Portal Examples, see *Installing WebLogic Platform* at the following URL:

<http://e-docs.bea.com/platform/docs81/install/index.html>

2. Start WebLogic Workshop.
3. Start the Configuration Wizard by choosing **Tools→WebLogic Server→Configuration Wizard**, and create a domain that includes the resources required for building a file-based portal.

In the Select a Configuration Template window of the Configuration Wizard, choose one of the following templates:

- If you are building a Workshop application that includes a file-based portal, choose the **Basic WebLogic Workshop Domain** template.
- If you are building an application that includes WebLogic Server Process Edition as well as a file-based portal, choose the **Basic WebLogic Integration Domain** template.

For information about using the Configuration Wizard to create a new domain, see “Creating a New WebLogic Domain” in *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL:

<http://e-docs.bea.com/platform/docs81/configwiz/newdom.html>

4. Restart the Configuration Wizard, and extend the domain to include the WebLogic Portal Extension.

In the Select a Configuration Extension Template window of the Configuration Wizard, choose **WebLogic Portal Extension**.

For information about using the Configuration Wizard to extend a domain via extension templates, see “Extending Domains” in *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL:

<http://e-docs.bea.com/platform/docs81/configwiz/exten.html>

5. In WebLogic Workshop, create a Workshop application that uses the domain by completing the following steps.

- a. In the Workshop main window, choose **File→New→Application...**
This displays the New Application dialog box.
 - b. Select **All** in the upper left list box, and **Default Application** in the upper right list box.
 - c. In the Directory field, click **Browse** to locate the directory that will contain your Workshop application.
 - d. In the Name field, enter a name for the Workshop application you are creating.
 - e. In the Server field, click **Browse** to locate the directory of the domain you created in [step 3](#). In the Select a WebLogic Server config.xml File window, click the `config.xml` file from that directory, and click **Open**.
 - f. In the New Application dialog box, click **Create**.
6. In the Workshop main window, select **Tools→WebLogic Server→Server Properties...**
This displays the Application Properties dialog box.
 7. In the Application Properties dialog box, make sure **WebLogic Server** is highlighted in the column on the left, and that the Server Home Directory field in the upper right displays the directory of the domain you created in [step 3](#).
 8. Scroll down in the Application Properties dialog box to display the list boxes titled **Default server classpath** and **Server classpath additions**.
 9. Make sure the following JAR files are in the server classpath, where *BEA_HOME* represents the BEA Home directory on your machine; for example, C:\bea.

```

BEA_HOME/weblogic81/p13n/lib/p13n_system.jar
BEA_HOME/weblogic81/portal/lib/netuix/system/netuix_system.jar
BEA_HOME/weblogic81/common/lib/log4j.jar
BEA_HOME/weblogic81/server/lib/knex.jar
BEA_HOME/weblogic81/javelin/lib/javelin.jar
BEA_HOME/weblogic81/server/lib/wlw-lang.jar
BEA_HOME/weblogic81/server/lib/debugging.jar
BEA_HOME/weblogic81/server/lib/weblogic.jar
BEA_HOME/weblogic81/server/lib/xbean.jar

```

To add a JAR file to the server classpath, click **Add Jar...**, and select the required file.

10. Using either a command window or Windows Explorer, copy the following JAR files to the `<portalApp>/APP-INF/lib` directory, where `<portalApp>` represents the path of the Workshop application you created in [step 5](#). For example,
C:\bea\user_projects\applications\ISVportalApp.

```
BEA_HOME/weblogic81/samples/portal/portalApp/APP-INF/lib/dom4j-full.jar
BEA_HOME/weblogic81/samples/portal/portalApp/APP-INF/lib/netuix_util.jar
```

11. Using either a command window or Windows Explorer, copy the following JAR files to the `<portalAppWeb>/WEB-INF/lib` directory, where `<portalAppWeb>` represents the path of the Web application you created; for example,
C:\bea\user_projects\applications\ISVportalApp\ISVportalAppWeb.

These JAR files are located in the
BEA_HOME/weblogic81/samples/portal/portalApp/sampleportal/WEB-INF/lib/
directory.

```
client_taglib.jar
html_taglib.jar
l10n_taglib.jar
netuix_servlet.jar
netuix_taglib.jar
render_taglib.jar
```

12. Using either a command window or Windows Explorer, copy the files from the directories listed in the left column of the following table into the corresponding locations in your domain listed in the right column. (You may need to first create empty framework, skins, skeleton, classic, default, and text directories in your domain.)

In this table, `<portalAppWeb>` represents the path of the Web application you created; for example; C:\bea\user_projects\applications\ISVportalApp\ISVportalAppWeb.

Copy all files from this directory ...	To this directory ...
BEA_HOME/weblogic81/samples/portal/portalApp/sampleportal/framework/skins/classic	<portalAppWeb>/framework/skins/classic
BEA_HOME/weblogic81/samples/portal/portalApp/sampleportal/framework/skins/default	<portalAppWeb>/framework/skins/default

Copy all files from this directory . . .	To this directory . . .
<i>BEA_HOME</i> /weblogic81/samples/portal/ portalApp/sampleportal/framework/skins/ text	<portalAppWeb>/framework/skins/ text
<i>BEA_HOME</i> /weblogic81/samples/portal/ portalApp/sampleportal/framework/skeletons/ classic	<portalAppWeb>/framework/ skeletons/classic
<i>BEA_HOME</i> /weblogic81/samples/portal/ portalApp/sampleportal/framework/skeletons/ default	<portalAppWeb>/framework/ skeletons/default
<i>BEA_HOME</i> /weblogic81/samples/portal/ portalApp/sampleportal/framework/skeletons/ text	<portalAppWeb>/framework/ skeletons/text

Note: Do not copy the `beatools` or `avitek` directories into your domain.

13. Copy the four configuration files listed below in [Table 2-4](#) into your Web application directory as follows.

Note: If you are viewing these instructions in a browser, you can easily obtain copies of these files by downloading them from the Web, as explained in the following steps, and saving them in your Web application directory. Otherwise, see [Appendix A](#), “[Configuration Files Required for File-Based Portal Applications](#),” for complete listings of the configuration files, in case you must create them from scratch.

If you are viewing these instructions in a browser, complete the following steps:

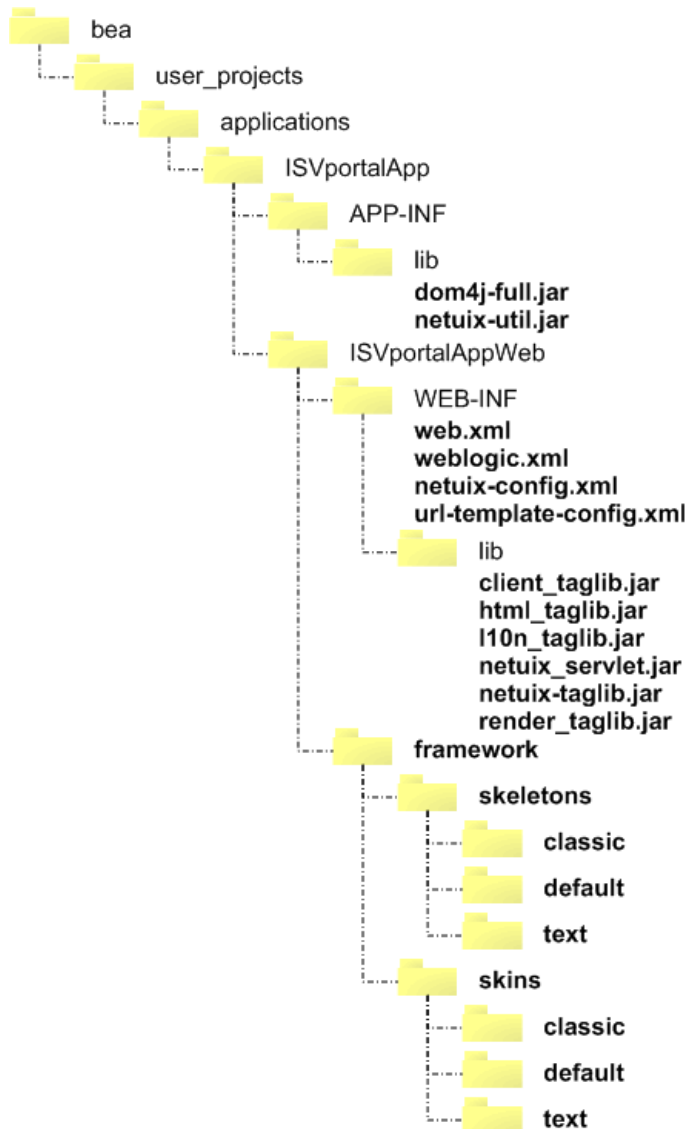
- a. For each configuration file listed in the left column, click the corresponding file location link shown in the right column.
- b. In the popup window that is displayed, choose **File**→**Save As...**
- c. Save the configuration file in the <portalAppWeb>/WEB-INF directory on your machine. The symbol <portalAppWeb> represents the path of your Web application’s directory, which is subordinate to the main Workshop application directory; for example, `C:\bea\user_projects\applications\ISVportalApp\ISVportalAppWeb`.

Table 2-4 Configuration Files to be Saved in the Web Application Directory

To download the following configuration file . . .	Click the following URL . . .
web.xml	http://e-docs.bea.com/platform/docs81/isv/scripts/web.xml
weblogic.xml	http://e-docs.bea.com/platform/docs81/isv/scripts/weblogic.xml
netuix-config.xml	http://e-docs.bea.com/platform/docs81/isv/scripts/netuix-config.xml
url-template-config.xml	http://e-docs.bea.com/platform/docs81/isv/scripts/url-template-config.xml

If you are not viewing these instructions in a browser, you can obtain these configuration files from [Appendix A, “Configuration Files Required for File-Based Portal Applications.”](#)

[Figure 2-1](#) shows the layout of an example domain directory structure into which the files and directories described in this section are copied. The files and directories that are copied are shown in **bold**. (Note that [Figure 2-1](#) does not display the names of the files you copy into the `classic`, `default`, and `text` subdirectories.)

Figure 2-1 Sample Configuration Directory Structure

Testing Your Portal Development Environment

This section explains how to test the portal development environment by creating, deploying, and viewing a test portal.

1. Create a new portal in WebLogic Workshop. To do this, right-click over the Web application name in the Application window. From the menu, select **New→Portal**.
2. Create a JSP/HTML portlet. To do this, right-click over the Web application name in the Application window. From the menu, select **New→Portlet**. Follow the instructions displayed by the Portlet Wizard to create a JSP/HTML portlet.

For information about using the Portlet Wizard to create a portlet, see “Creating Portlets” in the *WebLogic Workshop Online Help* at the following URL:

<http://e-docs.bea.com/workshop/docs81/doc/en/portal/buildportlets/navPortlet.html>

3. Select the portal name in the Application window, then drag the portlet from the Data Palette into the portal.
4. Select **Debug→Start**. This function starts WebLogic Server, deploys the application, and displays it in a browser window.

Note: To verify that your portal application uses only the portal framework components that are supported in ISV Edition, you need to create an ISV redistribution license and validate your application on a system that uses that ISV redistribution license. For more information, see “[Using an ISV Redistribution License](#)” on page 4-1.

Building Your Portals

After your environment is configured and working properly, you can build portal applications. Note that the features available for portal applications built with ISV Edition are listed in [Table 2-3](#) in the column labeled “Supported in File-Based Portals.”

For information about building portal applications, see the WebLogic Portal documentation set at the following URL:

<http://e-docs.bea.com/wlp/docs81/index.html>

For an introduction to portal development, see *Getting Started With Portal Development* at the following URL:

<http://e-docs.bea.com/wlp/docs81/startdev/index.html>

Administering Users of a File-Based Portal

A file-based portal application built with ISV Edition uses the WebLogic Security Service. The administration of users, groups, and roles for file-based portal applications, as with all WebLogic Server applications, can be performed using the WebLogic Server Administration Console.

For information about defining and administering users, groups, and roles, see the following topics:

- To secure application resources, and to create, add, modify, and delete users and groups, see “Users and Groups” in *Securing WebLogic Resources*, at the following URL:
http://e-docs.bea.com/wls/docs81/secwlrres/usrs_grps.html
- To obtain information about security tasks when using the WebLogic Server Administration Console, click the Security node in the appropriate domain tree, then click the Help button in the Console. The Security help topic includes information about tasks such as Defining Groups, Defining Users, and Defining Global Roles.
- To obtain general information about the LDAP security directory in which user and group information is maintained by default, see “Where User Information Is Stored” in “Using an External Store for User Information” in *Security in WebLogic Platform 8.1* at the following URL:

<http://e-docs.bea.com/platform/docs81/secintro/user.html>

For information about running and using the WebLogic Server Administration Console, see “Using the Administration Console” in the *Administration Console Online Help* at the following URL:

<http://e-docs.bea.com/wls/docs81/ConsoleHelp/console.html>

Note: The WebLogic Portal Administration Tools, Visitor Tools, and Portal Framework’s customization API cannot be used to manage a file-based portal.

Deploying the Portal Application

After you develop a portal application using ISV Edition, you need to deliver the completed application to your customers. The distribution model for a file-based portal is the same as any other WebLogic Portal application.

Note: The JAR files, configuration files, and JSP files that are required for the file-based portal environment must be included with the application when you deliver it. These required files were listed previously in the section “[Creating and Configuring Your Portal Development Environment](#)” on page 2-12.

After receiving the completed application, your customers need to prepare the application and deploy it into a production environment. For information about preparing and deploying portal applications, you can refer customers to the *Production Operations User Guide* on the BEA e-docs Web site at the following URL:

<http://e-docs.bea.com/wlp/docs81/prodOps/index.html>

For information about distributing custom applications that are built with and run on ISV Edition, see [Chapter 4, “Distributing WebLogic Platform.”](#)

Building Process Edition Applications for Use with ISV Edition

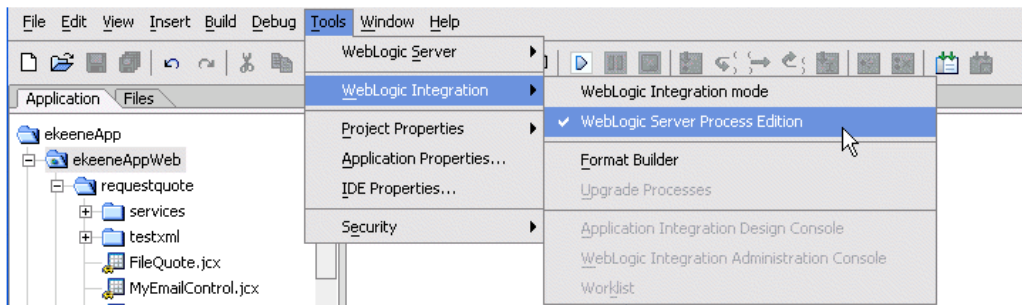
When developing WebLogic Server Process Edition applications in WebLogic Workshop, note that WebLogic Workshop provides a development mode setting that helps prevent you from adding features into your application that are not supported by run-time licenses for the following:

- WebLogic Server Process Edition
- WebLogic Platform ISV Edition

The licenses affected by this restriction include the test licenses that ISVs receive with ISV Edition, and also the ISV redistribution licenses for WebLogic Platform software that is redistributed. (For more information about the ISV redistribution license, see [“Using an ISV Redistribution License”](#) on page 4-1.)

This development mode, called *WebLogic Server Process Edition development mode* in the WebLogic Workshop IDE, disables items in menus, palettes, and other graphical tools that provide access to capabilities in WebLogic Integration that are not supported. This development mode was initially created specifically for WebLogic Server Process Edition, but it useful for ISV Edition applications as well that leverage WebLogic Integration capabilities.

After starting WebLogic Workshop, you can set this development mode by selecting **WebLogic Integration --> WebLogic Server Process Edition** from the Tools menu, as shown in the following figure.



Once you make this change, the new mode of operation is saved as an environment setting.

Caution: The WebLogic Workshop IDE does not have a similar development mode you can use to guide the development of portal applications for ISV Edition.

Using the Partner Software Bundle

As mentioned in [“ISV Resources, Services, and Tools” on page 1-2](#), when you enroll in the BEA Partner Program, you receive an application development and deployment software bundle. The software bundle includes a collection, on CD-ROM, of all major BEA products. The bundle also includes a set of licenses that enable you to develop, deploy, and test WebLogic Platform and Tuxedo applications in the types of environments in which your customers will use them, including configurations that potentially require a great deal of scalability. The usage rights granted by the software bundle is especially useful for creating the demonstrations, prototypes, and proofs-of-concept appropriate for your business.

Software Bundle Contents

The software bundle includes the following:

- WebLogic SDK, Professional Edition 8.1

A full-featured Java development environment that lets you visually build and assemble enterprise-scale Web services, Web applications, JSPs, portals, EJBs, and business process models on WebLogic Platform. WebLogic SDK, Professional Edition 8.1 includes a perpetual development license; developer support; and a special license that enables scale-limited, commercial deployments.

The development license enables you to run WebLogic Platform on a single server machine, with client connections from a maximum of five IP addresses. The scale-limited license enables you to run a single WebLogic Server instance with a server capacity limited to approximately 3 percent of the full capacity of a production server.

- WebLogic Platform test server licenses

Gives you the ability to load the WebLogic Platform software onto a designated number of test servers. Note that although there is no limitation on the number of CPUs per test server, or on the number of server connections, you need to purchase additional test server licenses if your test environment exceeds the number of designated test servers. Use of this license is limited specifically to testing purposes.

- BEA Tuxedo developer licenses

All partners receive ten developer licenses for BEA Tuxedo with Jolt, also limited specifically to development and testing purposes.

For comprehensive information about the contents of the software bundle, enrolled partners can access the *BEA Partner Program Guide* at the following URL:

ftp://pdownload:BUY_ME@ftpn2.bea.com/pub/program/BEA_Partner_Program_Guide.pdf

Installing the Software Bundle

When you receive your software, install the software from the CDs. For installation instructions, see [Installing WebLogic Platform](#).

Caution: If the home directory into which you plan to install the software bundle already contains other BEA products, back up the `BEA_HOME\license.bea` file that exists in that directory. (If you plan to install the software bundle into a *different* directory, you do not need to back up that existing `license.bea` file.) For more information about the BEA home directory and the `license.bea` file, see “Selecting Directories for the WebLogic Platform Installation” in [“Preparing to Install WebLogic Platform”](#) in *Installing WebLogic Platform*.

Downloading BEA Software

To start using BEA software immediately (instead of waiting for delivery of CDs), you can download it from the BEA Systems Download Center:

<http://commerce.beasys.com>

If you have an active WebSUPPORT account, use your WebSUPPORT login password for software downloads.

Software Products Not Included in the Software Bundle

Depending on your business, you might be interested in redistributing other BEA software that is not included in the software bundle. If you are interested in redistributing such software, contact your BEA account representative for more information.

Next Steps

After you have installed WebLogic Platform and built the WebLogic applications that you plan to distribute, complete the following tasks:

1. Configure a domain in which your applications can run. See [Chapter 3, “Configuring and Installing WebLogic Platform Applications.”](#)

2. Create a template of the domain created in the previous step. See [Chapter 3, “Configuring and Installing WebLogic Platform Applications.”](#)
3. Distribute your files. See [Chapter 4, “Distributing WebLogic Platform.”](#)

Configuring and Installing WebLogic Platform Applications

WebLogic Platform provides a rich set of system installation, configuration, and administration tools that enable you to install and configure a custom environment in which WebLogic applications can run. Whether you are redistributing WebLogic Platform with your applications, or distributing an application that runs on WebLogic Platform, you will want to use these tools to package a product that your customers can install and use easily.

This chapter introduces tools for the following tasks:

1. Configuring a domain in which your WebLogic applications can run
2. Creating a template or extension template that your customers can use to reproduce that domain in their environments
3. Installing WebLogic Platform and configuring your customers' environments

Configuring WebLogic Platform for Your Application

The configuration tasks that you or your customers undertake may include the following:

- Configuring new domains to which your application can be added and deployed
- Configuring new domains that include your application and the appropriate configuration settings for WebLogic Platform
- Modifying an existing WebLogic Platform domain so that your application can be deployed on it
- Modifying an existing domain by adding your application to it

As a partner, accommodating these tasks typically requires you to do the following:

- Create the domain template or extension template that your customers need for your software product. A template is a file used for creating a new domain, and an extension template is a file used for modifying an existing domain.
- Bundle this template or extension template in your product package.
- Document how your customers can obtain the template or extension template and then use it to create or modify a domain, as appropriate.

The remaining sections of this chapter introduce two sets of tools: configuration tools for creating a domain, a domain template, or an extension template, and tools for incorporating templates or template extensions into your customers' environments.

Configuration Wizard

As summarized in [“Configuration Wizard and Template Builder” on page 1-7](#), the Configuration Wizard is a stand-alone Java application that can be run independently of WebLogic Server to simplify the jobs of creating, configuring, and customizing a WebLogic domain. A domain is the basic unit of administration in a WebLogic Platform environment: it comprises the WebLogic Platform components, user-written programs, machines, and resources, such as databases, needed for a particular application.

Typically, the Configuration Wizard is used to:

- Create a new domain for stand-alone servers, Administration Servers with Managed Servers, and clustered servers.
- Create a new domain based on a template. Templates allow you to recreate an existing domain for use in another context, such as migrating a domain containing an application under development to a production environment.
- Extend an existing domain, by using an extension template, to include new components, such as JMS configurations, JDBC connection pools, MultiPools, and DataSources.

For layered ISVs this Configuration Wizard capability is useful because it enables you to add your applications to existing customer domains.

Creating Domains for WebLogic Platform ISV Edition

If you are creating a domain for developing or running applications for use with WebLogic Platform ISV Edition, use the following domain templates as appropriate.

To build and run applications that use features in . . .	Use the following domain template . . .
WebLogic Server Process Edition only	Basic WebLogic Integration Domain
WebLogic Portal's portal framework only	Basic WebLogic Workshop Domain Note: Follow the instructions provided in “Creating and Configuring Your Portal Development Environment” on page 2-12 to create this domain.
WebLogic Process Edition and portal framework	Basic WebLogic Integration Domain Note: Follow the instructions provided in “Creating and Configuring Your Portal Development Environment” on page 2-12 to create this domain.

For More Information

For complete details about using the Configuration Wizard, see [Creating WebLogic Configurations Using the Configuration Wizard](#).

For complete details about WebLogic domains, see the following:

- “Brief Introduction to Domains” in [“Overview of the WebLogic Configuration Wizard and Configuration Template Builder”](#) in *Creating WebLogic Configurations Using the Configuration Wizard*
- [“Overview of WebLogic Server Domains”](#) in *Configuring and Managing WebLogic Server*

Template Builder

The Template Builder is a standalone Java application that prompts you to create custom configuration and extension templates that can be used later for creating and updating domains via the Configuration Wizard. Using an existing domain or template, the Template Builder guides you through the process of creating custom configuration and extension templates. You can then package these templates or extension templates with your software so your customers can use them to customize their WebLogic Platform environments.

The Template Builder is especially useful for ensuring that the required environment for WebLogic-based products can be installed and configured easily by your customers at their sites.

For complete details about the Template Builder, see the following topics from *Creating WebLogic Configurations Using the Configuration Wizard*:

- [“Creating Configuration Templates Using the WebLogic Configuration Template Builder”](#)
- [“Creating Extension Templates Using the WebLogic Configuration Template Builder”](#)

WebLogic Platform Administration Tools

WebLogic Platform provides a number of administration tools, in addition to the Configuration Wizard, that you can use to customize the environment in which you run WebLogic Platform or any of its components. This section briefly describes some of these tools.

WebLogic Server Administration Console

The WebLogic Server Administration Console is a browser-based, graphical user interface that you can use to manage a domain and, as an alternative to the Configuration Wizard, to modify a domain. As a Web application, it is accessible from any supported browser with network access to the Administration Server, on which it resides.

The Administration Server for a domain is the central point from which that domain is managed. In every domain, one instance of WebLogic Platform in each domain is configured to serve this function. All other WebLogic Platform instances in a domain are called Managed Servers. In a domain with only a single WebLogic Platform instance, that server functions as both the Administration Server and a Managed Server.

You can use the WebLogic Server Administration Console to:

- Configure, start, and stop WebLogic Platform instances
- Configure WebLogic Platform clusters
- Configure WebLogic Server Services, such as database connectivity (JDBC) and messaging (JMS)
- Configure security parameters, including managing users, groups, and roles
- Configure and deploy your applications
- Monitor server and application performance
- View server and domain log files
- View application deployment descriptors

- Edit selected run-time application deployment descriptor elements

For complete details about the WebLogic Server Administration Console, see [Administration Console Online Help](#).

WebLogic Integration Administration Console

The WebLogic Integration Administration Console is a browser-based, graphical user interface that you can use to manage and monitor the resources required for your WebLogic Integration applications. The job of managing and monitoring these resources involves the following tasks:

- Business process management and monitoring: deploying and configuring business process types, viewing statistics on running workflow instances, viewing lists of filter instances, and terminating, deleting, and suspending business process instances
- Message broker monitoring (not supported in ISV Edition)
- System and security configuration for users and groups who use WebLogic Integration applications, including trading partners
- User management, monitoring of Application Integration activity through the Application Integration Console, Worklist administration, B2B process management (not supported in ISV Edition), and trading partner management (not supported in ISV Edition)

Note: Not all functionality in the WebLogic Integration Administration Console is available with WebLogic Platform ISV Edition. For a complete list of functionality you can use from this tool, see “[WebLogic Platform Components Included in ISV Edition](#)” on [page 2-7](#).

For more information, see [Managing WebLogic Integration Solutions](#).

WebLogic Administration Portal

Portal administration involves many traditional system administration activities, as well as some relatively newer tasks, such as controlling the behavior, content, and appearance of portals. While portal administrators do not typically develop the resources required for a portal Web site, they use those resources to build, maintain, and modify the portals posted there.

Note: **The WebLogic Administration Portal is not available with WebLogic Platform ISV Edition.**

To help you manage portal Web sites built with BEA WebLogic Platform, the WebLogic Administration Portal provides you with tools for various functions, including the following:

- Portal Creation—For generating portals, desktops, books, pages, portlets, and other portal resources
- Content Management—For controlling both content and repositories for it
- User Management—For direct administration of users and groups, delegated administration, and entitlements
- Interaction Management—For administrations of Campaigns, Placeholders, Content Selectors, and User Segments
- Server Administration—For setting up servers
- Search—For searches within your WebLogic Administration Portal

WebLogic Portal also provides third-party administration tools, available from a user interface in WebLogic Portal, that allow you to manage various applications. For more information, see the [WebLogic Administration Portal Online Help](#).

Startup Scripts

WebLogic Platform provides several techniques for starting server instances and clusters, all of which can be encapsulated in WebLogic Server startup scripts that are installed with the domain template. Your choice of technique for starting an embedded server depends on the complexity of the WebLogic Platform domain that you install, the number of computers that host server instances within the domain, and the underlying operating system.

For example, which startup script you use may depend on whether the domain in the target environment is characterized by one of the following:

- The domain runs on a single WebLogic Platform host.
- The domain includes multiple server instances and you want them to run on multiple WebLogic Platform hosts. Such a domain may include clusters, which provide scalability and failover capabilities.
- The domain runs on a platform that includes the Microsoft Windows operating system.

For more information about startup scripts, see the following topics:

- “Starting Managed Servers From a WebLogic Server Script” in the *Administration Console Online Help*

- [“Setting Up a WebLogic Server Instance as a Windows Service”](#) in *Configuring and Managing WebLogic Server*

Silent-Mode Installation and Configuration

After you create the domain templates or extension templates appropriate for your application, you need to consider how you want your customers to install WebLogic Platform (or selected components) and configure their environments. This section introduces WebLogic Platform utilities that can help you provide your customers with a simplified, easy-to-use method of configuring their environments:

- [Silent-Mode Installation](#)
- [Silent-Mode Domain Configuration](#)
- [Running Silent-Mode Installation and Configuration from a Script](#)

Silent-Mode Installation

Silent-mode installation is a non-interactive method of installing software on one or more machines. It is accomplished via a script that specifies the installation options that users provide during an interactive installation. A silent-mode installation can be performed on both Windows and UNIX systems.

When a customer uses silent-mode, the installation program reads the values for various installation options from the silent-mode script, instead of prompting the customer to enter values in real time. Once you set the required configuration options in a script, the same script can be used for multiple installations on different machines.

For more information about silent-mode installation, see [“Installing WebLogic Platform Using Silent-Mode Installation”](#) in *Installing BEA WebLogic Platform*.

Silent-Mode Domain Configuration

As a partner, you may especially appreciate one very useful feature of the Configuration Wizard: its ability to run in silent mode. When run in this way, the Configuration Wizard is executed as a non-interactive process.

Silent-mode configuration enables you to define configuration information for a domain once and then duplicate that domain on other machines. Thus, by using this feature, you can easily duplicate a domain on customers' machines.

In silent mode, the Configuration Wizard reads configuration settings from a script that you create, manually, before execution. Before your customers can run the Configuration Wizard in silent mode, however, you must first create a script in which you define the settings usually entered manually by a customer during GUI-mode or console-mode configuration. When running in silent mode, the Configuration Wizard queries the specified configuration script to obtain the configuration settings it needs. It does not prompt the user to provide information in real time.

For more information about silent-mode domain configuration, see [“Creating a Script for Silent-Mode Configuration”](#) in *Creating WebLogic Configurations Using the Configuration Wizard*.

Running Silent-Mode Installation and Configuration from a Script

When you combine silent-mode installation with silent-mode configuration, you have a powerful method that lets you install your WebLogic product and configure a domain for it in a way that is transparent to customers. When your customers run a silent-mode installation and configuration, they experience the work of setting up an environment as a seamless process.

Note that a silent-mode installation script cannot invoke the Configuration Wizard. However, you can create a simple shell script or command file that sequentially runs both the WebLogic Platform installer and the Configuration Wizard in silent mode. For details about silent installation and configuration, see the following:

- [“Installing WebLogic Platform Using Silent-Mode Installation”](#) in *Installing BEA WebLogic Platform*.
- [“Creating a Script for Silent-Mode Configuration”](#) in *Creating WebLogic Configurations Using the Configuration Wizard*

Distributing WebLogic Platform

This chapter provides details about the following topics:

- [Using an ISV Redistribution License](#)
- [Distributing Files](#)
- [Service Packs and License Files](#)

Using an ISV Redistribution License

This section includes the following topics, which describe the contents and terms of an ISV redistribution license, and how to install the license:

- [About the ISV Redistribution License](#)
- [ISV Redistribution License Technical Enforcement](#)
- [Creating an ISV Redistribution License and Linking It to the WebLogic Platform Files](#)
- [Validating the ISV Redistribution License File](#)

About the ISV Redistribution License

The WebLogic Platform package available from BEA normally includes a license that prohibits the buyer from redistributing the software. However, for partners—Independent Software Vendors (ISVs) in the business of redistributing WebLogic Platform—BEA provides a mechanism that enables you to create a redistribution license. Although the purpose of all ISV redistribution licenses is the same (to give an ISV's customers the right to run the WebLogic

Platform software that has been redistributed by the ISV), each ISV redistribution license is unique, linked specifically to each partner.

If you will be redistributing one or more components of WebLogic Platform, you need to sign a BEA standard Channel License Agreement (CLA), available from your BEA partner account representative. The terms of the CLA define how and when the ISV redistribution license will be distributed. After you sign the CLA, BEA delivers you an `isv.jar` file, which enables you to create two things:

- An ISV redistribution license, which is the ISV customer's run-time license to use the redistributed WebLogic Platform software
- A customized `weblogic.jar` file, which specifically links the WebLogic Platform software to the ISV redistribution license

To run the ISV's value-added solution with the redistributed WebLogic Platform software, the ISV customer must have both the ISV redistribution license and the `weblogic.jar` file properly installed on their systems.

To learn how to create this license and the `weblogic.jar` file, see [“Creating an ISV Redistribution License and Linking It to the WebLogic Platform Files” on page 4-3](#).

ISV Redistribution License Technical Enforcement

The ISV redistribution licenses that you create and distribute enable your customers to use WebLogic Platform functionality as follows:

- The specific set of WebLogic Platform components to which your customers can have access—for example, WebLogic Server, or WebLogic Platform ISV Edition—is specified in your license distribution agreement with BEA and is enforced by the ISV redistribution license. This mechanism ensures that run-time support is restricted to the WebLogic Platform components specified in that agreement.
- If your license distribution agreement is for WebLogic Platform ISV Edition only, the ISV redistribution license prevents the run-time use of WebLogic Platform components that are excluded from ISV Edition, such as the WebLogic Administration Portal tools and the WebLogic Integration Message Broker.
 - For a list of the components included, and not included, in WebLogic Platform ISV Edition, see [“WebLogic Platform Components Included in ISV Edition” on page 2-7](#).
 - For details about developing applications for WebLogic Platform ISV Edition, and ensuring that those applications do not use functionality that is made unavailable at run

time by the ISV redistribution license, see [“Building Process Edition Applications for Use with ISV Edition” on page 2-20](#).

- There is no enforced limit on the number of IP or databases connections that can be made by an installation of the WebLogic Platform software distributed by the ISV.
- There is no expiration date on the use of the WebLogic Platform software distributed by the ISV.
- The ISV redistribution license is not tied to any specific IP address. Therefore, the WebLogic Platform software that you distribute with an ISV redistribution license can be used on any machine.

The technical enforcements described in the preceding list may in some ways be less restrictive for your customer than the terms of your license agreement with BEA. For example, you are constrained to use your WebLogic Platform software only on the specific machines with which you are licensed with BEA. However, there is no technical enforcement in the ISV redistribution license that constrains the specific machines on which your customers may run the redistributed software they receive from you.

The ISV redistribution license is intended to restrict the usage of the WebLogic Platform components to the usage required by the ISV’s software, in accordance with the terms of the ISV’s redistribution agreement with BEA. An ISV redistribution license is not intended to grant the ISV’s customer unlimited rights to use WebLogic Platform software independently from the usage of the ISV’s application.

Creating an ISV Redistribution License and Linking It to the WebLogic Platform Files

To create an ISV redistribution license and link it to the WebLogic Platform files, complete the following two-step procedure:

- [Step 1: Prepare to Create an ISV Redistribution License](#)
- [Step 2: Extract the License Data and Link the WebLogic Platform Files](#)

Step 1: Prepare to Create an ISV Redistribution License

Before you create an ISV redistribution license file, complete the following steps:

1. If you have not already done so, install your WebLogic Platform software as described in *Installing BEA WebLogic Platform* at the following URL:

<http://e-docs.bea.com/platform/docs81/install/index.html>

2. Note the location of the BEA Home directory on your system. This directory contains the file `license.bea`, which is updated during subsequent steps in this procedure. For more information about the BEA Home directory and the `license.bea` file, see “Selecting Directories for the WebLogic Platform Installation” in “[Preparing to Install WebLogic Platform](#)” in *Installing BEA WebLogic Platform*.
3. Copy the `isv.jar` file you received from BEA into the BEA Home directory associated with this installation.
4. Open a command shell and go to the `server/bin` directory for WebLogic Platform; for example:

```
prompt> cd BEA_HOME/weblogic81/server/bin
```

Here, `BEA_HOME` is the environment variable representing the BEA Home directory on your system.

5. Run the following script:
 - **Windows:** `setWLSEnv.cmd`
 - **UNIX:** `setWLSEnv.sh`
6. Add `isv.jar` to your system `CLASSPATH` by entering the appropriate command:
 - **Windows:** `set CLASSPATH=.\isv.jar;%CLASSPATH%`
 - **UNIX:** `export CLASSPATH=./isv.jar:$CLASSPATH`

Step 2: Extract the License Data and Link the WebLogic Platform Files

To extract the ISV redistribution license data and associate it with your WebLogic Platform files, complete the following steps:

1. Go to the BEA Home directory on your system:

```
prompt> cd BEA_HOME
```

2. Enter the following command:

– **Windows:**

```
java -Xmx128m -Dbea.home=%BEA_HOME%
-Dbea.jar=%BEA_HOME%\weblogic81\server\lib\weblogic.jar install
```

– **UNIX:**

```
java -Xmx128m -Dbea.home=$BEA_HOME
-Dbea.jar=$BEA_HOME/weblogic81/server/lib/weblogic.jar install
```

Caution: Do not interrupt this process once it starts!

This command does two things:

1. Generates a file named `license_isv.bea` in the BEA Home directory. The `license_isv.bea` file contains the ISV redistribution license data.
2. Modifies the `weblogic.jar` file in your WebLogic Platform installation to create a specific linkage between that file and the ISV redistribution license data in the `license_isv.bea` file.

Only the `weblogic.jar` file in the BEA Home directory tree that you processed with this command can use the ISV redistribution license data extracted to the `license_isv.bea` file. The linkage and association between those two files is unique.

Note: On some operating systems and JDKs, you might encounter an “Out of Memory Error” message. To address this error, increase the value for the `-Xmx` argument, which sets the maximum heap size in megabytes, and run the command again. For example, if you specify `-Xmx150m` on the command line, the default heap size is increased to 150 MB.

Validating the ISV Redistribution License File

To validate that the updated `weblogic.jar` file works with the newly generated `license_isv.bea` file, which you created in [“Step 2: Extract the License Data and Link the WebLogic Platform Files” on page 4-4](#), complete the following steps:

1. Open Windows Explorer or a command shell and navigate to the BEA Home directory from which you generated the ISV redistribution license, typically `c:\bea`.
2. Make a backup copy of your `license.bea` file. For example, at the command prompt, enter the following command:

```
c:\bea> RENAME license.bea license.bea.backup
```

3. Rename the ISV redistribution license file `license_isv.bea` to `license.bea`. For example, at the command prompt, enter the following command:

```
c:\bea> RENAME license_isv.bea license.bea
```

4. Test your value-added solution in this BEA Home directory to ensure that no error conditions associated with license usage are generated.

Distributing Files

After you have validated that your value-added solution operates properly with the WebLogic Platform software that you will be redistributing, you need to create a process for ensuring that your customers install the following into the proper locations on their system:

- The ISV redistribution license
- The `weblogic.jar` file associated with the ISV redistribution license

You have a variety of choices for ensuring that the preceding two files are installed properly by your customers. For example, you can create a custom installer, or a silent-installation script that can be used by the WebLogic Platform installation program. However, note that there may be circumstances in which your customers might need to replace a pre-existing BEA license file with the ISV redistribution license. In these cases, make certain that your customers use the proper installation method, or follow documentation that you provide, to ensure that all necessary files are installed properly.

If your installation process does not result in both files being installed correctly, your customer will not be able to start your embedded WebLogic Platform software.

Note: You can use the same two files for all customer installations.

Service Packs and License Files

As of WebLogic Platform 8.1 Service Pack 2, the `license.bea` file that you generate and distribute, as described in [“Using an ISV Redistribution License” on page 4-1](#), will continue to be valid for any customers who later use the Upgrade Installer to upgrade their installations with service packs for WebLogic Platform. As a result, customers who install your distribution of the initial release of WebLogic Platform 8.1 can later use the Upgrade Installer to install Service Pack 2 without obtaining a new license.

If, however, you provide an upgrade to your existing customers by supplying a whole new version of WebLogic Platform that includes a service pack, then you must generate a new `license.bea` file and include it in your distribution.

Creating and Packaging WebLogic Workshop Java Controls

This chapter includes the following topics:

- [Java Controls Overview](#)
- [Building Advanced Controls](#)
- [Creating Online Help](#)
- [Creating a Control Archive](#)

Java Controls Overview

WebLogic Workshop provides Java controls: software modules that give developers the ability to easily connect with existing data, systems, applications, and business logic. Java controls are visual components with methods and properties that meet all of the J2EE requirements governing how connections are made to an external source or piece of business logic. Developers interact with controls by handling events and setting properties.

In addition to the Java controls embedded in WebLogic Workshop 8.1 itself, Workshop 8.1 offers an architecture based on *extensible* controls. Extensible controls enable developers to create reusable components of business logic as modules that can be integrated with any enterprise resource, ISV application, or piece of business logic. Custom Java controls can leverage native Workshop Java controls. They can also be nested and easily reused. Moreover, any Java control can be incorporated in any project that spans multiple Web services, Web applications, and other WebLogic Platform applications. The process of creating a custom control is primarily one of declaratively specifying control behavior, and then focusing on handling events and calling

methods with standard procedural Java code. This process does not require you to learn a new API.

Custom Java Controls and Workshop's Extensibility Model

Custom Java controls play an important role in Workshop's extensibility model, which provides a means for you to enhance Workshop's integrated environment for building Web applications efficiently. Through the Workshop extensibility model, you can enhance this environment to provide support for application and productivity needs that are not yet addressed by WebLogic Workshop.

Ways You Can Extend Workshop's Environment

Workshop's extensibility model encompasses four main areas:

- *Custom controls* that expose complex functionality in simple ways through customizable interfaces, properties, and control-specific user interface components. Controls are used as components of applications built with WebLogic Workshop.
- *Templates* that set up the initial structure for new projects, adding dependencies that might otherwise have to be imported or created, and generating files that help the project's user get started quickly. Templates enhance productivity.
- *Support for custom JSP tag libraries through tag library extensions* that integrate the tags seamlessly with the Workshop design-time environment. Tag library extensions enhance productivity.
- *Components that augment the IDE* with toolbar buttons and menus, views in dockable frames, additional document and project types, editing support for various kinds of source code, source control integration, debugging tools for data views, and so on.

In addition, the extensibility model provides a means by which you can deliver online help that is both specific to your extension and fully integrated with the WebLogic Workshop help system.

For more information about Workshop's extensibility model, see [“Extending WebLogic Workshop”](#) in the *WebLogic Workshop Help*.

Extensibility Portal

BEA's Extensibility Portal contains the latest updates to the Extensibility Development Kit, (described in [“Building Advanced Controls” on page 5-4](#)), along with updates to the documentation and samples, and other useful resources. Visit the Extensibility Portal at the following URL:

http://dev2dev.bea.com/products/wlworkshop81/workshop_ext.jsp

Note that one valuable resource at the Extensibility Portal of interest to ISVs is the Workshop Controls and Extensibility Program. This program provides a framework for assisting partners in building, testing, distributing, and marketing custom controls and IDE extensions. More information about the Workshop Controls and Extensibility Program is available at the following URL:

http://dev2dev.bea.com/products/wlworkshop81/ext_overview.jsp

Key Benefits of Java Controls to ISVs

For ISVs, the use of Java controls is beneficial because these controls:

- Describe themselves at design time, giving developers a visual representation of server-side business logic in the familiar Workshop development environment
- Encapsulate best practices for accessing resources and application logic, ensuring that resources can be used with optimal efficiency
- Expose a set of simple properties by which a user can implement advanced run-time functionality, such as asynchronous communication, security roles, lifecycle events, transaction support, and so on. Authors of custom controls have the option of populating the Properties sheet for the benefit of downstream users of the controls.
- Make it easy for ISVs to package and distribute them as standard JAR files. Controls can easily be discovered and made available to end users, even before the control JAR files have been completely downloaded. Moreover, because Java controls can be nested, they can easily be re-used. Java controls make it possible to connect to any IT asset, ISV application, or piece of business logic, while protecting the developer's abstraction from most of the complexity of the infrastructure. As a result, developers can largely focus on procedural Java instead of J2EE or other vendor-specific APIs.

How to Make Custom Java Controls Available to End Users

To make it easy for ISVs to deliver custom Java controls to end users, BEA provides the following services and features:

- When you create custom controls, you can specify how the controls are displayed on the WebLogic Workshop Insert menu. For example, if you create two or more controls, you may want to group them together in the Insert menu.
- WebLogic Workshop tools make it easy to create ZIP files containing all the files, including online help, required for your controls.

- BEA hosts the ComponentSource depot, an online facility at the dev2dev Web site, at which ISVs can make their custom controls available to end users for just-in-time downloading.
- WebLogic Workshop provides a Help menu item labeled Premiere Component Gallery, which provides a link to the preceding Web site. End users can browse the component gallery to select the ISV controls they want to download.
- WebLogic Workshop makes it easy for users who download your controls to install the controls and incorporate online help for them into the WebLogic Workshop help system.

For More Information About Java Controls

For a complete introduction to Workshop controls, see the following topics in the *WebLogic Workshop Help*:

- [Working with Java Controls](#)
- [Using Built-In Java Controls](#)
- [Building Custom Java Controls](#)
- [Tutorial: Java Control](#)

For the latest samples and documentation updates for building extensions to WebLogic Workshop, visit the Extensibility Portal at the following URL.

http://dev2dev.bea.com/products/wlworkshop81/workshop_ext.jsp

Building Advanced Controls

The Extensibility Development Kit, provided by WebLogic Workshop, includes a package called the Control Development Kit. The latter kit provides tools for incorporating advanced features into Java controls. For example, you might use the tools in the Control Development Kit to make your Java controls highly customizable by end users, or to provide sophisticated property editors that are not required by all applications.

The Control Development Kit includes sample code and documentation that demonstrate, as well as describe, the advanced features of the Java control architecture. Using the samples as a guide, you can build controls with increasingly sophisticated capabilities. For example, your custom controls might:

- Define properties that can be mapped to the Properties Editor, which is particularly useful for custom data types

- Use a custom control insert wizard, which guides the end control user through the process of adding the control to the application and specifying required properties.
- Use a class that you create to validate custom property attributes in both the Property Editor and in source code
- Include code that handles specific user actions in the Design View
- Handle lifecycle callback events for efficient management of resource dependencies
- Include branding information

After you install WebLogic Platform, the Control Development Kit is located in the following directory:

`BEA_HOME/weblogic81/samples/workshop/ExtensibilityDevKit/ControlDevKit`

For complete details about using the Control Development Kit, see “[Developing Advanced Controls](#)” in the *WebLogic Workshop Help*.

Note: The Extensibility Development and Control Development Kits are delivered in WebLogic Platform 8.1 Service Pack 2 and later. However, we recommend you visit the Extensibility Portal to obtain the latest versions of those kits, along with the latest samples and updates to the documentation.

Creating Online Help

Depending on the nature or complexity of your control or IDE extension, you may choose to include online help. WebLogic Workshop includes a documentation system that enables you to add new content to support that control or extension. When you follow our guidelines for delivering help, you ensure that your documentation is integrated with the table of contents and search engine for the existing help. If you are delivering controls, you also guarantee that when a user selects your control in Source or Design View, and then presses F1, your documentation is available.

The topic “[Help Authoring Guide](#)” in the *WebLogic Workshop Help* explains how to do the following tasks for the documentation you provide for your controls and extensions.

- Organize the files in your help system
- Create individual HTML pages to be added to the help system
- Make your help searchable by the built-in WebLogic Workshop online help search engine

- Add an index to your help files and list it in the WebLogic Workshop online help table of contents

The Help Authoring Guide includes links to a stylesheet you can use for creating your help files in HTML format, and your table of contents, in XML.

Creating a Control Archive

WebLogic Workshop makes it easy to package two types of custom controls:

- Control deliverables—JAR files containing the code that implements the control, along with the help files and sample files for the control
- Control stubs—Placeholders for control deliverables that are dynamically downloaded and installed the first time an end user adds them to an application

The following sections describe these two control types in detail.

Control Deliverables

Control deliverables are archive files that contain the top-level directories described in the following table.

Table 5-1 Control Deliverable Directories

This directory . . .	Contains . . .
controls	Control implementation archive and any dependency archive JARs
help	Documentation and Javadoc for the control
samples	Samples that utilize the control

Note that while only a *single* control implementation JAR can exist in the `controls` folder, multiple controls may be bundled into that JAR. The first time an end user tries to install your custom control into any application, the control implementation archive and all its dependency archives are copied to the application's `Libraries` folder.

Once your control implementation files are complete, you right-click the name of the control project folder, then click Build Control Deliverable. WebLogic Workshop collects all the required files into a ZIP file and places that file in the application folder.

For more information about how to package control deliverables, see the following:

- The topic “[Packaging Controls for Installation](#)” in the *WebLogic Workshop Help*
- The Extensibility Portal, at the following URL, for the latest documentation updates:
http://dev2dev.bea.com/products/wlworkshop81/workshop_ext.jsp

Control Stubs

Control stubs allow you to make custom controls available to end users without requiring them to install the implementation files for those controls on their machines. Custom controls that can be accessed through stubs are listed on the menu displayed by the Workshop when you choose Insert→Controls. When an end user selects a control from this menu, the implementation files for that control are downloaded to the user’s machine. Thus, the difference between control deliverables and control stubs is transparent to the end user.

With control stubs, you can deliver controls *just in time* to your end users. As an ISV, you can take advantage of this capability to keep the controls offered at your Web site up to date. Customers who download your controls through stubs are guaranteed to receive the latest versions of your control implementations.

Note: If an end user who has downloaded a control through a control stub later wants a subsequent version, the user must reinstall the control, using the newer version.

For more information about creating control stubs, see “Using Control Stubs as Placeholders” in “[Packaging Controls for Installation](#)” in the *WebLogic Workshop Help*.

Supporting and Distributing WebLogic JRockit

This chapter includes the following topics:

- [About WebLogic JRockit](#)
- [Obtaining WebLogic JRockit](#)
- [Support for WebLogic JRockit in WebLogic Platform Applications](#)
- [Distributing the JRockit JVM as a Standalone Product](#)

About WebLogic JRockit

The BEA WebLogic JRockit SDK provides a complete run-time environment, including all the tools and utilities needed to develop and run applications using the Java programming language. It includes the WebLogic JRockit Java Virtual Machine (JVM).

The WebLogic JRockit Java Virtual Machine (JVM) is the first commercial JVM developed expressly for server-side applications. As a crucial component of WebLogic Platform, WebLogic JRockit is optimized for high-performance Java applications deployed on 32-bit and 64-bit Intel architecture systems. Furthermore, WebLogic JRockit provides seamless interoperability across multiple hardware and operating system configurations. WebLogic JRockit makes it possible to gain optimal performance for your server-side application when running it on either the Windows and Linux operating systems and on either 32-bit and 64-bit architectures.

For complete details about WebLogic JRockit, see the following documents:

- [Introduction to WebLogic JRockit SDK](#)
- [Using WebLogic JRockit SDK](#)
- [Tuning WebLogic JRockit JVM](#)
- [Developing Applications](#)
- [Using the Monitoring and Management APIs](#)
- [WebLogic JRockit SDK Release Notes](#)

For a complete list of hardware/operating system configurations on which WebLogic JRockit is supported, *BEA WebLogic JRockit 1.4.2 SDK Platform Support* at the following URL:

<http://e-docs.bea.com/wljrockit/docs142/certif.html>

Obtaining WebLogic JRockit

You can obtain the BEA WebLogic JRockit 1.4.2 SDK simply by downloading it from the following Web site:

<http://commerce.bea.com/showproduct.jsp?family=WLJR&major=8.1&minor=0>

If you want to redistribute this product, you must sign a contract called the *BEA WebLogic JRockit Binary Re-Distribution Agreement*. This agreement entitles you to obtain the JRockit software at no charge and re-sell it to customers, with or without your own software.

The *Binary Re-Distribution Agreement* also includes support contracts for BEA partners. The cost of this type of support is calculated from two factors: anticipated redistribution volume and the type of support structure implemented by the partner.

To obtain a copy of the *BEA WebLogic JRockit Binary Re-Distribution Agreement*, contact BEA at the following email address: jrockit-partner@bea.com.

Support for WebLogic JRockit in WebLogic Platform Applications

When you distribute WebLogic Platform or any of its components, the WebLogic JRockit SDK is included by default. BEA automatically provides support for your customers who use WebLogic JRockit as part of a WebLogic Platform package.

If you have special support requirements for your business, send your requests to BEA at the following email address: jrockit-partner@bea.com.

For information about BEA support for WebLogic JRockit, see the following Web site:

http://commerce.bea.com/products/weblogicjrockit/support_services.jsp

Distributing the JRockit JVM as a Standalone Product

You may distribute WebLogic JRockit as a standalone product. If you do so, however, any of your customers who use the JRockit JVM separately from WebLogic Platform must buy a separate support contract from BEA. For more information, advise your customers to visit the following Web site:

http://commerce.bea.com/products/weblogicjrockit/support_services.jsp

Note: Any Java applications can be run on WebLogic JRockit. When Java applications are not running in a BEA WebLogic environment, however, the configuration settings for the JRockit JVM may need to be adjusted. For more information, see the following documents:

- *[Tuning WebLogic JRockit JVM](#)*
- “[Migrating to WebLogic JRockit](#)” in *Developing Applications*

Preparing Customer Documentation

This chapter includes advice and considerations on preparing customer documentation, including for two scenarios:

- Silent installation and configuration
- Interactive installation and configuration

General Advice on Preparing Customer Documentation

Regardless of the type of partner you are (see [“Types of ISVs” on page 1-1](#)), whenever you prepare any software product to be distributed to customers, you should provide at a minimum documentation that includes the following:

- Contents of the distribution

Document completely the contents of your distribution; for example, the parts of WebLogic Platform that are included—that is, WebLogic Server, WebLogic Integration, WebLogic JRockit, and so on. Identify the version numbers in several obvious locations (this will help with support).

- Support and license information

Describe completely the terms of the support and license provided with your product, and the warranty. Identify who your customers should contact, or what Web site they should use, if they need to use support services. Distinguish between the support you provide, versus that provided by BEA. (The support provided by BEA may depend on your support contract with BEA. See [Chapter 8, “About BEA Partner Support,”](#) for more information.)

- **Installation**

Document the installation application for your product, even if WebLogic Platform is installed and configured in silent-mode. Describe every step the customer needs to take; for example, how to download the distribution from the internet, how to identify the appropriate CD from a physical container, what item the customer chooses from the Start menu (if a Windows product).

- **Read-first documents**

Include, and identify plainly, any documentation that customers need to read before they install or use your product. Typically this information includes hardware and software prerequisites, memory and disk space requirements, and other resources that the customer will need to install or use your product. If applicable, this document should include any usage restrictions, or possibly any known problems that customers should know about.

- **User documentation**

This may be self-evident, but be sure to include comprehensive instructions and reference information for product use. For example, state clearly how to start and run your custom software.

Documentation for Silent-Mode Installation and Configuration

If you are using the silent-mode installation and configuration features of WebLogic Platform to install and configure your product, be sure to prepare documentation for the customer that describes the following:

- The choices that were made during the installation. For example, WebLogic Platform components installed, databases selected, configuration details for various system resources.
- The result of the installation. For example, the directories and files added to the customer's system, including the files associated with the customer's domain (that is, `user_projects`).
- Location of the licenses installed, as well as the terms of the licenses. Include instructions on how customers can extend the terms of the license.

Documentation for Interactive Installation and Configuration

If your product is designed to be installed interactively—that is, your customers must run the WebLogic Platform installation program, followed by the Configuration Wizard—be sure to prepare documentation for the customer that describes the following:

- A comprehensive description of the steps required to install WebLogic Platform.

Depending on your license agreement with BEA, you might have the ability to redistribute *Installing BEA WebLogic Platform*. Explain clearly any required installation choices customers must make during the installation.

- How to start and run the Configuration Wizard to use the template, or extension template, that you have created. This documentation should state the following:
 - Whether and how to create or extend an existing WebLogic configuration
 - How to locate the template or template extension
 - What the choices mean that you are instructing customers to make. If they have multiple choices for a given entry, explain the possible choices.

If you must include a cross-reference to a topic in the WebLogic Platform configuration documents for more information, be sure to make the cross-reference accurate and explicit. That is, make the cross-reference to a specific topic in a document, and not simply to the entire document itself. Customers do not want to wade through 200 pages of documentation for details about a specific configuration choice. It is assumed that you have also created a domain template, or template extension, for your application. (For information about templates and extension templates, see [Chapter 3, “Configuring and Installing WebLogic Platform Applications.”](#))

- The result of the installation. For example, the directories and files added to the customer’s system. For example, the directories and files added to the customer’s system, including the files associated with the customer’s domain (that is, `user_projects`).
- Location of the licenses installed, as well as the terms of the licenses. Include instructions on how customers can extend the terms of the license.

Preparing Customer Documentation

About BEA Partner Support

This chapter includes the following topics:

- [Partner Support Resources](#)
- [Upgrading WebLogic Platform](#)

Partner Support Resources

BEA provides a rich set of resources to support our partners enrolled in the Partner Program. These resources are organized in the following categories:

- [Partner Enablement Resources](#)
- [Software Development Resources](#)
- [Sales and Marketing Resources](#)
- [Support Resources](#)

For comprehensive information about all partner resources and benefits, enrolled partners can access the *BEA Partner Program Guide* at the following URL:

ftp://pdownload:BUY_ME@ftpna2.bea.com/pub/program/BEA_Partner_Program_Guide.pdf

Partner Enablement Resources

Partner enablement resources are designed to help new partners get started in their relationship with BEA, and to remain connected with the variety of tools, training, news, and partner events available. Resources include the following:

- Welcome kit
- Product sales training, including the partner Webinar series that encompass technical, business development, sales, and marketing topics.
- PartnerNet—BEA's portal to partner services. This password-protected, partners-only site, available at the following URL, contains the latest information about BEA's Partner Program:

<http://partnernet.bea.com>

- Partner Response Center—The comprehensive partner information and business support hotline, offering direct 5x9 telephone or email assistance with identifying resources to answer questions or providing information. Also acts as a conduit for technical support.
- Discounts on instructor-led, open enrollment technical education courses
- PartnerFlash, a monthly email newsletter that delivers partner news, such as how BEA is engaging its partners in successful businesses, and other updates
- Partner sales toolkit, and more

Software Development Resources

BEA's software development resources for partners include the software bundle and support described in [“Using the Partner Software Bundle” on page 2-21](#), and also the following:

- Validation program—A framework for verifying the technical readiness of partner solutions that complement BEA's platform. The BEA Validation Program verifies that partner applications meet a well-defined set of BEA approved integration requirements, and that the partner applications have been tested against these requirements.
- Workshop Controls and Extensibility Program—Designed to assist BEA partners build, test, distribute, and market their custom controls and IDE extensions. The Controls and Extensibility Program provides a community and set of resources for partners to distribute and market their custom controls and IDE extensions via [Premier Component Gallery](#), hosted at the BEA dev2dev Web site, and [Component Source](#).

(For more information about developing custom controls, see [Chapter 5, “Creating and Packaging WebLogic Workshop Java Controls.”](#))

- **Software Updates**—Unlimited access to the BEA Download Center, where you can download the latest BEA products for evaluation purposes. During your membership term, you can also download BEA product updates. You can access the Download Center directly from [PartnerNet](#) or from [eSupport](#).
- *BEA WebLogic Platform 8.1 SP3 Evaluation Guide*—An easy-to-use guide that provides a comprehensive, hands-on introduction to the BEA WebLogic Platform 8.1. More information about this document, and instructions for downloading it, are available at the following URL:

<http://dev2dev.bea.com/products/wlplatform81/technicalguides/evalguide81.jsp>

Sales and Marketing Resources

The Partner Program’s sales and marketing resources provide the tools and resources to help partners gain access to new markets and develop new business opportunities, and develop new business opportunities. Resources include:

- BEA Partner Solutions Catalog, an online partner locator tool where customers, prospects and BEA sales can learn about BEA partners and their solutions.
- BEA Partner Program logo, for branding usage with partner software products
- Marketing toolkit
- Relationship managers, lead registration, and sales incentive programs

Support Resources

The support resources available to partners include the following:

- Technical support hotline, which provides 5x9 developer-level phone support for help with technical problems encountered while developing with, implementing, or using BEA’s products.
- BEA eSupport, BEA’s online technical support Web site. Within the eSupport Web site, partners can submit online support requests, find the answers to the most commonly asked technical support questions, and can search the BEA support knowledge base using AskBEA, BEA’s natural language search engine. The eSupport Web site is located at <http://support.bea.com/>.

- Community Forums—Local user groups and online discussion forms, led by BEA developer relations engineers and BEA WebLogic developers world-wide, are available to partners to answer questions and provide advice and support. To join an online discussion forum, visit the following URL:

<http://newsgroups.bea.com>

- Access to e-docs online product documentation, BEA's comprehensive online product documentation site for all BEA products. Documentation includes release notes, supported platforms, and detailed installation, configuration, development, and administration information for all products. The e-docs Web site is located at the following URL:

<http://e-docs.bea.com>

- BEA dev2dev Online Portal—Developer-centric site that offers development tips and tools, sample code, newsgroups, and a host of Java community services to help with your development efforts and shorten the time it takes to learn and use the latest BEA technologies. The dev2dev online portal is available at the following URL:

<http://dev2dev.bea.com>

Upgrading WebLogic Platform

Periodically, BEA releases a service pack or a rolling patch for WebLogic Platform:

- A *service pack* is an update to an existing release that provides bug fixes and minor product enhancements. Typically, a service pack does not *replace* an installation of WebLogic Platform or its components; instead, it changes and expands the installation.
- A *rolling patch* is an update to an existing release that provides bug fixes between service packs.

Note: For the purposes of this document, we assume that your customers have valid BEA support contracts.

When you distribute software that is either bundled with WebLogic Platform or layered on top of it, you also need to provide a mechanism for keeping your customers informed about the various versions of WebLogic Platform, including service packs and rolling patches, for which your software is supported.

Before you establish customer support policies for your software, know the answers to the following questions about WebLogic Platform:

- Are all service packs and rolling patches for the current release of WebLogic Platform supported by your software product? Make sure you clearly identify the versions of WebLogic Platform for which you certify your product.
- Which service packs or rolling patches of WebLogic Platform do your customers need in order to use your product? How do you communicate this information to your customers? For example, do you send customers email or post the information to a Web site?
- When you distribute software that includes a specific version of WebLogic Platform or one of its components:
 - Do you include all the service packs and/or rolling patches required by your customers in the WebLogic Platform distribution that is part of your product?
 - How will you handle future service packs and rolling patches that need to be added to your customers' installations?

One solution is to simply redistribute your software, along with the latest version of WebLogic Platform. Another approach is to include, in your software package, instructions for upgrading the WebLogic Platform software. Find the solution that works best for you and your customers.

- Do you want to take responsibility for distributing updates to WebLogic Platform?

BEA provides update mechanisms such as the Download Installer Web site and the Smart Update tool. For information about these utilities, see [“Installing Service Packs and Rolling Patches”](#) in the *Installing BEA WebLogic Platform*.

Note: As of WebLogic Platform 8.1 Service Pack 2, the `license.bea` file that you generate and distribute, as described in [“Using an ISV Redistribution License” on page 4-1](#), will continue to be valid for any customers who later use the Upgrade Installer to upgrade their installations with service packs for WebLogic Platform. As a result, customers who install your distribution of the initial release of WebLogic Platform 8.1 can later use the Upgrade Installer to install Service Pack 2 without obtaining a new license.

If, however, you provide an upgrade to your existing customers by supplying a whole new version of WebLogic Platform that includes a service pack, then you must generate a new `license.bea` file and include it in your distribution.

About BEA Partner Support

Configuration Files Required for File-Based Portal Applications

This appendix lists the following four configuration files that need to be copied into the /WEB-INF directory of each portal application you create using WebLogic Platform ISV Edition:

- [Configuration File web.xml](#)
- [Configuration File weblogic.xml](#)
- [Configuration File netuix-config.xml](#)
- [Configuration File url-template-config.xml](#)

For more information about these files and how to download them using a browser, see [“Building Portal Applications for Use with ISV Edition” on page 2-9](#). These files are listed in this appendix for the convenience of those who are unable to download them using a browser.

Listing A-1 Configuration File web.xml

```
<?xml version="1.0" ?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

  <!-- NetUIx Servlet -->
  <servlet>
    <servlet-name>AppManagerServlet</servlet-name>
```

Configuration Files Required for File-Based Portal Applications

```
<servlet-class>com.bea.netuix.servlets.manager.PortalServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<!-- NetUIx Portlet Servlet -->
<servlet>
  <servlet-name>PortletServlet</servlet-name>

<servlet-class>com.bea.netuix.servlets.manager.PortletServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<!-- NetUIx Servlet Mapping -->
<servlet-mapping>
  <servlet-name>AppManagerServlet</servlet-name>
  <url-pattern>/appmanager/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>AppManagerServlet</servlet-name>
  <url-pattern>*.portion</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>AppManagerServlet</servlet-name>
  <url-pattern>*.portal</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>PortletServlet</servlet-name>
  <url-pattern>/portletmanager/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>PortletServlet</servlet-name>
  <url-pattern>*.portlet</url-pattern>
</servlet-mapping>

<!-- Welcome files -->
<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>

<!-- Define the netuix tag library tld -->
<taglib>

<taglib-uri>http://www.bea.com/servers/netuix/xsd/controls/netuix/1.0.0</taglib-
uri>
  <taglib-location>/WEB-INF/lib/netuix_taglib.jar</taglib-location>
</taglib>
```

```

<!-- Define the html tag library tld -->
<taglib>
  <taglib-uri>http://www.w3.org/1999/xhtml-netuix-modified/1.0.0</taglib-uri>
  <taglib-location>/WEB-INF/lib/html_taglib.jar</taglib-location>
</taglib>

<!-- Render Control Tag Library Descriptor -->
<taglib>
  <taglib-uri>render.tld</taglib-uri>
  <taglib-location>/WEB-INF/lib/render_taglib.jar</taglib-location>
</taglib>

<!-- Localization Tag Library Descriptor -->
<taglib>
  <taglib-uri>l10n.tld</taglib-uri>
  <taglib-location>/WEB-INF/lib/l10n_taglib.jar</taglib-location>
</taglib>

<login-config>
  <auth-method>BASIC</auth-method>
</login-config>

</web-app>

```

Listing A-2 Configuration File weblogic.xml

```

<?xml version="1.0" ?>
<!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//DTD Web Application
8.1//EN"
"http://www.bea.com/servers/wls810/dtd/weblogic810-web-jar.dtd">

<weblogic-web-app>

  <jsp-descriptor>
    <jsp-param>
      <param-name>jspServlet</param-name>
      <param-value>weblogic.servlet.WlwJSPServlet</param-value>
    </jsp-param>
    <jsp-param>
      <param-name>debug</param-name>
      <param-value>true</param-value>
    </jsp-param>
    <jsp-param>
      <param-name>keepgenerated</param-name>
      <param-value>true</param-value>
    </jsp-param>
    <jsp-param>

```

Configuration Files Required for File-Based Portal Applications

```
        <param-name>pageCheckSeconds</param-name>
        <param-value>1</param-value>
    </jsp-param>
    <jsp-param>
        <param-name>encoding</param-name>
        <param-value>UTF8</param-value>
    </jsp-param>
</jsp-descriptor>

</weblogic-web-app>
```

Listing A-3 Configuration File netuix-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<weblogic-portal-app
    xmlns="http://www.bea.com/servers/portal/weblogic-portal/8.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.bea.com/servers/portal/weblogic-portal/8.0
netuix-config.xsd">

    <!-- Turn customization on or off -->
    <customization>
        <enable>false</enable>
        <propagate-preferences-on-deploy propagate-to-instances="false"/>
    </customization>

    <!-- Turn off netui dependancy -->
    <pageflow>
        <enable>false</enable>
    </pageflow>

    <!-- Turn off entitlement dependency and runtime check -->
    <entitlements>
        <enable>false</enable>
    </entitlements>

    <!-- Enable or disable localization -->
    <localization>
        <enable>true</enable>
    </localization>

    <!-- Container supported widow states -->
    <window-state name="minimized">
        <activate-image>titlebar-button-minimize.gif</activate-image>
        <deactivate-image>titlebar-button-unminimize.gif</deactivate-image>
        <alt-text>
            <locale language="en">
```

```

                <activate>Minimize</activate>
                <deactivate>Restore</deactivate>
            </locale>
        </alt-text>
    </window-state>
    <window-state name="maximized">
        <activate-image>titlebar-button-maximize.gif</activate-image>
        <deactivate-image>titlebar-button-unmaximize.gif</deactivate-image>
        <alt-text>
            <locale language="en">
                <activate>Maximize</activate>
                <deactivate>Restore</deactivate>
            </locale>
        </alt-text>
    </window-state>
    <window-state name="delete">
        <activate-image>titlebar-button-delete.gif</activate-image>
        <alt-text>
            <locale language="en">
                <activate>Delete</activate>
            </locale>
        </alt-text>
    </window-state>
    <window-state name="float">
        <activate-image>titlebar-button-float.gif</activate-image>
        <alt-text>
            <locale language="en">
                <activate>Float</activate>
            </locale>
        </alt-text>
    </window-state>

    <!-- Container supported window modes -->
    <window-mode name="help">
        <activate-image>titlebar-button-help.gif</activate-image>
        <deactivate-image>titlebar-button-help-exit.gif</deactivate-image>
        <alt-text>
            <locale language="en">
                <activate>Help</activate>
                <deactivate>Leave Help</deactivate>
            </locale>
        </alt-text>
    </window-mode>
    <window-mode name="edit">
        <activate-image>titlebar-button-edit.gif</activate-image>
        <deactivate-image>titlebar-button-edit-exit.gif</deactivate-image>
        <alt-text>
            <locale language="en">
                <activate>Edit</activate>

```

Configuration Files Required for File-Based Portal Applications

```
        <deactivate>Leave Edit</deactivate>
      </locale>
    </alt-text>
  </window-mode>

</weblogic-portal-app>
```

Listing A-4 Configuration File url-template-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<url-template-config
  xmlns="http://www.bea.com/servers/weblogic/url-template-config/8.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xsi:schemaLocation="http://www.bea.com/servers/weblogic/url-template-config/8.0 url-template-config.xsd">

  <!-- URL templates -->
  <url-template name="default">
    {url:scheme}://{url:domain}:{url:port}/{url:path}?{url:queryString}
  </url-template>
  <url-template name="proxyurl">

    {url:scheme}://{url:domain}:{url:port}/{url:prefix}/{url:path}?{url:queryStrin
g}
  </url-template>
  <url-template name="finurl">

    https://{fin.domain.com:7004}/{url:prefix}/{url:path}?{url:queryString}&dept
=finance
  </url-template>
  <url-template name="default-complete">

    {url:scheme}://{url:domain}:{url:port}/{url:prefix}/{url:path}?{url:queryStrin
g}
  </url-template>
  <url-template name="jpf-default">

    http://{url:domain}:{url:port}/{url:path}?{url:queryString}{url:currentPage}
  </url-template>
  <url-template name="jpf-action">
    http://{url:domain}:{url:port}/{url:path}?{url:queryString}
  </url-template>
  <url-template name="jpf-secure-action">
    https://{url:domain}:{url:securePort}/{url:path}?{url:queryString}
  </url-template>
  <url-template name="jpf-resource">
```



```

        http://{url:domain}:{url:port}/{url:path}?{url:queryString}
    </url-template>
    <url-template name="jpf-secure-resource">
        https://{url:domain}:{url:securePort}/{url:path}?{url:queryString}
    </url-template>
    <url-template name="extraVarTemplate">

{url:scheme}://{url:domain}:{url:port}/{url:path}?{url:queryString}&{foo:bar}
    </url-template>
    <url-template name="cr124687Template">
        foo://{fooDomain:9999}/{url:path}?{url:queryString}&{foo:bar}
    </url-template>
    <url-template name="cr126105Template">
        foo://{fooDomain:9999}/{url:path}?{url:queryString}{url:currentPage}
    </url-template>
    <jpf-url-templates>
        <url-template-ref type="action" name="jpf-default"/>
        <url-template-ref type="secure-action" name="jpf-default"/>
        <url-template-ref type="resource" name="jpf-default"/>
        <url-template-ref type="secure-resource" name="jpf-default"/>
    </jpf-url-templates>

    <java-portlet-url-templates>
        <url-template-ref type="action" name="portlet-default"/>
        <url-template-ref type="secure-action" name="portlet-default"/>
        <url-template-ref type="resource" name="portlet-default"/>
        <url-template-ref type="secure-resource" name="portlet-default"/>
    </java-portlet-url-templates>

</url-template-config>

```

Configuration Files Required for File-Based Portal Applications

Index

A

- adapters
 - developing 2-2
- Administration Console
 - WebLogic Server 3-4
- AskBEA 8-1

B

- BEA software
 - downloading 2-22
- business processes
 - developing 2-2

C

- CDK
 - about 1-8
- Channel License Agreement
 - see CLA
- CLA
 - and ISV Edition 2-9
- components excluded from 2-7
- configuration
 - silent-mode 3-7
- configuration files
 - required for ISV Edition portals 2-15
- Configuration Wizard
 - about 3-2
- Control Development Kit
 - See CDK
- controls
 - Java, developing 1-8

D

- data transformations
 - developing 2-2
- dev2dev 8-1
- domains
 - and WebLogic Portal Extension 2-12
 - for portals with ISV Edition 2-12
- downloading
 - BEA software 2-22

E

- eSupport 8-1
- extension template
 - creating 3-3

F

- file-based portals 2-10
 - administering users of 2-18

I

- Independent Software Vendor
 - See ISV
- installation
 - customer documentation for 7-2
 - silent-mode 3-7
- Inter-Portel Communication (IPC) package
 - support
 - in ISV Edition 1-4
- ISV Edition 2-7
 - and CLA 2-9
 - and portals 2-11

- and WebLogic Server Process Edition 1-4, 2-20
- building portals with 2-9
- components included in 2-7
- installing 2-8
- licenses 2-9
- portal development environment 2-12
- portal framework support features 1-4
- support for Inter-Portal Communication (IPC)
 - package 1-4
- using 2-5

ISV redistribution license

- about 1-6
- creating 4-3
- terms 4-2
- updating 4-5
- using 4-1
- validating 4-5

ISV track

- Partner Program 1-5

isv.jar 4-1

ISVs

- bundled 1-2
- complementary 1-2
- component 1-2
- JRockit 1-2
- layered 1-2
- types 1-1

J

J2EE programming 2-4

JAR files

- for portal applications 2-12

Java controls

- and CDK 1-8

L

license.bea 4-4

license_isv.bea 4-4

licenses

- ISV Edition 2-9

N

netuix-config.xml 2-15

P

Partner Program 1-5

Partner Response Center 8-1

partner software bundle

- contents 2-21

- installing 2-22

partner support 8-1

partner support services 1-8

PartnerNet 8-1

portal development environment

- creating for ISV Edition 2-12

portal framework

- features in ISV Edition 1-4

Portal User Interface Framework 2-11

portals

- and ISV Edition 2-9

- building with ISV Edition 2-18

- deploying with ISV Edition 2-19

- developing 2-2

- features supported in ISV Edition 2-11

- file-based 2-10

- JAR files needed for 2-12

- streaming 2-10

- User Interface Framework 2-11

R

rolling patches

- upgrading 8-4

S

scripts

- startup 3-6

service packs

- upgrading 8-4

silent-mode

- installation and configuration 3-7
- startup scripts 3-6
- streaming portals 2-10
- support services 1-8

T

- template
 - creating 3-3
- Template Builder
 - about 3-3
- types of ISVs 1-1

U

- url-template-config.xml 2-15
- users
 - administering for file-based portals 2-18

W

- Web applications
 - developing
 - Enterprise JavaBeans
 - developing 2-2
- Web services
 - developing 2-2
- Web Services for Remote Portlets
 - see WSRP
- web.xml 2-15
- WebLogic Administration Portal 3-5
- WebLogic Integration
 - Administration Console 3-5
- WebLogic Platform
 - upgrading 8-4
- WebLogic Platform ISV Edition
 - see ISV Edition
- WebLogic Server
 - configuring 3-4
- WebLogic Server Administration Console 3-4
- WebLogic Server applications
 - developing 2-4

- WebLogic Server Process Edition
 - support in ISV Edition 1-4
- WebLogic Server Process Edition applications
 - building with ISV Edition 2-20
- weblogic.jar 4-4
- weblogic.xml 2-15
- WSRP
 - support in ISV Edition 1-4, 2-11

