BEAProducts

ISV Partners' Guide

BEA WebLogic Server® Version 10.0
Revised: March 30,2007

Contents

3. Configuring and Installing WebLogic Applications

4. Distributing WebLogic Server

5. Using the BEA Registry API

6. Supporting and Distributing BEA JRockit

7. Preparing Customer Documentation

8. About BEA Partner Support

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 BEA WebLogic Server or their own custom applications that interoperate with WebLogic Server

- 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 or host WebLogic Server 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 . . .
Embedded	WebLogic Server, or one or more components of WebLogic Server, packaged with their application. (The WebLogic Server software being distributed, in combination with the ISV application, is referred to in this document as the ISV's *value-added solution*.)
Layered	Prepackaged WebLogic applications that run on WebLogic Server, but that do not include WebLogic Server. It is assumed that customers who buy layered products have valid WebLogic Server installations.
JRockit	BEA JRockit JDK or Mission Control tools, either as stand-alone products or with the ISV's value-added solution.
Complementary	Products that interoperate with WebLogic Server, but that do not necessarily run as WebLogic 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, services, and tools for its partners:

- BEA Partner Program
- Software Bundle
- ISV Redistribution License
- WebLogic Server Installation, Maintenance, and Configuration Tools
- Registry API
- Administration Console Extensions
- Partner Support

This section describes each of these resources in detail.

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

- Comprehensive technical services and support to software developers

- Validation program—a framework for verifying the technical readiness of partner solutions that complement BEA's platform

- Certification program—qualifies applicants for development, implementation, and administration competencies with WebLogic Server

This document, *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 Server. 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 Server. For a complete list of supported configurations, see Supported Configurations.

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 test server licenses for BEA Tuxedo with Jolt and WebLogic Server. These licenses give partners the ability to load the WebLogic Server software onto test servers for several purposes: performing 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_Guid`
`e.pdf`

ISV Redistribution License

The WebLogic Server 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 Server, BEA provides the ability for ISVs to generate a license for their customers so that they can run the WebLogic Server software that is embedded with the ISV's application. 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 Server software distributed by the ISV— each license is unique, customized to control the WebLogic Server components that customers of a particular ISV partner can use at run time.

ISV partners who have a current agreement to redistribute WebLogic Server receive software that allows them to generate both an ISV redistribution license and a version of the WebLogic Server software that is associated with that license. The ISV redistribution license allows ISV customers to run the specific WebLogic Server software associated with that license.

To find out how to generate an ISV redistribution license and package it with the WebLogic Server software developed by your company, see Chapter 4, "Distributing WebLogic Server."

WebLogic Server Installation, Maintenance, and Configuration Tools

To help you create ISV-ready WebLogic Server configurations that your customers can install and configure easily, BEA provides the following tools:

- Installation Program
- Smart Update
- Configuration Wizard
- Domain Template Builder

- pack and unpack Commands
- WebLogic Scripting Tool
- Silent Installation, Maintenance, and Configuration

Installation Program

BEA's installation program makes it easy to install either the entire WebLogic Server product or individual components of it—whichever you prefer. When run in noninteractive or *silent* mode, the installation program makes it even easier for your customers to install WebLogic Server.

Smart Update

Periodically, BEA releases a maintenance pack (MP) for WebLogic Server:

A maintenance pack is an update to an existing release that provides bug fixes but no product enhancements. Typically, a maintenance pack does not replace an installation of WebLogic Server; instead, it changes and expands the installation.

Smart Update is a standalone Java application that you can run independently of WebLogic Server to updates to BEA software. In WebLogic Server 9.1, the capabilities of Smart Update have been greatly expanded to include the ability to log in to the BEA Customer Support Web site, eSupport, to download private as well as publicly available patches. You can then use Smart Update to apply and manage those patches for installed BEA products via either its graphical or command-line interface.

ISVs who redistribute WebLogic Server can leverage the capabilities of Smart Update to include patches that are provided by BEA Customer Support in the ISV redistribution. The installation of these patches can be automated, via script, as part of the installation of the ISV's value-added solution. Or patches can be redistributed separately, as appropriate. By using the patch management capabilities of Smart Update, you can have a reliable and flexible means for ensuring that your value-added solution runs on the proper maintenance level of the underlying BEA software.

Configuration Wizard

The Configuration Wizard is a stand-alone Java application you can use to create the domain in which your ISV application is developed or run. A domain is the basic administrative unit for WebLogic Server. It consists of an interrelated set of WebLogic Server resources that are managed as a unit and includes one or more WebLogic Server instances. It may also include WebLogic Server clusters.

Depending on how you deliver your value-added solution to your customers, you may want them to run the Configuration Wizard after they have installed WebLogic Server to set up their environments to run your software.

Domain Template Builder

A domain template defines the full set of resources within a domain, including infrastructure components, applications, services, security options, and general environment and operating system parameters. After you have created a domain for your ISV application, and added resources and various customizations to it, you can use the Domain Template Builder to create a domain template or extension template of that domain. When users subsequently create or extend a domain based on that template, the domain is automatically configured with all the resources required to run your value-added solution.

pack and unpack Commands

Two new commands are available to simplify the creation of domain templates and domains, including Managed Server domains:

- The `pack` command creates a template archive (`.jar`) file that contains a snapshot of either an entire domain or the part of a domain needed to create a Managed Server directory hierarchy on a remote machine in the domain.

- The `unpack` command creates a domain directory or Managed Server domain directory using a template created with the `pack` command. It also creates necessary start scripts, and certain security and configuration files for the domain.

When the `unpack` command is run from a script in your customer's environment, you can provide an automatic and seamless process for configuring the environment required for your value-added solution.

WebLogic Scripting Tool

The WebLogic Scripting Tool (WLST) is a new command-line interface that you use to configure WebLogic Server instances and domains, and manage and persist WebLogic Server configuration changes.

WLST interprets commands either interactively, as they are supplied, one at a time, from a command prompt, or in batches, when they are supplied in a file (script) or embedded in your Java code. You can use the scripting tool **online** (connected to a running server) or **offline** (not connected to a running server).

Offline, WLST provides access to persisted configuration information. You can create a new domain, or update an existing one, without connecting to a running WebLogic Server.

Silent Installation, Maintenance, and Configuration

By creating a simple shell script or command file that sequentially runs silent-mode installation, a Smart Update script, and a WLST script to configure one or more domains for running your value-added solution, you can simplify even further your customers' work of installing and configuring WebLogic Server at the proper maintenance level. This approach gives your customers a streamlined process, right out of the box, so that they can start running your applications more quickly and easily.

Registry API

The BEA Registry API is a Java package included with WebLogic Server that provides a means to access information about a BEA software installation, such as which products and which versions of those products are installed. Use of the BEA Registry API gives ISV applications a more manageable, predictable means of interoperating with BEA software as usage environments change.

Administration Console Extensions

The WebLogic Server Administration Console is a Java EE Web application that uses the WebLogic Portal framework, Apache Beehive, Apache Struts, Java Server Pages (JSP), and other standard technologies to render its user interface (UI) and content. This architecture makes the console highly extensible, enabling you to add or replace console content and to change the logos, styles, and colors without modifying the files that are installed with WebLogic Server. For example, you can add a portlet that provides custom monitoring and management facilities for your applications.

The extensibility features of the Administration Console enable ISVs to do the following:

- Rebrand, that is, the ability to change the look and feel attributes of the Administration Console, such as colors, logos, and fonts

- Add a custom portlet that exposes monitoring information or configuration data for an application, a custom security provider, or other ISV-built resource

- Add and remove fields from existing console pages

- Add and remove tabs to the existing console

- Build a highly simplified console containing pages for only the relevant features of the ISV's application

- Divert help requests to an alternate URL and modify the set of tasks displayed in the *How do I* portlet.

Customizations to the Administration Console are provided via a console extension, which is a JAR file you create that contains the resources for a subsection of a WebLogic Portal Web application. Once deployed, the extension becomes part of the Administration Console: it is secured by the WebLogic Server security realm, it is navigable to other sections of the Administration Console, and if the extension modifies WebLogic Server resources, it participates in the change control process.

For more information, see *Extending the Administration Console*.

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:

- Embedded

- Layered

- Stand-alone BEA 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 . . .
Embedded	Redistributing part or all of BEA WebLogic Server	1. Enroll in the Partner Program. See "BEA Partner Program" on page 1-3. 2. Back up your license.bea file. See "Using the Partner Software Bundle" on page 2-5. 3. Install the partner software bundle. See "Using the Partner Software Bundle" on page 2-5. 4. Create the ISV redistribution license and link it to the WebLogic Server software you will redistribute. 5. Create a package to distribute to your customers, ensuring that the ISV redistribution license and the WebLogic Server software to which it is linked are installed correctly. See Chapter 4, "Distributing WebLogic Server." **Note:** For instructions for distributing custom software along with any or all of WebLogic Server, see steps 3 through 6 in the procedure for ISVs of layered products, later in this table.

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 Server	1. Enroll in the Partner Program. See "BEA Partner Program" on page 1-3. 2. Install the partner software bundle. See "Using the Partner Software Bundle" on page 2-5. 3. Build, test, and deploy the application. See "Building WebLogic Server Applications" on page 2-1. 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 Server, see Chapter 3, "Configuring and Installing WebLogic Applications." For information about the Configuration Wizard, see "Configuration Wizard" on page 3-2. For information about the Template Builder, see "Domain 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 "Domain 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."
JRockit	Redistributing the BEA JRockit JVM	1. Enroll in the Partner Program. See "BEA Partner Program" on page 1-3. 2. Familiarize yourself with the BEA JRockit J2SE 5.0 JDK, and with BEA's policies for providing and supporting it. 3. Learn about how BEA distributes JRockit and provides support, for this product, to partners and their customers. See Chapter 6, "Supporting and Distributing BEA JRockit."

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 . . .
Complementary	Creating non-BEA software for use in a WebLogic Server environment	1. Enroll in the Partner Program. See "BEA Partner Program" on page 1-3. 2. If you are creating management or monitoring tools, see "Creating JMX-Compatible Management Systems" on page 2-4.

Developing Applications

This chapter provides links to documentation that explains how to build, test, and deploy applications for WebLogic Server. This chapter also describes the software bundle that is provided to partners who enroll in the Partner Program. The software bundle contains WebLogic Server, and includes developer support and special licensing geared specifically to ISVs.

The following topics are included:

- Building WebLogic Server Applications
- Creating Custom Diagnostic Tools
- Creating Custom Security Providers
- Creating JMX-Compatible Management Systems
- Using the Partner Software Bundle
- Next Steps

Building WebLogic Server Applications

WebLogic Server provides the infrastructure you need for creating, testing, and deploying e-commerce applications using the tools provided by Java Platform, Enterprise Edition (Java EE) Version 5 from Sun Microsystems, including:

- Web application modules—HTML pages, servlets, JavaServer Pages, and related files

- Enterprise Java Beans (EJB) modules—entity beans, session beans, and message-driven beans

- Connector modules (resource adapters)—stand-alone code components that contain the Java modules and, optionally, native modules required to interact with an Enterprise Information System (EIS)

- Enterprise applications—Web application modules, EJB modules, and resource adapters packaged as applications

- WebLogic Web services—functions packaged into a single entity that is available to other systems on a network and can be shared by and used as a component of distributed Web-based applications

Table 2-1 provides links to documentation for information about how to create each type of application.

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

For information about developing . . .	See . . .
Web applications	*Developing Web Applications, Servlets, and JSPs for WebLogic Server*
Enterprise JavaBeans	*Programming WebLogic Enterprise JavaBeans*
Connector modules	*Programming WebLogic Resource Adapters*
WebLogic Web services	*Programming Web Services for WebLogic Servers*—Explains how to build Web services using the WebLogic Server programming environment
Additional WebLogic Server applications	• *Developing Applications 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 XML* • *Monitoring and Managing with the J2EE Management APIs*

Creating Custom Diagnostic Tools

The WebLogic Diagnostic Framework (WLDF) is a monitoring and diagnostic framework that defines and implements a set of services that run within the WebLogic Server process and participate in the standard server life cycle. Using WLDF, you can create, collect, analyze, archive, and access diagnostic data generated by a running server and the applications deployed within its containers. This data provides insight into the run-time performance of servers and applications and enables you to isolate and diagnose faults when they occur.

WLDF includes several components for collecting and analyzing data, including the following:

- Instrumentation—adds code to instances of WebLogic Server and the applications running on them to execute diagnostics

- Server Image Capture—creates a diagnostic snapshot from the server that can be used for post-failure analysis

- Archiver—captures and persists all data events, log records, and metrics from server instances and applications

- Harvester—captures metrics from run-time MBeans, including WebLogic Server MBeans and custom MBeans

- Watches and Notifications—provides the means for monitoring server and application states and sending notifications based on criteria set in the watches

WLDF provides a set of standardized application programming interfaces (APIs) that enable dynamic access and control of diagnostic data, as well as improved monitoring that provides visibility into the server. ISVs can use these APIs to develop custom monitoring and diagnostic tools for integration with WLDF. For more information, see *Configuring and Using the WebLogic Diagnostic Framework*.

Creating Custom Security Providers

WebLogic Security Service includes modules that provide security services to applications for the protection of WebLogic resources. WebLogic Server provides a number of these modules, called security providers, out-of-the-box. Security providers "plug in to" a WebLogic Server security realm to provide security services, such as authentication, authorization, identity assertion, adjudication, role-mapping, credential-mapping, auditing, and so on.

Although the WebLogic Security Service provides end-to-end security out-of-the-box for a WebLogic Server deployment environment, the use of custom security providers enables users to

accommodate specialized application security requirements. The WebLogic Security Service provides several Security Service Provider Interfaces (SSPIs), which ISVs can use to develop custom security providers for the WebLogic Server environment. SSPIs are available for Adjudication, Auditing, Authentication, Authorization, Credential Mapping, Identity Assertion, Role Mapping, and Certificate Lookup and Validation.

To develop a custom security provider, an ISV must:

- Create run-time classes for the security provider by implementing the appropriate security service provider interfaces (SSPIs) from the `weblogic.security.spi` package.

- Create an MBean type, which is used for configuring and managing the security provider.

- Create a console extension to the WebLogic Server Administration Console to support management and configuration capabilities for the custom provider.

For detailed information about creating custom security providers, see *Developing Security Providers for WebLogic Server*.

Creating JMX-Compatible Management Systems

Java Management Extensions (JMX) is a specification for monitoring and managing Java applications. It enables a generic management system to monitor an application, raise notifications when the application needs attention, and change the state of the application to remedy problems. Like SNMP and other management standards, JMX is a public specification and many vendors of commonly used monitoring products support it.

To enable ISVs to integrate their management systems with the WebLogic Server management system, WebLogic Server provides a set of standards-based interfaces that are fully compliant with the Java Management Extensions (JMX) specification. WebLogic Server uses the JMX 1.2 implementation that is included in JDK 1.5.

ISVs can use these interfaces to build management applications that monitor WebLogic Server managed beans (MBeans) or custom MBeans, change the configuration of a WebLogic Server domain, and monitor the distribution of these changes to all server instances in the domain. ISVs can also use these interfaces to manage end-user applications that contain JMX instrumentation. Because BEA's implementation of JMX is fully compliant with JMX 1.2, JMX clients can perform all WebLogic Server management functions without using BEA's proprietary classes.

For more information about building JMX-compatible management systems, see *Developing Custom Management Utilities with JMX*.

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 unlimited license rights for developing, deploying, and testing applications for WebLogic Server, AquaLogic Service Bus, and BEA Tuxedo 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 are especially useful for creating demonstrations, prototypes, and proofs-of-concept appropriate for your business.

Software Bundle Contents

The software bundle includes the following:

- Unlimited WebLogic Server test server licenses

    Gives you the ability to load the WebLogic Server software onto an unlimited number of test servers. Use of these licenses is limited specifically to testing purposes.

- Unlimited AquaLogic and BEA Tuxedo developer and test licenses

    All partners receive unlimited developer and test licenses for AquaLogic family products and BEA Tuxedo with Jolt. Use of these licenses is limited specifically to development, testing, and proof-of-concept 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@ftpna2.bea.com/pub/program/BEA_Partner_Program_Guid`
`e.pdf`

Installing the Software Bundle

When you receive your product CDs, install the software from them. For installation instructions, see the *WebLogic Server Installation Guide*.

Caution: If the BEA 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

Your Installation" in "Preparing for Your Installation" in the *WebLogic Server Installation Guide*.

Downloading BEA Software

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

http://commerce.bea.com

If you have an active eSupport account, use your eSupport login and 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 install WebLogic Server and finish building the applications that you plan to distribute, complete the following tasks:

1. Configure a domain for developing and running your applications. See Chapter 3, "Configuring and Installing WebLogic Applications."

2. Create a template of the domain created in the previous step that reproduces the domain required for running your applications. See Chapter 3, "Configuring and Installing WebLogic Applications."

3. Distribute your files. See Chapter 4, "Distributing WebLogic Server."

Configuring and Installing WebLogic Applications

WebLogic Server 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 Server with your value-added solution, or distributing an application that runs on WebLogic Server, 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 Server and configuring your customers' environments

The following topics are included:

- Configuring WebLogic Server for Your Application

- WebLogic Server Administration Tools

- Silent-Mode Installation, Maintenance, and Configuration

Configuring WebLogic Server 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 Server

- Modifying an existing WebLogic Server 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" on page 1-5, 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 Server environment: it comprises the WebLogic Server components, user-written programs, machines, and resources, such as databases, needed for a particular application.

The Configuration Wizard is designed to simplify and automate the following processes:

- Initial domain configuration

- The addition of product component functionality and well-defined applications and services to existing domains

The Configuration Wizard operates off-line; that is, when WebLogic Server is not running. It is not designed for use online when a WebLogic Server instance is running.

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 JDBC Data Sources and JMS File Store settings.

  For layered ISVs, extension templates are useful because they provide a means for you to add your applications to existing customer domains.

For complete details about using the Configuration Wizard, see *Creating WebLogic Domains Using the Configuration Wizard*.

For complete details about WebLogic domains, see the following:

- "Introduction" in *Creating WebLogic Domains Using the Configuration Wizard*.

- "Understanding WebLogic Server Domains" in *Understanding Domain Configuration*.

Domain Template Builder

The Domain Template Builder is a stand-alone Java application you use to create custom configuration and extension templates that can be used later for creating and updating domains. Using an existing domain or template, the Domain 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 Server environments.

The Domain 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 more information about the Domain Template Builder, see *Creating Templates Using the Domain Template Builder*.

unpack Command

The `unpack` command provides a highly streamlined way to create a full domain or a Managed Server domain directory. You may use `unpack` only with a template compatible with your current installation. The template can be any of the following:

- A domain template provided by BEA and packaged with your current installation

- A domain template created using the Domain Template Builder

- A domain template created using the `pack` command

- A Managed Server template created by using the `pack` command. A Managed Server template, by default, contains only those files necessary for creating a Managed Server domain directory.

When you use the `unpack` command with a domain template, `unpack` creates a domain containing all the application and resource files defined in the template. It also creates necessary start scripts, and certain security and configuration files.

For more information, see *Creating Templates and Domains Using the pack and unpack Commands*.

WebLogic Scripting Tool

The WebLogic Scripting Tool (WLST) is a command-line interface that you use to create and configure WebLogic Server instances and domains, and manage and persist WebLogic Server configuration changes.

WLST enables you to do the following:

- Create domains and domain templates

- Retrieve domain configuration and run-time information

- Edit the domain configuration and persist the changes in the `config.xml` file

- Navigate and edit custom, user-created MBeans and non-WebLogic Server MBeans, such as WebLogic Integration Server and WebLogic Portal Server MBeans

- Automate configuration tasks and application deployment

- Clone WebLogic Server domains

- Access Node Manager and start, stop, and suspend server instances remotely or locally, without requiring the presence of a running Administration Server

Based on the Java scripting interpreter, Jython, WLST interprets commands either interactively, as they are supplied, one at a time, from a command prompt, or in batches, when they are supplied in a file (script) or embedded in your Java code. You can use the scripting tool **online** (connected to a running server) or **offline** (not connected to a running server).

- Online, WLST provides simplified access to MBeans. You can perform administrative tasks and initiate WebLogic Server configuration changes while connected to a running server.

- Offline, WLST provides access only to persisted configuration information. You can create a new domain or update an existing domain without connecting to a running WebLogic Server. This functionality resembles that of the Configuration Wizard.

For more information about WLST, see *WebLogic Scripting Tool*.

WebLogic Server Administration Tools

WebLogic Server 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 Server or any of its components. This section briefly describes some of these tools.

WebLogic Server Administration Console

The BEA WebLogic Server Administration Console is a Web browser-based, graphical user interface you use to manage a WebLogic Server domain. A WebLogic Server domain is a logically related group of WebLogic Server resources that you manage as a unit. A domain includes one or more instances of WebLogic Server and may also include WebLogic Server clusters. Clusters are groups of WebLogic Server instances that work together to provide scalability and high-availability for applications. You deploy and manage your applications as part of a domain.

One instance of WebLogic Server in each domain is configured as an Administration Server. The Administration Server provides a central point for managing a WebLogic Server domain. All other WebLogic Server instances in a domain are called Managed Servers. In a domain with only a single WebLogic Server instance, that server functions as both Administration Server and Managed Server. The Administration Server hosts the Administration Console, which is a portal application accessible from any supported Web browser with network access to the Administration Server.

You can use the Administration Console to:

- Configure, start, and stop WebLogic Server instances

- Configure WebLogic Server 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

Beginning with version 9.0, the architecture of the WebLogic Server Administration Console provides many new features for customization and extensibility. For a summary of the extensibility features, see "Administration Console Extensions" on page 1-7.

For complete details about the WebLogic Server Administration Console, see *Administration Console Online Help*.

Startup Scripts

WebLogic Server 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 Server 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 any of the following statements apply to the domain in the target environment:

- The domain runs on a single WebLogic Server host.

- The domain includes multiple server instances and you want them to run on multiple WebLogic Server 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 "Starting and Stopping Servers" in *Managing Server Startup and Shutdown*.

Silent-Mode Installation, Maintenance, 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 Server (or selected components), optionally apply patches to WebLogic Server, and configure their environments. This section introduces WebLogic Server utilities that can help you provide your customers with a simplified, easy-to-use method of configuring their environments:

- Silent-Mode Installation
- Applying Maintenance Patches from a Script
- Configuring a Domain from a Script
- Running Silent-Mode Installation, Maintenance, and Configuration from Scripts

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, including sample scripts, see "Running the Installation Program in Silent-Mode" in *WebLogic Server Installation Guide*.

Applying Maintenance Patches from a Script

The WebLogic Server software embedded with your value-added solution may require one or more private or publicly available patches from BEA Customer Support that you include with your product distribution. To ensure that your end-user customers run the embedded WebLogic Server software at a specific maintenance level, your overall installation procedure can include a script that invokes the Smart Update command-line interface to apply those maintenance patches, which are included with the distribution media of your value-added solution.

For more information about the patch management capabilities of Smart Update, see *Installing Maintenance Updates and Service Packs*.

For information about including patches with your product distribution and including a means to apply them to the embedded WebLogic Server software, see "Distributing Maintenance Patches" on page 4-6.

Configuring a Domain from a Script

Two primary mechanisms are available for automating the configuration of a domain or domain extension from a script:

- `unpack` command
- WebLogic Scripting Tool (WLST) offline

Domain configuration from a script 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 create a mechanism that allows you to duplicate a domain on customers' machines.

As mentioned in "unpack Command" on page 3-3, you can use the `unpack` command to create a full domain or a Managed Server domain directory quickly from a domain or Managed Server template, respectively.

Using a WLST offline script, you can create a new domain or update an existing domain without connecting to a running instance of WebLogic Server. Your WLST script can configure a new domain by using a domain template JAR created by the Domain Template Builder or `pack` command.

Running Silent-Mode Installation, Maintenance, and Configuration from Scripts

When you combine silent-mode installation, command-line maintenance, and configuration from scripts, you have a powerful method that lets you install your WebLogic product, apply all required maintenance patches, and configure a domain for it in a way that is transparent to customers. When your customers run a silent-mode installation script, followed by a scripts that apply patches and configure a domain, they experience the work of setting up an environment as a seamless process.

Note that a silent-mode installation script cannot invoke Smart Update, WLST, or the `unpack` command. However, you can create a simple shell script or command file that sequentially runs

the installation program in silent mode, Smart Update, and WLST. For details about silent installation, applying patches, and configuring domains, see the following:

- "Running the Installation Program in Silent Mode" in the *WebLogic Server Installation Guide*.

- "Using the Command-Line Interface" in *Installing Maintenance Updates and Service Packs*

- "Creating and Configuring WebLogic Domains Using WLST Offline" in *WebLogic Scripting Tool*.

Distributing WebLogic Server

This chapter provides details about the following topics:

- Using an ISV Redistribution License

- Distributing Files

- Distributing Maintenance Patches

- 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 Server Files

- Validating the ISV Redistribution License File

About the ISV Redistribution License

The WebLogic Server 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 Server—BEA provides a mechanism for

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

If you intend to redistribute WebLogic Server, 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 sends 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 Server software

- A customized `weblogic.jar` file, which links the WebLogic Server software to the ISV redistribution license

To run the ISV's value-added solution with the redistributed WebLogic Server software, ISV customers 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 Server 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 Server functionality as follows:

- The specific set of BEA components to which your customers can have access—for example, WebLogic Server—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 BEA components specified in that agreement.

- If your license distribution agreement is for WebLogic Server only, the ISV redistribution license prevents the run-time use of BEA products that are not included with WebLogic Server, such as WebLogic Portal.

- There is no enforced limit on the number of IP or database connections that can be made by an installation of the WebLogic Server software distributed by the ISV.

- There is no expiration date on the use of the BEA software distributed by the ISV.

- The ISV redistribution license is not tied to any specific IP address. Therefore, the BEA software that you distribute with an ISV redistribution license can be used on any machine.

Note that while you are constrained to use your WebLogic Server software on only the specific set of machines for which you are licensed, the ISV redistribution license does not similarly constrain your customers to use the WebLogic Server software on specific machines. In this sense, the ISV redistribution license may be somewhat less restrictive for your customers when compared to a license that is tied to one or more IP addresses.

The ISV redistribution license is intended to restrict the usage of WebLogic Server 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 Server software independently of the ISV's application.

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

To create an ISV redistribution license and link it to the WebLogic Server 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 Server 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 Server software as described in *WebLogic Server Installation Guide*.

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 "Preparing for Your Installation" in *WebLogic Server Installation Guide*.

3. Make a backup copy of the following file:

   `BEA_HOME/wlserver_10.0/server/lib/weblogic.jar`

4. Copy the `isv.jar` file you received from BEA into the BEA Home directory associated with this installation.

5. Open a command shell and go to the `server/bin` directory for WebLogic Server. For example:

   `prompt> cd BEA_HOME/wlserver_10.0/server/bin`

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

6. Run the appropriate script:

   – **Windows:** `setWLSEnv.cmd`

   – **UNIX:** `setWLSEnv.sh`

7. 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 Server Files

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

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

   prompt> `cd` *BEA_HOME*

2. Enter the appropriate command:

   – **Windows:**

   ```
   java -Xmx150m -Dbea.home=%BEA_HOME%
   -Dbea.jar=%BEA_HOME%\wlserver_10.0\server\lib\weblogic.jar install
   ```

   – **UNIX:**

   ```
   java -Xmx150m -Dbea.home=$BEA_HOME
   -Dbea.jar=$BEA_HOME/wlserver_10.0/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 Server installation to create a link 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 for which you ran 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.

Important Note if You Encounter JVM Errors

On some operating systems and JDKs, you might encounter an "Out of Memory Error" message. To address this error,

1. Retrieve a copy of the `weblogic.jar` file that you backed up, as described in "Step 1: Prepare to Create an ISV Redistribution License" on page 4-3 in step 3. Place this file in the `BEA_HOME/wlserver_10.0/server/lib` directory, overwriting the existing version of that file.

2. Increase the value for the `-Xmx` argument, and run the command again that is described in "Step 2: Extract the License Data and Link the WebLogic Server Files" on page 4-4. The `-Xmx` argument sets the maximum heap size in megabytes. For example, if you specify `-Xmx180m` in the command line, the maximum heap size is increased to 180 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 Server 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`, as `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 Server software that you plan to redistribute, you need to create a process for ensuring that your customers install the following items in the proper locations on their systems:

- 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 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 instructions 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 Server software.

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

Distributing Maintenance Patches

Publicly available, as well as private patches can optionally be included with your value-added solution and applied to your customer's target installation after installation is complete. Your customers can then install these patches by running Smart Update, or you can provide a script with your value-added solution that applies those patches automatically. By applying patches via a script, you can ensure that the patch maintenance process is seamless and transparent to your customer, while ensuring that the embedded WebLogic Server software operates at the appropriate maintenance level for your customers.

The process for distributing WebLogic Server software and required patches includes the following steps:

1. From Smart Update, log in to BEA Customer Support Web site, and download the patches required for the version of the WebLogic Server software embedded in your value-added solution.

2. Create the ISV redistribution license and link it to the WebLogic Server files that you are distributing, as described in "Using an ISV Redistribution License" on page 4-1.

3. Establish a location on your distribution media where the patch files are to be located.

4. Choose a mechanism for applying patches on the customer's system. The available mechanisms include:

   – The customer runs Smart Update and applies the patches individually, either via that tool's graphical or command-line interface. If you choose this mechanism, you need to prepare detailed instructions that explain how to run Smart Update, how to set the patch download directory, if necessary, and how to apply the patches.

– Create a script that invokes the Smart Update command-line interface and applies the patches that the download directory you have designated. This mechanism is typically the most reliable means to ensure that patches are applied correctly. This script can be standalone, which you instruct your customers to run after installation. Or you can create a shell script that first invokes the installation program—for example, a silent-mode installation script—then next invokes Smart Update.

5. When creating the procedure for installing the WebLogic Server files and ISV redistribution license on the end-user's system, make sure that the entire set of files located in the following directories are included in the installation as well:

– *BEA_HOME*/utils/bsu

– *BEA_HOME*/patch_weblogic100

– Directory designated as the patch download directory, if not a subdirectory of either of the two preceding directories

WARNING: These directories contain files required to run Smart Update and apply patches; without these directories on the end-user's system, patches cannot be applied.

For more information about the patch management capabilities of Smart Update, see Installing Maintenance Updates and Service Packs.

For information about creating a script to apply maintenance patches, see Using the Command-Line Interface and Best Practices for Distributing Maintenance Updates in *Installing Maintenance Updates and Service Packs*.

Service Packs and License Files

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 Server. As a result, customers who install your distribution of the initial release of WebLogic Server 9.1 can later use the Upgrade Installer to install a service pack without obtaining a new license.

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

Using the BEA Registry API

ISV applications that interoperate with WebLogic Server typically need to have specific information about the BEA software, for example, the current software version number or custom startup requirements. As BEA releases new versions and service packs of WebLogic Server, and as customers extend and modify their WebLogic Server installations, ISV software needs to detect changes in a way that results in minimal or no impact on the usage of that software.

The BEA Registry API is a mechanism that provides a uniform access method to information about BEA software installations that is contained in the BEA Registry file, `registry.xml`. The BEA Registry API gives ISV applications a manageable, predictable means of interoperating with BEA software as customer environments change.

This chapter contains the following topics:

- About the BEA Registry

- Using the Registry API

WARNING: Do not modify the contents of the BEA Registry file, `registry.xml`. Such modification may result in unpredictable system behavior.

About the BEA Registry

A BEA Registry file is created in each BEA Home directory on each machine where a BEA product has been installed, and it stores context data about the software installed on that machine, including:

- BEA product information—The BEA Registry tracks the following information for each BEA product installed:
  - Product name, version number, and installation directory
  - Service pack and patch level
  - BEA program group name
  - Start script information. For example, on Windows systems, this information includes the name of the program group under which the Start menu entry for this product is created, and the path to the directory that contains the Start menu entries.
  - List of installed subcomponents, for example, WebLogic Server examples
- Node Manager listen port
- Java 2 JDK information—The BEA Registry tracks the following Java 2 JDK information:
  - `JAVA_HOME`, `JAVA_VENDOR`, and version of the default Java 2 JDK
  - List of host directories for all Java 2 JDK installations
  - JVM version and vendor of a given Java home directory

Using the BEA Registry API to obtain this information is much more reliable and stable than other methods, such as examining the `weblogic.Admin` file and run-time MBeans. Regardless of how the volume and format of the information kept in the BEA Registry may change across successive releases of BEA software, the BEA Registry API returns data in a standard format.

Using the Registry API

The Registry API package contains the classes and interfaces summarized in Table 5-1. Click the name of a class or interface listed in this table to view the Javadoc for it.

Table 5-1 Registry API Package com.bea.plateng.gpr.registry

`ProductInfo`	Interface for returning information about the product installed in a BEA Home directory.
`ProductInfoImpl`	Class that implements the `ProductInfo` interface.

Table 5-1 Registry API Package com.bea.plateng.gpr.registry (Continued)

`RegistryInfo`	Interface for inspecting the BEA Registry file, `registry.xml`, in a BEA Home directory for installed products, components, and subcomponents.
`RegistryInfoImpl`	Class that implements the `RegistryInfo` interface.

To create an application that uses the Registry API, complete steps similar to the following:

1. Add the following JAR file to the system classpath on your machine:

   BEA_HOME/utils/bsu/commdev.jar

2. Create a Java application that includes the following `import` statements:

   ```
   import com.bea.plateng.gpr.registry.RegistryInfo;
   import com.bea.plateng.gpr.registry.RegistryInfoImpl;
   import com.bea.plateng.gpr.registry.ProductInfo;
   ```

3. In the Java application, declare the following two variables:

   – A variable of type `String`, which is set to the absolute path of the BEA Home directory

   – A variable of type `RegistryInfo`, which is a Java object for storing information obtained from the BEA Registry

   For example:

   ```
   String beaHomeDir = "c:\\bea";
       RegistryInfo regInfo;
   ```

   Note that as an alternative to creating a variable in which you set the absolute path of the BEA Home directory, you can invoke the `RegistryInfoImpl.getBEAHomes` method. This method returns an array of `String` objects that represent each BEA Home directory detected on the current machine.

4. Create an instance of a `RegistryInfo` object, which accepts the BEA Home directory as an input parameter. For example:

   ```
   try
       {
         regInfo = new RegistryInfoImpl(beaHomeDir);
       }
       catch(Exception e)
       {
         //log exception
   ```

```
        return;//cannot continue if we cannot parse the registry
    }
```

5. Invoke methods on the `RegistryInfo` object to obtain the desired information. For example, you can obtain a list of existing Java home directories, which are returned in an array of `String` objects. You can also return the names of installed BEA products in an array of `ProductInfo` objects.

For example:

```
//These are all the jdks in the registry.xml
    String jdks[] = regInfo.getJavaHomes();

//These are all the products found in the registry
    ProductInfo[] infos = regInfo.getProductInfos();
```

You can also return a specific product name using the `getProductInfo()` method. For example:

```
ProductInfo info = regInfo.getProductInfo("WebLogic Server",
"9.1.0.0");
```

Javadoc for the Registry API is available at the following URL:

http://e-docs.bea.com/common/docs100/javadoc/index.html

Supporting and Distributing BEA JRockit

This chapter includes the following topics:

About BEA JRockit

The BEA JRockit JDK 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 BEA JRockit Java Virtual Machine (JVM) and the BEA JRockit Mission Control tools.

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

BEA JRockit Mission Control is an extension to JRockit that includes a set of tools for monitoring, managing, profiling, and detecting memory leaks in your Java application without introducing the performance overhead normally associated with these types of tools. The BEA JRockit Mission Control functionality can always be available on-demand, and the small performance overhead is in effect only while the tools are running. To use the Mission Control, you need to obtain a separate license from BEA. Note that if you are distributing Mission Control, BEA provides the Mission Control ISV license, described in "Distributing BEA JRockit Mission Control" on page 6-3.

For complete details about JRockit, see the following:

- *Introduction to JRockit JDK*

- *BEA JRockit General Information*

- *Configuration and Tuning Guide*

- *Memory Management Guide*

- *Developing Java Applications*

- *Introduction to the JRockit Mission Control*

- *JRockit Management Console User Guide*

- *Using the JRockit Runtime Analyzer*

- *Using the JRockit Memory Leak Detector*

- *Using the JRockit Monitoring and Management APIs*

- *BEA JRockit 5.0 JDK Release Notes*

For a complete list of hardware/operating system configurations on which BEA JRockit 5.0 JDK is supported, see "BEA JRockit 5.0 Supported Configurations" in *BEA JRockit Supported Configurations*.

Obtaining BEA JRockit

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

`http://commerce.bea.com/products/weblogicjrockit/5.0/jr_50.jsp`

Support for BEA JRockit in WebLogic Applications

When you distribute WebLogic Server, the BEA JRockit JDK is included by default. BEA automatically provides support for your customers who use JRockit as part of a WebLogic Server 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 JRockit, see the following Web site:

`http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/support/supp_se`
`rvices/`

Distributing the BEA JRockit JDK as a Stand-alone Product

If you want to redistribute BEA JRockit as a stand-alone product, you must comply with certain terms and inform BEA of your intent. See the BEA JRockit End User License Agreement for details. Support contract for BEA partners are available. 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. Contact BEA for further details.

Optionally, your customers who use the JRockit JDK separately from WebLogic Server can buy a separate support contract from BEA. For more information, advise your customers to visit the following Web site:

http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/support target=new

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

- *Configuration and Tuning Guide*
- "Migrating Applications to BEA JRockit" in *Developing Java Applications*

Distributing BEA JRockit Mission Control

To enable ISVs to distribute BEA JRockit Mission Control, BEA provides the Mission Control ISV license. When you include this license in your distribution, your customers receive run-time usage access to all of the features provided as part of Mission Control, including the following:

- BEA JRockit Memory Leak Detector, a real-time profiling tool that detects memory leaks within Java applications running on BEA JRockit.

- BEA JRockit Runtime Analyzer (JRA), an on-demand "flight recorder" that produces detailed recordings about the JVM and the application it is running. The recorded profile can later be analyzed off line, using the JRA tool. Recorded data includes profiling of methods and locks, as well as garbage collection statistics, optimization decisions, and object statistics.

To obtain a Mission Control ISV license, contact BEA. When you create your distribution, you place the Mission Control ISV license file in the JRE subdirectory of the JRockit product home directory. For example:

```
C:\bea\jrockit90_150_04\jre\license.bea
```

For more information, see the BEA JRockit Mission Control documentation.

Preparing Customer Documentation

This chapter includes general advice and considerations on preparing customer documentation, as well as advice and considerations related to two specific 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 Server that are included: WebLogic Server, Web server plug-ins, BEA JRockit, maintenance patches, and so on. Store the version numbers of WebLogic Server and any relevant components in several easy-to-find locations. (When a customer needs to contact a support organization, having multiple records of the relevant version numbers for the software can be helpful.)

- 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 level of support provided by BEA may depend on your support contract with BEA. See Chapter 8, "About BEA Partner Support," for more information.)

If your customers need to periodically apply BEA-provided maintenance patches to the embedded WebLogic Server software, explain to customers how to obtain and apply those patches.

- Installation

  Document the installation application for your product, even if WebLogic Server is installed and configured in silent mode. Describe every step the customer needs to take, such as how to download the distribution from the internet, how to identify the appropriate CD from a physical container, and how the customer chooses an item 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. 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, Maintenance, and Configuration

If you are using the silent-mode installation, maintenance, and configuration features of WebLogic Server 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 Server components installed, maintenance level, and configuration details for various system resources.

- The results of the installation, such as a list of the directories and files added to the customer's system, including the files associated with the customer's domain (for example,

the files in the `user_projects` directory, patch files in the
BEA_HOME`/patch_weblogic100` and *BEA_HOME*`/utils/bsu` directories).

- 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, Maintenance, and Configuration

If your product is designed to be installed interactively—that is, if your customers must run the 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 Server.

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

- How to run Smart Update to apply maintenance patches. If the maintenance patches are included with your product distribution media, your documentation should identify the directory in which the patches are located and explain how to set that as the patch download directory in Smart Update. Your customers might also need to understand how to:

  – Obtain a BEA account and BEA login ID, and how to link that ID to a BEA support contract, if applicable

  – How to log in to the BEA Customer Support Web site, eSupport, to obtain patches

- 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 domain

  – How to locate the template or template extension

  – The meaning of the choices that you are instructing customers to make. If multiple options are available for a given entry, explain the advantages and disadvantages of each choice.

If you must include a cross-reference to related text in another part of the configuration documentation for BEA Products, be sure to make the cross-reference accurate and explicit. That is, make the cross-reference to a specific section in a document, and not

simply to the entire document. 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 template extensions, see Chapter 3, "Configuring and Installing WebLogic Applications.")

- The results of the installation, such as a list of the directories and files added to the customer's system, including the files associated with the customer's domain (that is, the files in the `user_projects` directory) and patch files.

- Locations and terms of the licenses installed, as well as instructions for extending the terms of a license.

About BEA Partner Support

This chapter includes the following topics:

- Partner Support Resources
- Installing Maintenance Updates

Partner Support Resources

BEA provides a rich set of resources to support members of our 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_Guid
e.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 that BEA makes available. Resources include the following:

- Welcome kit

- Product sales training—Specially designed classes, including the partner Webinar series that encompasses 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—A comprehensive partner information and business support hotline, offering direct 5x9 telephone or email assistance. Response Center staff direct callers to the BEA professional who can answer questions or provide information. The Center also serves as a conduit for technical support.

- Training discounts—Reduced fees for instructor-led, open-enrollment, technical education courses

- PartnerFlash, a monthly email newsletter that delivers partner updates and other news, such as how BEA is engaging its partners in successful businesses

- Partner sales toolkit

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 2-5, 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.

- BEA Download Center—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.

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:

- Telephone Hotline—Technical support hotline provides 5x9 developer-level help with technical problems encountered while using or implementing BEA products, or while developing your own product with BEA 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/.

- Newsgroups—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://forums.bea.com/bea/index.jspa

- BEA e-docs—BEA's comprehensive online product documentation site for all BEA products. Documentation includes release notes, reference information about supported hardware/software configurations, and detailed instructions for the installation, configuration, development, and administration of 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

Installing Maintenance Updates

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

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

Before you establish customer support policies for your software, make sure you have the answers to the following questions about WebLogic Server:

- Are all service packs and maintenance patches for the current release of WebLogic Server supported by your software product? Make sure you clearly identify the versions of WebLogic Server for which you certify your product.

- Which service packs or maintenance patches of WebLogic Server 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 Server or one of its components:

  – Do you include all the service packs or maintenance patches required by your customers in the WebLogic Server distribution that is part of your product?

  – Do you know how you will handle future service packs and maintenance patches that will need to be added to your customers' installations?

  – Do you know whether it is likely that your customers may encounter a conflict when applying a patch? Do you have instructions ready to help your customers resolve conflicts?

  One way of answering these questions is to simply redistribute your software with the latest version of WebLogic Server, to which all required patches are applied automatically when your value-added solution is installed. Another approach is to

include, in your software package, instructions for upgrading the WebLogic Server software. Find the solution that works best for you and your customers.

Note: 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 update their installations with service packs for WebLogic Server. As a result, customers who install your distribution of the initial release of WebLogic Server 9.1 can later use the Upgrade Installer to install a service pack without obtaining a new license.

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