BEA WebLogic Platform™®

Introducing Administration in WebLogic Platform 8.1

Version 8.1 Service Pack 6
Revised:June 2006

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

All other trademarks are the property of their respective companies.

Contents

Introducing Administration

Introducing Administration

This document introduces new WebLogic Platform users to the job of administration of a WebLogic Platform 8.1 environment. It summarizes the basic tasks and tools required to set up, maintain, and manage WebLogic Platform applications. The document also provides links to other sources of information about WebLogic Platform administration.

Topics include:

- Overview of WebLogic Platform Administration
- Creating WebLogic Domains
- Configuring, Managing, and Monitoring WebLogic Platform Applications
- Defining Administrators
- Managing Database Resources
- Accessing Log Files
- Automating Administration Tasks
- Moving Domains from Development to Production Environments

Overview of WebLogic Platform Administration

WebLogic Platform provides a comprehensive set of tools for administering your application environment. This section introduces you to that environment, describes the function of the

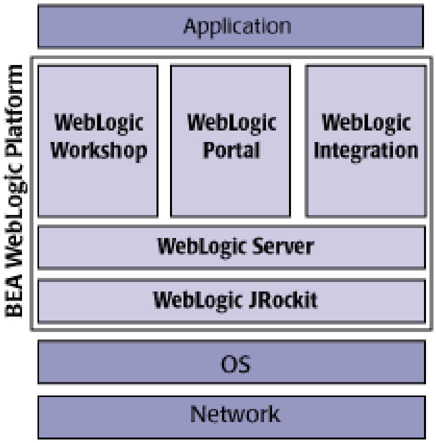
WebLogic Server domain in administration, explains the roles and responsibilities of an administrator, and provides a summary of administrative tasks and tools.

WebLogic Platform Application Environment

BEA WebLogic Platform consists of five component products that can be used independently, or in combination, as required by your application: WebLogic Server, WebLogic JRockit, WebLogic Workshop, WebLogic Portal, and WebLogic Integration.

The following figure shows the structure of these component products within a WebLogic Platform application environment.

Figure 1 WebLogic Platform Application Environment

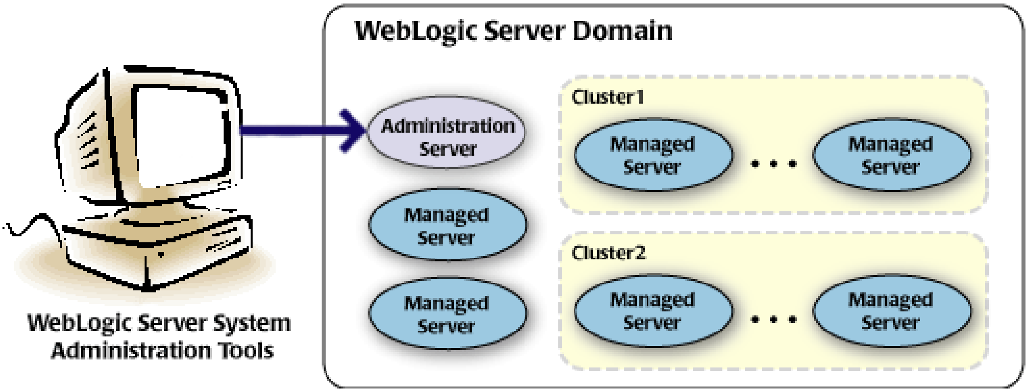


WebLogic Server provides the foundation for WebLogic Platform by hosting application services, such as Web server functionality, business components, and access to back-end enterprise systems in one central location. The WebLogic Workshop, WebLogic Portal, and WebLogic Integration components, and all applications built with these components, are run in the WebLogic Server run-time environment.

WebLogic Server Domain—The Basic Unit of Administration

Before you can develop and run a WebLogic Platform application, you must first create a WebLogic Server domain. A *domain* is the basic unit of administration for your WebLogic Platform applications.

Figure 2 WebLogic Server Domain



As shown in the previous figure, the basic domain infrastructure consists of one Administration Server and optional Managed Servers and clusters. It is possible to create a domain with only one instance of WebLogic Server. In this case, the single server provides the functionality for both the Administration Server and Managed Server.

The components of a domain are described in the following table.

Table 1 Domain Infrastructure Components

Feature	Description
Administration Server	A domain always includes one WebLogic Server instance that is configured as an Administration Server. The Administration Server provides a central point from which to manage the domain, and it provides access to WebLogic Server administration tools.
Managed Servers	All other WebLogic Server instances in a domain are called Managed Servers. Managed Servers host the application components and resources, which are also deployed and managed as part of the domain.
Clusters	A domain may also include WebLogic Server *clusters*, which are groups of WebLogic Server instances that work together to provide scalability and high availability for applications. Clusters can improve performance and provide failover should a server instance become unavailable.

For more information about WebLogic Server domains, see “Overview of WebLogic Server Domains” in *Configuring and Managing WebLogic Server* at the following URL:

http://e-docs.bea.com/wls/docs81/adminguide/overview_domain.html

WebLogic Platform Administrators' Roles and Responsibilities

The following table defines the roles and responsibilities of WebLogic Platform administrators at a high level.

Note: Specific responsibilities may overlap categories.

Table 2 WebLogic Platform Administrator Roles and Responsibilities

This administrator...	Must manage...
WebLogic Server Administrator	WebLogic Server and J2EE applications and resources.
Portal Administrator	Portal content, portal users, and the job of building portals with existing portal resources.
Integration Administrator	Integration applications and resources.
System Administrator	Hardware and software resources. Responsibilities may include tasks such as monitoring security configuration, managing allocation of user names and passwords, monitoring the use of disk space and other resources, performing backups, and setting up new hardware and software.
Application Administrator	Application-specific resources.
Database Administrator	Database-specific resources.

WebLogic Platform gives you flexibility in the way you define administrators. For example, you may want to create a hierarchy in which different administrators have varying degrees of access to domain components and functions. You can also create administrators who can, in turn, delegate administration tasks to other users. To learn more, see "Defining Administrators" on page 17.

WebLogic Platform Administration Tasks and Tools

WebLogic Platform provides a comprehensive set of tools for administering your application environment. Depending on the type of authority granted, an administrator can perform various administrative tasks, including those defined in the following table.

Table 3 WebLogic Platform Administration Tasks and Tools

The purpose of this task…	Is to…
Creating WebLogic Domains	Create a domain quickly and easily using the Configuration Wizard, a standalone Java application that can be run independently of WebLogic Server.
Configuring, Managing, and Monitoring WebLogic Platform Applications	Set up and maintain your WebLogic Platform application environment using the four administration consoles provided by WebLogic Platform. Each console is a Web browser-based, graphical user interface used for managing features of the application environment: • WebLogic Server Administration Console for WebLogic Server domains • WebLogic Administration Portal for WebLogic Portal applications • WebLogic Integration Administration Console for WebLogic Integration applications • WebLogic JRockit Management Console for running instances of WebLogic JRockit JVM
Defining Administrators	Define administrators for your application by adding users to administrator groups (with the WebLogic Platform administration tools).
Managing Database Resources	Manage WebLogic JDBC components, such as connection pools and data sources, as required by your database management system (DBMS).
Accessing Log Files	Monitor your run-time applications and use the tools provided by WebLogic Platform to identify the source of errors.
Automating Administration Tasks	Generate scripts that perform some of the work that must otherwise be performed manually by an administrator.
Moving Domains from Development to Production Environments	Move domains from a development environment to a production environment (after development is complete).

These tasks are described in more detail in the following sections.

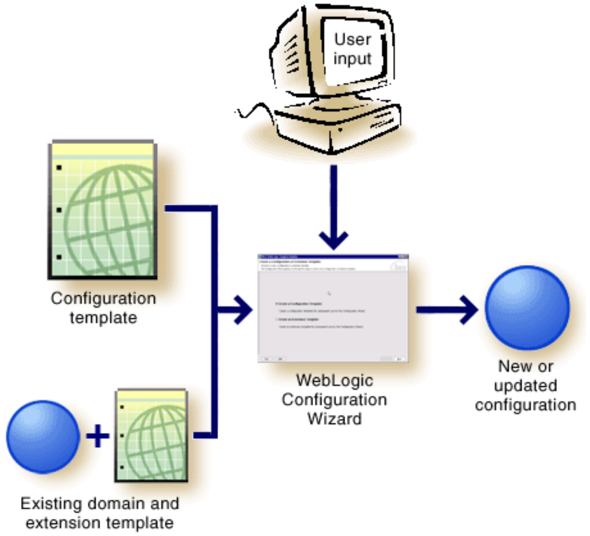
Note: In addition to the WebLogic Platform administration tasks and tools described in the following sections, BEA supports a Star Partner Program that provides other enterprise management solutions. For more information, see BEA PartnerNet at:

`http://partnernet.bea.com/`

Creating WebLogic Domains

Before you can develop and run a WebLogic Platform application, you must first create a domain. A domain defines the basic unit of administration for WebLogic Platform applications. The Configuration Wizard, shown in the following figure, simplifies the domain creation process.

Figure 3 Creating WebLogic Domains Using the Configuration Wizard


User input
Configuration template
WebLogic Configuration Wizard
New or updated configuration
Existing domain and extension template

The Configuration Wizard guides you through the process of creating or extending a domain, quickly and easily, by using a template that defines the domain configuration. To meet the needs of your target environment, you can customize the information used as input for the template. Specifically, you can modify the parameters governing any of the following characteristics of the domain:

- Infrastructure components, including managed servers, clusters, and physical host machines
- Database and messaging services—Java Database Connectivity (JDBC) and Java Message Service (JMS)
- Targets (servers and clusters)
- Security settings
- General environment and operating system features

After you finish creating your domain, start an instance of WebLogic Server in the domain. This server can be used to support application development, testing, or production.

Every domain that you create with the Configuration Wizard includes:

- A new configuration file, `config.xml`, that describes the infrastructure and basic network parameters governing all WebLogic Server instances in the domain
- Configuration of basic security features that are activated the first time the domain is booted

The domain may also include the following:

- Server startup scripts (for example, `startWebLogic.cmd` and `startWebLogic.sh`) that are populated with values that you specify via the Configuration Wizard
- A directory containing the applications provided by the template
- A directory containing information used to create and initialize a database
- Other files and directories to help you get started

To learn more about the Configuration Wizard:

- See *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL:

  `http://e-docs.bea.com/platform/docs81/confgwiz/index.html`

- Select **Help** from the Configuration Wizard interface.

Starting the Configuration Wizard

The Configuration Wizard can be started in one of the following ways:

- On a Windows platform, open the Start menu and choose Programs→BEA WebLogic Platform 8.1→Configuration Wizard.
- In a command-line shell, go to the `\common\bin` subdirectory of the product installation directory and enter one of the following commands:
  - Windows: `config.cmd`
  - UNIX: `sh config.sh`

When you use one of these methods, the Configuration Wizard is started in graphical mode. If you prefer, however, you can start the Configuration Wizard in console (command-line) or silent interface mode.

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

`http://e-docs.bea.com/platform/docs81/confgwiz/startcw.html`

Tutorials: Using the Configuration Wizard

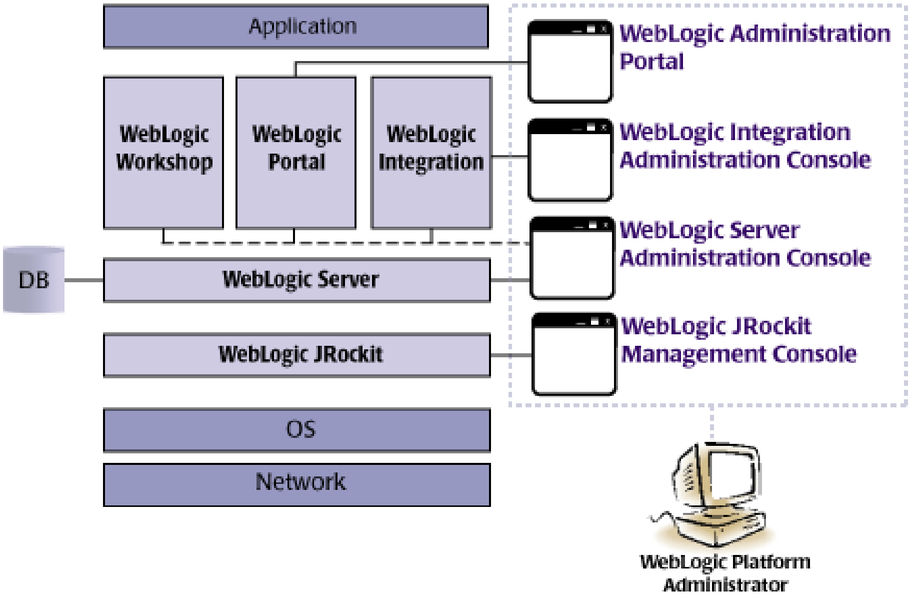
WebLogic Platform provides a set of tutorials that demonstrate how the Configuration Wizard can be used to create and update a domain quickly and easily. For more information, see "Tutorials: Using the Configuration Wizard" in *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL:

`http://e-docs.bea.com/platform/docs81/confgwiz/tutorials.html`

Configuring, Managing, and Monitoring WebLogic Platform Applications

WebLogic Platform provides several administration consoles that enable you to configure, manage, and monitor your WebLogic Platform application environment.

Figure 4 Administration Consoles for Configuring, Managing, and Monitoring WebLogic Platform Applications


Application
WebLogic Workshop
WebLogic Portal
WebLogic Integration
DB
WebLogic Server
WebLogic JRockit
OS
Network
WebLogic Administration Portal
WebLogic Integration Administration Console
WebLogic Server Administration Console
WebLogic JRockit Management Console
WebLogic Platform Administrator

The following table summarizes the purpose of each WebLogic Platform administration console shown in the previous figure.

Table 4 Administration Consoles for Configuring, Managing, and Monitoring WebLogic Platform

Use this administration console...	To configure, manage, and monitor...
WebLogic Server Administration Console	WebLogic Server domains containing multiple WebLogic Server instances, clusters, and applications.
WebLogic Administration Portal	Portal Web sites built with WebLogic Portal that include tools for controlling the behavior, content, and appearance of portals.
WebLogic Integration Administration Console	Software components and resources required for your WebLogic Integration applications.
WebLogic JRockit Management Console	Running instances of WebLogic JRockit JVM.

The WebLogic Platform administration consoles are described in more detail in the following sections.

WebLogic Server Administration Console

The WebLogic Server Administration Console enables you to configure, manage, and monitor domains containing multiple WebLogic Server instances, clusters, and applications. The WebLogic Server Administration Console consists of a Web application hosted by the Administration Server. You can use the WebLogic Server Administration Console to perform the following tasks for your domain:

- 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, for example, by defining users, groups, and roles
- Configure and deploy your applications
- Monitor server and application performance
- View server and domain log files
- View application deployment descriptors and edit selected run-time application deployment descriptor elements

The WebLogic Server Administration Server persists changes to attributes in the `config.xml` file for the domain you are managing.

To start the WebLogic Server Administration Server:

- From the Start menu, choose Server Admin Console to launch the tool in a Web browser. For example: Start→Programs→BEA WebLogic Platform 8.1→Examples→WebLogic Platform→Server Admin Console.
- From a Web browser, enter the URL `http://`*`hostname`*`:`*`port`*`/console`, where *`hostname`* is the DNS name or IP address of the server, and *`port`* is the address of the port on which the server is listening for requests (7001, by default). For example, if your server is running on your local machine, you can enter `http://localhost:7001/console`.
- From WebLogic Workshop, choose Tools→WebLogic Server→Start WebLogic Server.

Once the WebLogic Server Administration Console is started, you must log in to the administration console. By default, you can specify `weblogic` as both the username and password. To log out, select Logout.

To learn more about the WebLogic Server Administration Console:

- View a demo, "Configuring, Managing, and Monitoring WebLogic Server Domains," that shows how to use the WebLogic Server Administration Console.
- See *Configuring and Managing WebLogic Server* at the following URL:

  `http://e-docs.bea.com/wls/docs81/adminguide/index.html`
- 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`

WebLogic Administration Portal

The WebLogic Administration Portal enables you to configure, manage, and monitor Portal Web sites built with WebLogic Portal that include tools for controlling the behavior, content, and appearance of portals.

You can use the WebLogic Administration Portal to manage the following portal features:

- Users, Groups, and Roles—Define visitors and administrative users, group memberships, and global roles. Set up delegated administration and define visitor entitlements that determine which users can administer and view specific portal components and content.
- Interaction Management—Set up and maintain campaigns, content selectors, placeholders, and user segments.
- Portal Management—Set up and maintain desktops, books, pages, portlets, and other portal resources.
- Content Management—Set up and maintain repositories, types, and content.
- Configuration Settings—Set up and maintain the parameters governing portal server and portal search engine configuration.

To start the WebLogic Administration Portal:

- From the Start menu, choose WebLogic Administration Portal to launch the tool in a Web browser. For example: Start→Programs→BEA WebLogic Platform 8.1→Examples→ WebLogic Platform→WebLogic Administration Portal.

- From a Web browser, enter the URL `http://hostname:port/portal_appAdmin`, where *hostname* is the DNS name or IP address of the server, *port* is the address of the port on which the server is listening for requests (7001, by default), and *portal_app* is the name of the portal application. For example, if your server is running on your local machine and your portal application is called `e2ePortal`, you can enter: `http://localhost:7001/e2ePortalAdmin`.
- From WebLogic Workshop, open the portal application and choose Portal→Open Portal Administration.

Once the WebLogic Administration Portal is started, you must log in to the administration console. By default, you can specify `weblogic` as both the username and password. To log out, select Logout.

To learn more about the WebLogic Administration Portal:

- View a demo, "Configuring, Managing, and Monitoring WebLogic Portal Applications," that shows how to use the WebLogic Administration Portal.
- See *Getting Started With Portal Administration* at the following URL:

  `http://e-docs.bea.com/wlp/docs81/startadm/startadm.html`
- See the "WebLogic Administration Portal Tutorial" in the *WebLogic Administration Portal Online Help* at the following URL:

  `http://e-docs.bea.com/wlp/docs81/adminportal/tutorials/TutorialOV.html`
- See the *WebLogic Administration Portal Online Help* at the following URL:

  `http://e-docs.bea.com/wlp/docs81/adminportal/index.html`

WebLogic Integration Administration Console

You can use the WebLogic Integration Administration Console to manage the following integration features:

- Process Instance Monitoring—Monitor instances of business processes.
- Process Configuration—Configure business processes.
- Message Broker—Monitor Message Broker message-based communication.
- Event Generators—Create, view, and edit event generators.
- Worklist Administration—Administer and monitor worklist task instances.

- Application Integration—Monitor enterprise adapters.
- Trading Partner Management—Configure and manage trading partners.
- System Configuration—Configure system and security information.
- User Management—Configure users and roles that access integration system resources.
- Business Calendar Configuration—Create and edit the business calendars that determine user availability or timing of system events.

To start the WebLogic Integration Administration Console:

- From the Start menu, choose Integration Admin Console to launch the tool in a Web browser. For example: Start→Programs→BEA WebLogic Platform 8.1→Examples→WebLogic Platform→Integration Admin Console.
- From a Web browser, enter the URL `http://`*`hostname`*`:`*`port`*`/wliconsole`, where *`hostname`* is the DNS name or IP address of the server, and *`port`* is the address of the port on which the server is listening for requests (7001, by default). For example, if your server is running on your local machine, you can enter `http://localhost:7001/wliconsole`.
- From WebLogic Workshop, choose Tools→WebLogic Integration→WebLogic Integration Administration Console.

Once the WebLogic Integration Administration Console is started, you must log in to the administration console. By default, you can specify `weblogic` as both the username and password. To log out, select Logout.

To learn more about the WebLogic Integration Administration Console:

- View a demo, “Configuring, Managing, and Monitoring WebLogic Integration Applications,” that shows how to use the WebLogic Integration Administration Console.
- See *Managing WebLogic Integration Solutions* at the following URL:

  `http://e-docs.bea.com/wli/docs81/manage/index.html`

WebLogic JRockit Management Console

You can use the WebLogic JRockit Management Console to view real-time information about a running application that can be used during development. For example, you can see where memory is being consumed in an application’s life cycle. You can also view such information in a deployed environment. For example, you may want to monitor the system health of a running application server. The console enables you to do so.

To start the WebLogic JRockit Management Console:

1. Start WebLogic JRockit JVM with the `-Xmanagement` option.

   For more information, see "Starting and Configuring WebLogic JRockit JVM" in *Using WebLogic JRockit SDK* at the following URL:

   `http://e-docs.bea.com/wljrockit/docs142/userguide/start.html`

2. Make sure that the paths to the JRE and `.jar` file are in your classpath. Then, enter the following command:

   ```
   console
   ```

   If you prefer to invoke the WebLogic JRockit Management Console without the launcher, enter the following command, instead:

   ```
   java -jar jrockit-install-directory/console/ManagementConsole.jar
   ```

To learn more about the WebLogic JRockit Management Console, see "Using the WebLogic JRockit Management Console" in *Using WebLogic JRockit SDK* at the following URL:

`http://e-docs.bea.com/wljrockit/docs142/userguide/mancons.html`

Other Administration Tools

The following table lists tools other than the administration consoles, that are available for configuring, managing, and monitoring WebLogic Platform applications.

Table 5 Other Administration Tools

This tool...	Enables you to...
Node Manager	Start, shut down, restart, and monitor remote WebLogic Server instances. For more information, see "Overview of Node Manager" in *Configuring and Managing WebLogic Server* at the following URL: `http://e-docs.bea.com/wls/docs81/adminguide/nodemgr.html`
Command-line interface	Manage a WebLogic Server domain when it is impractical or undesirable to use the Administration Console. For more information, see "weblogic.Admin Command-Line Reference" in the *WebLogic Server Command Reference* at the following URL: `http://e-docs.bea.com/wls/docs81/admin_ref/cli.html`

Table 5 Other Administration Tools (Continued)

This tool...	Enables you to...
`config.xml`	Manually edit the contents of the `config.xml` file, the persistent store for configuration information. (Other system administration tools automatically save all configuration changes in the `config.xml` file.) **Note:** Although this method of configuration is not recommended, it may provide advantages in limited situations. If you decide to edit your `config.xml` file manually, be sure that the Administration Server is not running when you do so. For more information, see the *WebLogic Server Configuration Reference* at the following URL: `http://e-docs.bea.com/wls/docs81/config_xml/index.html`
Java utilities	Perform common tasks, such as deploying applications and testing DBMS configurations. For more information, see "Using the WebLogic Server Java Utilities" in the *WebLogic Server Command Reference* at the following URL: `http://e-docs.bea.com/wls/docs81/admin_ref/utils.html`
Ant tasks	Create powerful build scripts for demonstrating or testing your application with custom domains. In a development environment, perform common configuration tasks, such as starting and stopping WebLogic Server instances, and creating and configuring WebLogic Server domains. For more information, see "Using Ant Tasks to Configure a WebLogic Server Domain" in the *WebLogic Server Command Reference* at the following URL: `http://e-docs.bea.com/wls/docs81/admin_ref/ant_tasks.html`
Deployment tools	Deploy applications and standalone modules to WebLogic Server. For more information, see "Deployment Tools Reference" in *Deploying WebLogic Server Applications* at the following URL: `http://e-docs.bea.com/wls/docs81/deployment/tools.html`
SNMP	Use the Simple Network Management Protocol (SNMP) to communicate with enterprise-wide management systems. The WebLogic SNMP agent gathers WebLogic Server management data, converts it to SNMP communication modules (trap notifications), and forwards the trap notifications to third-party SNMP management systems. For more information, see the *WebLogic SNMP Management Guide* at the following URL: `http://e-docs.bea.com/wls/docs81/snmpman/index.html`
JMX	Write your own management components as one or more Java classes. This option is available for advanced Java programmers with knowledge of the JMX API from Sun Microsystems Inc. and WebLogic Server MBeans. For more information, see *Programming WebLogic Management Services with JMX* at the following URL: `http://e-docs.bea.com/wls/docs81/jmx/index.html`

Table 5 Other Administration Tools (Continued)

This tool…	Enables you to…
WebLogic Workshop configuration files	Configure WebLogic Workshop run-time and security parameters. For more information, see “Configuration File Reference” in the *WebLogic Workshop Help* at the following URL: `http://e-docs.bea.com/workshop/docs81/doc/en/workshop/reference/configfiles/navConfigurationFileReference.html`
WebLogic Integration Bulk Loader	Import, export, and delete trading partner data, including trading partner profiles, certificates from keystores, service definitions, and service profiles. For more information, see “Using the Trading Partner Bulk Loader” in *Managing WebLogic Integration Solutions* at the following URL: `http://e-docs.bea.com/wli/docs81/manage/bulkloader.html`

Defining Administrators

To define administrators using the WebLogic Platform administration tools, you add a *user* to an administrator *group*. Administrative privileges can be further defined by using *roles*. To summarize each of these key terms:

- A *user* is a person or software module, such as a Java client, that can be authenticated.
- A *group* is a collection of users who usually have something in common, such as working in the same department of a company.
- A *role* is a privilege granted to users or groups based on specific conditions.

To learn more about these components, see "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`

The following table describes the administrator groups with full access privileges that are provided for each of the WebLogic Platform components by default, and the roles to which they are assigned.

Table 6 Default Administrator Groups and Roles

Component	Default Administrator Group	Assigned Administrator Role	Capabilities
WebLogic Platform	Administrators[1]	Admin	• View the server configuration, including the encrypted values of encrypted attributes. • Modify the entire server configuration. • Deploy enterprise applications, startup and shutdown classes, and Web Application, EJB, J2EE Connector, and Web Service modules. • Start, resume, and stop servers by default.
WebLogic Server			
WebLogic Workshop			

Table 6 Default Administrator Groups and Roles (Continued)

Component	Default Administrator Group	Assigned Administrator Role	Capabilities
WebLogic Portal	PortalSystemAdministrators	PortalSystemAdministrator	• Administer portal-related information for all servers in a cluster. • Create additional roles.
WebLogic Integration	IntegrationAdministrators	IntegrationAdmin	• Administer integration-related information for all servers in a cluster. • Create additional roles using the administration console.

1. If you belong to the Administrators group, you can also administer WebLogic Portal and WebLogic Integration.

To learn more about default groups and roles, see "Users, Groups, and Roles Preconfigured in a Platform Domain" 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`

By default, the user `weblogic` is assigned to the administrator groups defined in the previous table, with full system administrator privileges. You can log in by specifying `weblogic` as the password.

To define additional administrators, use the administration tools described in the following table.

Table 7 Defining Administrators

To define...	Use...
WebLogic Server Administrator	When creating your WebLogic Server Domain, use the Configuration Wizard, as described in "Configuring Security" in *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL: `http://e-docs.bea.com/platform/docs81/confgwiz/secur.html` When your application is running, use the WebLogic Server Administration Console, as described in the following sections: • "Users and Groups" in *Securing WebLogic Resources* at the following URL: `http://e-docs.bea.com/wls/docs81/secwlres/usrs_grps.html` • "Security Roles" in *Securing WebLogic Resources* at the following URL: `http://e-docs.bea.com/wls/docs81/secwlres/secroles.html`
Portal Administrator	The WebLogic Administration Portal, as described in "How Do I Set Up a New Administrator?" in the *WebLogic Administration Portal Online Help* at the following URL: `http://e-docs.bea.com/wlp/docs81/adminportal/ManageUsersAndGroups/UserNewAdmin.html`
Integration Administrator	The WebLogic Integration Administration Console, as described in "User Management" in *Managing WebLogic Integration Solutions* at the following URL: `http://e-docs.bea.com/wli/docs81/manage/users.html`

By default, the users and groups you add through the WebLogic Server Administration Console are users of WebLogic Integration, WebLogic Portal, and WebLogic Workshop applications and resources. Conversely, as you add users through the WebLogic Integration or WebLogic Portal administration consoles, those users become WebLogic Server users by default.

Managing Database Resources

The following sections provide an overview of managing database resources:

- Overview of WebLogic JDBC
- PointBase—The Default Development Database

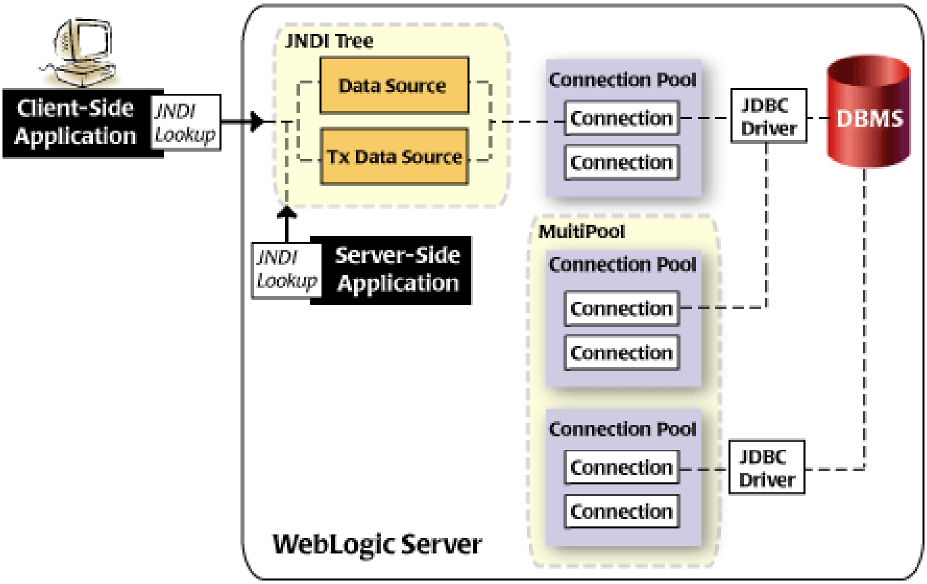
Overview of WebLogic JDBC

Java Database Connectivity (JDBC) is a standard Java API that consists of classes and interfaces written in the Java programming language. Application, tool, and database developers use JDBC to write database applications and execute SQL statements.

WebLogic JDBC enables programmers to interact seamlessly with different database management systems (DBMSs), such as Oracle, Microsoft SQL Server, Sybase, DB2, MySQL, and others.

The following figure shows how you can establish a connection to a DBMS using WebLogic JDBC components.

Figure 5 Components of WebLogic JDBC



The main components of WebLogic JDBC are a *connection pool*, a *MultiPool*, and a *data source*:

- A *connection pool* is a set of ready-to-use connections to your DBMS. When an application finishes using a connection, it returns the connection to the pool.
- A *MultiPool* is a group of connection pools that you can set up according to a high-availability or load-balancing algorithm. A MultiPool can be used only with WebLogic Server applications.

- A *data source* is an interface between an application and a connection pool. Client-side and server-side applications can obtain a DBMS connection using a data source on the Java Naming and Directory Interface (JNDI) tree.

  Data source objects can be defined with support for global transactions (TX Data Source). Such support is required for any applications in which distributed transactions are used.

A connection pool requires a *JDBC driver* to establish the physical connections between WebLogic Server and the DBMS. JDBC drivers implement the interfaces and classes of the JDBC API.

For a list of databases and drivers that are supported for use with WebLogic Platform, see "Supported Database Configurations" in *Supported Configurations for WebLogic Platform 8.1* at the following URL:

`http://e-docs.bea.com/platform/suppconfigs/configs81/81_over/supported_db.html`

To create and configure JDBC components, use the database administration tools described in the following table.

Table 8 Tools for WebLogic JDBC Database Administration

This tool...	Enables you to...
Configuration Wizard	Configure database connectivity when creating a domain. For more information, see "Configuring a Database Service" in *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL: `http://e-docs.bea.com/platform/docs81/confgwiz/jdbc.html`
WebLogic Server Administration Console	Add or edit configuration information for database connectivity when your application is running. For more information, see "Configuring Database Connectivity" in *WebLogic Server System Administration* at the following URL: `http://e-docs.bea.com/wls/docs81/admin.html`

To learn more about WebLogic JDBC, see:

- *Managing WebLogic Platform Database Resources* at the following URL:

  `http://e-docs.bea.com/platform/docs81/db_mgmt/db_resource_mgmt.html`

- *Programming WebLogic JDBC* at the following URL:

  `http://e-docs.bea.com/wls/docs81/jdbc/index.html`

- *dev2dev: Java Database Connectivity (JDBC)* at the following URL:

  http://dev2dev.bea.com/technologies/jdbc/index.jsp

PointBase—The Default Development Database

PointBase is the default database used for running the sample domains and creating a domain with the Configuration Wizard. When you start the server to run your applications, the PointBase server starts automatically.

You can administer the PointBase database using the PointBase console, or any third-party database visualization and management tool that can be connected to PointBase via JDBC.

To start the PointBase console:

- From the Start menu, choose PointBase Console to launch the tool. For example:

  Start→Programs→BEA WebLogic Platform 8.1→Examples→WebLogic Platform→ PointBase Console.

- On the command line, enter the following:

  ```
  java -cp
  WL_HOME\common\eval\pointbase\lib\pbclient44.jar;WL_HOME\common\
  eval\pointbase\lib\pbtools44.jar com.pointbase.tools.toolsConsole
  com.pointbase.jdbc.jdbcUniversalDriver
  ```

  WL_HOME specifies the root directory of your WebLogic Platform installation (for example, `c:\bea\weblogic81`).

Once the PointBase console starts, you may be prompted to enter several parameters required to connect to the database. For example, to start the PointBase console for the WebLogic Platform Tour, enter the following connection parameters:

- **Driver**: `com.pointbase.jdbc.jdbcUniversalDriver`
- **URL**: `jdbc:pointbase:server://localhost:9093/platform`
- **User**: `weblogic`
- **Password**: `weblogic`

To learn more about the PointBase database, see:

- *PointBase Console Guide* (PDF) at the following URL:

  http://e-docs.bea.com/wls/docs81/pdf/pbconsole.pdf

- *PointBase Developer's Guide* (PDF) at the following URL:

  `http://e-docs.bea.com/wls/docs81/pdf/pbdeveloper.pdf`

- *PointBase System Guide* (PDF) at the following URL:

  `http://e-docs.bea.com/wls/docs81/pdf/pbsystem.pdf`

PointBase is delivered with WebLogic Platform as a convenience for you; you are not required to use it. If you prefer to use another database, see "Switching Databases" in *Managing WebLogic Platform Database Resources* at the following URL:

`http://e-docs.bea.com/platform/docs81/db_mgmt/db_resource_mgmt.html`

Accessing Log Files

The WebLogic Server logging services provide facilities for writing, viewing, filtering, and listening for log messages. The information in log files is useful for detecting and troubleshooting problems, and monitoring your applications.

The WebLogic Platform components use the log files to provide information about events such as the deployment of new applications or the failure of one or more subsystems.Your application can also use them to communicate its status, respond to specific events, and record debugging information.

This section describes the log files, and the debugging and auditing log facilities that are available for troubleshooting and monitoring your applications. Topics include:

- Standard Log Files
- Database Log Files
- Debugging and Auditing Log Facilities

To learn more about using the WebLogic Server logging services, see *Using WebLogic Logging Services* at the following URL:

`http://e-docs.bea.com/wls/docs81/logging/index.html`

For a list of valid messages, along with the cause and suggested action for each, see the *WebLogic Server Message Catalogs* at the following URL:

`http://e-docs.bea.com/wls/docs81/messages/index.html`

Standard Log Files

The following table summarizes the standard log files that can be generated for each WebLogic Server instance.

Note: The default location of the log file is specified relative to the server root directory. For more information about the server root directory, see “A Server’s Root Directory” in *Configuring and Managing WebLogic Server* at the following URL:

`http://e-docs.bea.com/wls/docs81/adminguide/overview_domain.html#server_root_directory`

Table 9 Standard Log Files

The following log(s)...	Records...	Default Location in the Server Root Directory
Server Log	Events such as the startup and shutdown of servers, the deployment of new applications, or the failure of one or more subsystems.	`*server-name*/*server-name*.log`
Domain Log	Messages that are generated on multiple server instances within the domain.	`wl-domain.log`
Standard Out and Standard Error	Subset of messages generated by WebLogic Server instance. By default, `WARNING` and `ERROR` level messages are recorded.	N/A
HTTP Access Log	HTTP requests.	`logs/access.log`
WebLogic Workshop Logs	Events related to WebLogic Workshop.	• `workshop.log` • `workshop_debug.log` • `workshop_errors.log` • `netui.log`

Table 9 Standard Log Files (Continued)

The following log(s)...	Records...	Default Location in the Server Root Directory
Embedded LDAP Logs	Embedded LDAP server events.	`server-name/ldap/log/...` • `EmbeddedLDAP.log` • `EmbeddedLDAPAccess.log`
Node Manager Logs	Events related to the Node Manager. Node Manager logs are created only when the Node Manager is running.	`common/nodemanager/...` • `NodeManagerLogs/NodeManagerInternal/nm_hostname_date-time.log` • `NodeManagerLogs/domain_serverName` • `NodeManagerClientLogs/serverName`

Note: Each WebLogic Server instance also maintains a Java Transaction API (JTA) log: a repository for information about committed transactions coordinated by the server that may not have been completed. WebLogic Server uses the transaction log when recovering from system crashes or network failures. Because the transaction log is a binary file, you cannot view it directly.

The following sections describe each of the standard log files in more detail.

Server Log

The server log records information about events such as the startup and shutdown of servers, the deployment of new applications, or the failure of one or more subsystems. The following code is a sample excerpt from the server log file:

```
1####<Nov 13, 2003 3:59:44 PM EST> <Notice> <WebLogicServer> <MyComputer>
 <cgServer> <main> <<WLS Kernel>> <> <BEA-000327> <Starting WebLogic Admin
 Server "cgServer" for domain "sample">
2#####<Nov 13, 2003 3:59:44 PM EST> <Info> <WebLogicServer> <MyComputer>
<cgServer> <main> <<WLS Kernel>> <> <BEA-000214> <WebLogic Server "cgServer"
 version:
WebLogic Server 8.1 SP2  Tue Oct 21 22:57:26 PDT 2003 303940
WebLogic XMLX Module 8.1 SP2  Tue Oct 21 22:57:26 PDT 2003 303940
...
3####<Nov 18, 2003 4:01:46 PM EST> <Info> <Management> <MyComputer>
 <cgServer> <main> <<WLS Kernel>> <> <BEA-141187> <Java system properties
 are defined as follows:
...
java.home = k:\bea\JDK141~2\jre
java.io.tmpdir = M:\Temp\2\
java.runtime.name = Java(TM) 2 Runtime Environment, Standard Edition
java.runtime.version = 1.4.1_05-b01
java.security.policy = k:bea\WEBLOG~1\server\lib\weblogic.policy
java.specification.name = Java Platform API Specification
java.specification.vendor = Sun Microsystems Inc.
java.specification.version = 1.4
javax.rmi.CORBA.PortableRemoteObjectClass =
 weblogic.iiop.PortableRemoteObjectDelegateImpl
...
4####<Nov 19, 2003 3:54:38 PM EST> <Warning> <EJB> <MyComputer> <cgServer>
 <main> <<WLS Kernel>> <> <BEA-010061> <The Message-Driven EJB:
 ProcessTrackingEventErrorListener is unable to connect to the JMS
 destination: wli.internal.tracking.buffer_error. The Error was:
[EJB:011010] The JMS destination with the JNDI name:
wli.internal.tracking.buffer_error could not be found. Please ensure that
the JNDI name in the weblogic-ejb-jar.xml is correct, and the JMS
destination has been deployed.>
5####<Nov 18, 2003 4:20:27 PM EST> <Error> <WLW> <MyComputer> <cgServer>
 <ExecuteThread: '14' for queue: 'weblogic.kernel.Default'> <<anonymous>>
 <BEA1-0096EDCD7454B16BE9F2> <000000> <Process Tracking init failed for:
/test6/stress/StressProcess.jpd
javax.transaction.TransactionRolledbackException: Error in ejbCreate::
 javax.ejb.CreateException: [WLI-Core:484042]ProcessTracking failed to
 initialize the buffer.javax.naming.NameNotFoundException: While trying to
 lookup 'wli.internal.instance.info.buffer' didn't find subcontext
 'instance' Resolved wli.internal; remaining name 'instance/info/buffer'
       at
weblogic.jndi.internal.BasicNamingNode.newNameNotFoundException(BasicNamin
gNode.java:858)...>
```

As shown, whenever a WebLogic Server instance writes a message to the log file, the first line begins with ####, followed by a set of message attributes, enclosed in angle brackets. The following provides a brief description of each message by key:

1. Status information pertaining to the server startup. The severity level `NOTICE` indicates that logged message is for informational purposes.
2. Information about the server versions. The severity level `INFO` indicates that logged message is for informational purposes.
3. Information about the Java system properties. The severity level `INFO` indicates that logged message is for informational purposes.
4. Warning indicating that a JMS destination was not found. The severity level `WARNING` indicates that the logged message may or may not affect normal operation.
5. Error indicating that a specific transaction has been rolled back. The severity level `ERROR` indicates that the logged message requires attention.

The following table summarizes the message attributes for which values are stored in the server log.

Table 10 Server Log Message Attributes

Attribute	Example
Timestamp	`<Nov 13, 2003 3:59:44 PM EST>`
Severity	`<Notice>`
Subsystem	`<WebLogicServer>`
Machine Name	`<MyComputer>`
Server Name	`<cgServer>`
Thread ID	`<main>`
User ID	`<<WLS Kernel>>`
Transaction ID	`<>` **Note:** This field is empty if there is no transaction context.
Message ID	`<BEA-000327>`
Message Text	`<Starting WebLogic Admin Server "cgServer" for domain "sample">`

For complete details about message attributes and valid severity levels, see "Overview of WebLogic Server Log Messages and Log Files" in "Server Log" in the *WebLogic Server Administration Console Online Help* at the following URL:
`http://e-docs.bea.com/wls/docs81/ConsoleHelp/logging.html`

By default, each WebLogic Server instance maintains a server log, which is written to the following file in the server root directory: `server-name/server-name.log`.

To configure a server log file location and other settings, open the WebLogic Server Administration Console and choose the Server→Logging→Server tab. For more information about configuring the server log, see "Server→Logging→Server" in the *WebLogic Server Administration Console Online Help* at the following URL:

`http://e-docs.bea.com/wls/docs81/ConsoleHelp/domain_server_logging_server.html`

You can view the server log from either the WebLogic Server Administration Console or a text editor.

To learn more about viewing and using the server log, see "Server Log" in the *WebLogic Server Administration Console Online Help* at the following URL:

`http://e-docs.bea.com/wls/docs81/ConsoleHelp/logging.html`

Domain Log

In addition to having messages written to the local server log file, you can configure the logging services to collect the messages generated on multiple server instances in a single, domain-wide message log. This *domain log* enables you to monitor or archive the overall status of the domain through a single file.

The format of the messages written to the domain log is similar to that of the server log, as described in "Server Log" on page 25.

By default, each WebLogic Server instance sends all messages of severity level `ERROR` or higher to the domain log. It is written, by default, to the following file in the server root directory: `wl-domain.log`.

To configure the domain log:

- Update the domain log file name and attributes by opening the WebLogic Server Administration Console and choosing the Domain→Configuration→Logging tab. For more information, see "Domain→Configuration→Logging" in the *WebLogic Server Administration Console Online Help* at the following URL:

http://e-docs.bea.com/wls/docs81/ConsoleHelp/domain_domain_config_logging.html

- Define a custom domain log filter to control the type of messages recorded in the domain log. For more information, see "Domain Log Filters" in the *WebLogic Server Administration Console Online Help* at the following URL:

  http://e-docs.bea.com/wls/docs81/ConsoleHelp/domain_log_filters.html

- Specify whether or not messages are recorded in the domain log and, if so, which domain filter is to be used. To do so, open the WebLogic Server Administration Console and choose the Server→Logging→Domain tab. For more information, see "Server→ Logging→Domain" in the *WebLogic Server Administration Console Online Help* at the following URL:

  http://e-docs.bea.com/wls/docs81/ConsoleHelp/domain_server_logging_domain.html

You can view the domain log from either the WebLogic Server Administration Console or a text editor.

To learn more about viewing and using the domain log, see "Server Log" in the *WebLogic Server Administration Console Online Help* at the following URL:

http://e-docs.bea.com/wls/docs81/ConsoleHelp/logging.html

Standard Out and Standard Error

Each WebLogic Server instance sends a subset of its messages to standard out and standard error. In certain situations, standard out and standard error may provide important information about JVM errors, such as out-of-memory errors, that may not have been captured in the log file.

The format of the messages written to standard out and standard error is similar to that of the server log, as described in "Server Log" on page 25. However, when a WebLogic Server instance writes a message to standard out, the output does not include the #### prefix or the following message attributes: Server Name, Machine Name, Thread ID, and User ID fields. The following excerpt shows how a message from the server log example, provided in "Server Log" on page 25, is printed to standard out:

```
<Nov 13, 2003 3:59:44 PM EST> <Notice> <WebLogicServer> <BEA-000327>
 <Starting WebLogic Admin Server "cgServer" for domain "sample">
```

To control the types of messages written to standard out by a server instance, choose the Server→ Logging→Server tab in the WebLogic Server Administration Console. For more information, see “Server→Logging→Server” in the *WebLogic Server Administration Console Online Help* at the following URL:

http://e-docs.bea.com/wls/docs81/ConsoleHelp/domain_server_logging_server.html

By default, you can view standard out and standard error messages in the shell in which you are running the server instance.

Note: If you start a Managed Server with the Node Manager, the Node Manager redirects the standard error for the server instance to a file. In this case, you can view the standard error for the Managed Server by choosing Domain→Server→Remote Start Output→View Server error output.

To learn more about viewing and using standard out and standard error, see “Server Log” in the *WebLogic Server Administration Console Online Help* at the following URL:

http://e-docs.bea.com/wls/docs81/ConsoleHelp/logging.html

HTTP Access Log

The HTTP access log records all HTTP requests. The following code is a sample excerpt from the HTTP access log:

```
127.0.0.1 - john [17/Nov/2003:16:26:51 -0500] "GET
/e2ePortalProject/Controller.jpf HTTP/1.1" 200 805
127.0.0.1 - - [18/Nov/2003:16:36:07 -0500] "GET /sampleTool/PropertySetWS
HTTP/1.1" 401 1526
```

The messages provide status information about each HTTP access request, for example, an HTTP GET operation. The following table summarizes the message attributes for which values are stored in the HTTP access log.

Table 11 Server Log Message Attributes

Attribute	Example
IP address	`127.0.0.1`
User ID	`john` **Note:** A dash (—) specifies an anonymous user.
Timestamp	`[17/Nov/2003:16:26:51 -0500]`
HTTP action	`GET /e2ePortalProject/Controller.jpf HTTP/1.1`

Table 11 Server Log Message Attributes (Continued)

Attribute	Example
HTTP status code	`200`
Transfer length of message	`805` **Note:** If no data is transferred, the field length is 0.

Note that in the previous example, a status code of `200` indicates a successful request. A status code of `401` indicates that the request is unauthorized. For more information about HTTP status codes, see the *Hypertext Transfer Protocol -- HTTP/1.1* functional specification, available from the *W3C Consortium* Web site at the following URL:

http://www.w3.org/Protocols

By default, each WebLogic Server instance maintains an HTTP access log, which is written to the following file in the server root directory: `logs/access.log`.

To configure the HTTP access log file location and other settings, open the WebLogic Server Administration Console and choose the Server→Logging→HTTP tab. For more information, see "Server→Logging→HTTP" in the *WebLogic Server Administration Console Online Help* at the following URL:

http://e-docs.bea.com/wls/docs81/ConsoleHelp/domain_server_logging_http.html

To learn more about the HTTP access log, see "Setting Up HTTP Access Logs" in "Configuring Web Server Functionality for WebLogic Server" in *Configuring and Managing WebLogic Server* at the following URL:

http://e-docs.bea.com/wls/docs81/adminguide/web_server.html

WebLogic Workshop Logs

WebLogic Workshop generates a set of log files in which to capture status and error messages related to the WebLogic Workshop run-time software and applications.

The following code is a sample excerpt from a WebLogic Workshop log file:

```
04 Nov 2003 16:30:11,620 INFO  e2ePortal      : ConversationID=null;
 Protocol=java-call; URI=/e2ePortalProject/joindb/UsersDBControl.jcx;
 Method=lookupUser; Phase=none; Callback=null
```

WebLogic Workshop uses the log4j Java logging facility developed by the Jakarta Project of the Apache Foundation. Log4j defines three components: loggers, appenders, and layouts. These components enable you to log messages according to message type and level, specify the file location, and control message formatting. For more information about log4j, see the Apache Jakarta Project Web site at the following URL:

`http://jakarta.apache.org/log4j/docs/index.html`

The following table defines the log files generated by WebLogic Workshop.

Table 12 WebLogic Workshop Log Files

This log file...	Records...
`workshop.log`	All internal logging messages generated by the WebLogic Workshop run-time software and by user code in WebLogic Workshop applications
`workshop_debug.log`	Debugging information
`workshop_errors.log`	Exceptions
`netui.log`	Page flow information

By default, the WebLogic Workshop log files are written to the server root directory.

To configure the WebLogic Workshop log files, edit the `workshopLogCfg.xml` log4j configuration file located in *`BEA_HOME`*`/weblogic81/common/lib`. You can override the default configuration location using the `log4j.configuration` Java property. For example, on the command line used to start WebLogic Server, you can specify the following command-line argument:

`-Dlog4j.configuration=`*`path-to-config-file`*.

To learn more about the WebLogic Workshop logs, see "workshopLogCfg.xml Configuration File" in the *WebLogic Workshop Help* at the following URL:

`http://e-docs.bea.com/workshop/docs81/doc/en/workshop/reference/configfiles/con_knexLogCfg_xml_ConfigurationFile.html`

Embedded LDAP Logs

The embedded LDAP server is the default security provider database for the WebLogic Authentication, Authorization, Credential Mapping, and Role Mapping providers. It contains information about users, groups, group memberships, security roles, security policies, and credential mapping.

The embedded LDAP server generates a set of log files in which it can capture status and error messages related to its execution. The following code is a sample excerpt from an Embedded LDAP log file:

```
[01/Apr/2003 15:15:42 EST] INFO    - LDAPServer:  VDE Engine Starting
[01/Apr/2003 15:15:42 EST] DEBUG   - ACLChecker:  Root User is: cn=Admin
[01/Apr/2003 15:15:42 EST] INFO   - BackendHandler:  Initializing Adapters
```

The following table summarizes the message attributes for which values are stored in the Embedded LDAP log.

Table 13 Embedded LDAP Log Message Attributes

Attribute	Example
Timestamp	`[01/Apr/2003 15:15:42 EST]`
Severity	`INFO`
Subsystem	`LDAPServer`
Message text	`VDE Engine Starting`

The following table lists the Embedded LDAP logs that are generated by default.

Table 14 Embedded LDAP Logs

This log file...	Records information about...
`EmbeddedLDAP.log`	Embedded LDAP server
`EmbeddedLDAPAccess.log`	Controlling access to the Embedded LDAP server

The embedded LDAP log files are created in the following directory in the server root directory: *`server-name`*`/ldap/log.`

To learn more about Embedded LDAP, see “Managing the Embedded LDAP Server” in *Managing WebLogic Security* at the following URL:

http://e-docs.bea.com/wls/docs81/secmanage/ldap.html

Node Manager Logs

Node Manager is a Java program provided with WebLogic Server that enables you to start, shut down, restart, and monitor remote WebLogic Server instances. When running, Node Manager generates log files for the following components:

- Node Manager—Records its own startup and status messages in a new file created each time it starts. By default, these log files are written to the *WL_HOME*/common/nodemanager/NodeManagerLogs/NodeManagerInternal directory, and are named nm_hostname_*date-time*.log, where *date-time* indicates the time at which Node Manager started.

  Note: Because Node Manager creates a new log file each time it starts, you should periodically remove the NodeManagerLogs subdirectory to reclaim the space used by old log files.

- Managed Server—If you start a server instance with Node Manager, the Managed Server records startup and error messages about itself. By default, these log files are written to the *WL_HOME*/common/nodemanager/NodeManagerLogs/*domain_serverName* directory, where *domain_serverName* designates the name of the domain and Managed Server. The NodeManagerLogs directory contains one subdirectory for each Managed Server started by the Node Manager process on the machine.

  You can view the standard output and error messages for a server, as well as the Node Manager log messages for a particular Managed Server, by choosing the Server→ Monitoring→Remote Start Output tab in the WebLogic Server Administration Console.

- Node Manager Client—Records information that is used by the Node Manager client residing in the Administration Server. Each log entry corresponds to an attempt to carry out some action, such as starting or killing the server process. By default, the log files are written to the *WL_HOME*/common/nodemanager/NodeManagerLogs/NodeManagerClientLogs directory. The NodeManagerClientLogs directory contains one subdirectory for each Managed Server started by the Node Manager process on the machine. The name of the log file includes a timestamp that indicates the time at which the action was attempted.

For more information about Node Manager log files, see “Troubleshooting Node Manager” in “Configuring, Starting, and Stopping Node Manager” in *Configuring and Managing WebLogic Server* at the following URL:

http://e-docs.bea.com/wls/docs81/adminguide/confignodemgr.html

Database Log Files

The JDBC log records all JDBC-related activities. The following code is a sample excerpt from the JDBC log file:

```
DRVR OPER Enabled logging (moduleMask 0x0fffffff, categoryMask 0x0000008f)
DRVR OPER OracleResultSetImpl.getString(columnIndex=1)
DRVR OPER OracleConnection.getTransactionIsolation() returned 2
DRVR OPER OracleConnection.getDefaultFixedString() returning false
DRVR SQLS SQL: "SELECT 1 FROM DUAL"
DRVR SQLS Input SQL:  "SELECT 1 FROM DUAL"
```

The JDBC log file is generated by the JDBC driver. For information about the JDBC log message attributes, see the appropriate JDBC driver documentation.

By default, JDBC logging is disabled. Once enabled, the JDBC log is written, by default, to the following file in the server root directory: `server-name/jdbc.log`.

To enable the JDBC log and configure its location, choose the Server→Logging→JDBC tab in the WebLogic Server Administration Console. For more information, see "Server→Logging→ JDBC" in the *WebLogic Server Administration Console Online Help* at the following URL:

http://e-docs.bea.com/wls/docs81/ConsoleHelp/domain_server_logging_jdbc.html

To learn more about JDBC and JDBC logs, see "JDBC" in the *WebLogic Server Administration Console Online Help* at the following URL:

http://e-docs.bea.com/wls/docs81/ConsoleHelp/jdbc.html

PointBase is the default database management system installed with WebLogic Server. By default, when the PointBase server is started, log messages are redirected to the following log file in the server root directory: `pointbase/log/pointbase.log`. You can control logging features by using the PointBase parameters. For more information, see the *PointBase System Guide* (PDF) at the following URL:

http://e-docs.bea.com/wls/docs81/pdf/pbsystem.pdf

Similar log files are generated for other database management tools, such as Oracle. For information about configuring the logging services for third-party database management tools, see the appropriate third-party documentation.

Debugging and Auditing Log Facilities

To enable you to augment the information captured in the log files, WebLogic Platform provides a convenient set of debugging and auditing log facilities. The following sections describe each of these facilities in more detail.

Debugging Facilities

The debugging facilities capture verbose descriptions of low-level activity within an application.

Application Debugging

While your application is under development, you can use the DEBUG severity level to categorize and capture debug messages. Messages of the DEBUG severity level are automatically sent to the server log file.

Note: Unlike Log4j, which is a third-party logging service that enables you to dynamically exclude log messages based on level of severity, the WebLogic Server log includes all levels of messages generated by your application.

The following code is an example of a debug message:

```
####<Nov 13, 2003 4:40:20 PM EST> <Debug> <HTTP> <MyComputer> <cgServer>
 <ExecuteThread: '1' for queue: 'weblogic.socket.Muxer'> <<WLS Kernel>> <>
 <BEA-101147> <HttpServer(1188855,null default ctx,cgServer) Found no
 context for "/RMwebservice/RMHelloWorld". This request does not match the
 context path for any installed Web applications, and there is no default
 Web application configured.>
```

You also can configure WebLogic Server to send debug messages to standard out. For more information, see "Specifying Which Messages a Server Sends to Standard Out" in the *Administration Console Online Help* at the following URL:

http://e-docs.bea.com/wls/docs81/ConsoleHelp/logging.html#Specifying_Messages_Standard_Out

It is recommended that you create a debug mode for your application when using the DEBUG severity level. For more information, see “Writing Debug Messages” in “Writing Messages to the WebLogic Server Log” in *Using WebLogic Logging Services* at the following URL:

http://e-docs.bea.com/wls/docs81/logging/writing.html

LDAP Debugging

If you want to capture LDAP debug messages related to authentication and authorization, set up an LDAP debug log called `ldap_trace.log`. To create this log, open the `config.xml` file and add the following line to the definition of the `<Server>` tag:

```
<ServerDebug DebugSecurityAtn="true" DebugSecurityAtz="true"
DebugEmbeddedLDAPLogLevel="n" Name="myserver"/>
```

In order for this change to take effect, you must restart the Administration Server and, if necessary, the Managed Server for which you are capturing LDAP debug messages.

For more information about updating the `config.xml` file, see the *WebLogic Server Configuration Reference* at the following URL:

`http://e-docs.bea.com/wls/docs81/config_xml/index.html`

Auditing Facilities

The auditing facilities allow you to generate an electronic trail by collecting, storing, and distributing information about operating requests and the outcomes of those requests for the purposes of non-repudiation. You can configure either a WebLogic Auditing provider or a custom Auditing provider in a security realm.

Note: The default security realm (`myrealm`) does not include a pre-configured Auditing provider configured.

The following table defines the events logged by the WebLogic Auditing provider.

Table 15 Events Logged By the WebLogic Auditing Provider

This event...	Indicates...
`AUTHENTICATE`	A simple authentication (username and password) has been performed.
`ASSERTIDENTITY`	A perimeter authentication (based on tokens) has been performed.
`USERLOCKED`	A user account has been locked after multiple invalid login attempts.
`USERUNLOCKED`	The lock on a user account has been cleared.
`USERLOCKOUTEXPIRED`	The lock on a user account has expired.

The WebLogic Auditing provider saves all auditing information in `WL_HOME/yourdomain/yourserver/DefaultAuditRecorder.log`. Although a separate Auditing provider is configured for each security realm, each server writes auditing data to its own log file in the server directory.

For more information about configuring the WebLogic Auditing provider and the custom Auditing provider, see "Configuring Security Providers" in *Managing WebLogic Security* at the following URL:

`http://e-docs.bea.com/wls/docs81/secmanage/providers.html`

Automating Administration Tasks

Some administration tasks can be automated using the methods described in the following table.

Table 16 Methods for Automating Administration Tasks

Administration Tasks	Method	Description
Creating WebLogic Domains	Silent-Mode Configuration	Create a configuration script and execute the Configuration Wizard as a noninteractive process. This method enables you to define the configuration information for a domain once and then duplicate that domain on multiple machines. For more information, see "Creating a Script for Silent-Mode Configuration" in *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL: `http://e-docs.bea.com/platform/docs81/confgwiz/silent.html`
	Template Builder	Create custom configuration and extension templates that can be used later for creating and updating domains with the Configuration Wizard. Template Builder enables you to define and propagate a standard domain across a development project, or to distribute a domain with an application that has been developed to run on it. For more information, see *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL: `http://e-docs.bea.com/platform/docs81/confgwiz/index.html`

Table 16 Methods for Automating Administration Tasks (Continued)

Administration Tasks	Method	Description
Configuring, Managing, and Monitoring WebLogic Platform Applications	WebLogic Server Command-Line Interface	Manage a WebLogic Server domain by issuing commands manually on the WebLogic Server command line. This method enables you to manage a domain without the Administration Console, when use of the console is neither practical nor desirable. For more information, see "weblogic.Admin Command-Line Reference" in the *WebLogic Server Command Reference* at the following URL: `http://e-docs.bea.com/wls/docs81/admin_ref/cli.html`
	Java Utilities	Perform common tasks, such as deploying applications and testing DBMS configurations. For more information, see "Using the WebLogic Server Java Utilities" in the *WebLogic Server Command Reference* at the following URL: `http://e-docs.bea.com/wls/docs81/admin_ref/utils.html`
	Ant Tasks	Create powerful build scripts for demonstrating or testing your application with custom domains. In a development environment, perform common configuration tasks, such as starting and stopping WebLogic Server instances, and creating and configuring WebLogic Server domains. For more information, see "Using Ant Tasks to Configure a WebLogic Server Domain" in the *WebLogic Server Command Reference* at the following URL: `http://e-docs.bea.com/wls/docs81/admin_ref/ant_tasks.html`
	Deployment Tools	Deploy applications and standalone modules to WebLogic Server. For more information, see "Deployment Tools Reference" in *Deploying WebLogic Server Applications* at the following URL: `http://e-docs.bea.com/wls/docs81/deployment/tools.html`
	WebLogic Integration Bulk Loader	Import, export, and delete trading partner data, including trading partner profiles, certificates from keystores, service definitions, and service profiles. For more information, see "Using the Trading Partner Bulk Loader" in *Managing WebLogic Integration Solutions* at the following URL: `http://e-docs.bea.com/wli/docs81/manage/bulkloader.html`

Moving Domains from Development to Production Environments

You can launch your domain in either a development environment or a production environment.

Use this startup environment...	When...
Development	You are creating your applications. At this stage, security is relatively relaxed, so you can auto-deploy applications.
Production	Your application is running in its final form. At this stage, security is fully configured and advanced features, such as clusters, may be used.

For a complete description of the differences between development and production environments, see "Differences Between Configuration Startup Modes" in *Creating WebLogic Configurations Using the Configuration Wizard* at the following URL:

`http://e-docs.bea.com/platform/docs81/confgwiz/newdom.html#devprod`

To move domains from a development environment to a production environment, you must complete several steps. For complete details, see:

- "Transitioning Domains from Development to Production Environments: Main Steps" in "Creating, Configuring, and Monitoring Servers," which is available in the *WebLogic Server Administration Console Online Help* at the following URL:

  `http://e-docs.bea.com/wls/docs81/ConsoleHelp/servers.html`

- "Moving to Production Mode" in *Developing Applications* at the following URL:

  `http://e-docs.bea.com/wls/docs81/developing.html`

When you are ready to move your domain to a production environment, keep in mind the following deployment considerations:

- For both development and production environments, BEA recommends that you package and deploy your standalone Web applications, EJBs, and resource adapters as part of an Enterprise application. See "Overview of the Split Development Directory Environment" in "Creating WebLogic Server Applications" in *Developing WebLogic Server Applications* at the following URL:

  `http://e-docs.bea.com/wls/docs81/programming/environment.html`

- Redeployment is not recommended for use in a production environment because clients cannot access an application during redeployment.
- Auto-deployment is a method for quickly deploying an application to an Administration Server for evaluation or testing. Use of auto-deployment in a production environment or for deployment on Managed Servers is not recommended.
- To auto-deploy in a production environment, you must first put the exploded directory structure or archive file in the domain directory. When you deploy an application or standalone module using an exploded archive directory, the Administration Console allows you to edit selected deployment descriptors directly in a production environment. The changes you make are automatically persisted to the module's deployment descriptor files, and the changes are dynamically applied to all target servers that host the module. See "Dynamically Updating Descriptors for a Deployment Unit" in "Performing Common Deployment Tasks" in *Developing WebLogic Server Applications* at the following URL:

  `http://e-docs.bea.com/wls/docs81/deployment/scenarios.html`