BEA WebLogic Platform™

Security in WebLogic Platform 8.1

Version 8.1 Service Pack 6
Revised:June 2006

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

Introducing WebLogic Platform 8.1 Security

Managing WebLogic Platform Security

Using an External Store for User Information

About This Document

This document summarizes the basic tasks and tools used for providing security for a WebLogic Platform 8.1 environment.

It is organized as follows:

- Chapter 1, "Introducing WebLogic Platform 8.1 Security," gives an overview of the tasks involved in setting up security for a Platform-based application in which features of WebLogic Integration, WebLogic Portal, and WebLogic Server are used.
- Chapter 2, "Managing WebLogic Platform Security," discusses the high-level tasks associated with configuring WebLogic Platform security and securing a production environment.
- Chapter 3, "Using an External Store for User Information," discusses considerations for using an external store for user information, such as a third-party LDAP server.
- Chapter 4, "Using BEA WebLogic Enterprise Security," provides a high-level overview of BEA WebLogic Enterprise Security and explains how this product can be used with WebLogic Platform 8.1.
- Chapter 5, "Security Advisories and Online Support," provides links to the BEA Advisories & Notifications page and other online support resources.
- Chapter 6, "Where to Find More Information," provides links to key, security-specific topics in the documentation for WebLogic Server, WebLogic Integration, WebLogic Portal, and WebLogic Workshop.

What You Need to Know

The main audience for this document consists of new users who want to understand how WebLogic Platform security works and where to go for more information. It is assumed that readers are familiar with Web technologies and have a general understanding of Windows and UNIX systems.

Product Documentation on the dev2dev Web Site

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

`http://dev2dev.bea.com`

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

Related Information

Other WebLogic Platform documents that you may find helpful when learning about the WebLogic Platform software are:

- *Introducing BEA WebLogic Platform*
- *Introducing WebLogic Platform Administration*
- *Creating WebLogic Configurations Using the Configuration Wizard*

Contact Us!

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

In your e-mail message, please indicate which version of the WebLogic Platform documentation you are using.

If you have any questions about this version of BEA WebLogic Platform, or if you have problems installing and running BEA WebLogic Platform, contact BEA Customer Support at `http://support.bea.com`. You can also contact Customer Support by using the contact

information provided on the quick reference sheet titled "BEA Customer Support," which is included in the product package.

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

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

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.
`monospace text`	Indicates *user input*, as shown in the following examples: Filenames: `config.xml` Pathnames: *`BEAHOME`*`/config/examples` Commands: `java -Dbea.home=BEA_HOME` Code: `public TextMsg createTextMsg(`
	Indicates *computer output*, such as error messages, as shown in the following example: `Exception occurred during event dispatching:java.lang.ArrayIndexOutOfBoundsException: No such child: 0`
`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`

Convention	Item
monospace italic text	Identifies variables in code. *Example*: `String expr`
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: `java utils.MulticastTest -n name [-p portnumber]`
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed. *Example*: `java weblogic.deploy [list|deploy|update]`
. . .	Indicates one of the following in a command line: • An argument can be repeated several times in a command line • The statement omits additional optional arguments • You can enter additional parameters, values, or other information
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER 1

Introducing WebLogic Platform 8.1 Security

WebLogic Platform 8.1 provides a single, unified security framework for all deployments, including portal applications, integration applications, and J2EE applications running on WebLogic Server. All WebLogic Platform components use the WebLogic Server LDAP-based security model introduced in 7.0, and do not require Compatibility Security. In addition, third-party security providers that you may add apply, uniformly, to all components of the WebLogic Platform product.

The following sections provide a high-level overview of WebLogic Platform 8.1 security:

Overview of WebLogic Platform Security

You can use the WebLogic Platform security model to:

- Program your applications with end-to-end security
- Add custom security extensions to provide tighter security policies
- Protect specific resources within your WebLogic Platform-based applications, such as WebLogic Portal and WebLogic Integration resources, by specifying security policies for those resources
- Map company business rules to security policies in distributed deployments, providing easy customization of application security to business requirements

- Easily update security policies, with no downtime
- Run more than one security provider at a time, as part of a transition scheme or migration path
- Use standard J2EE security technologies and the Java Authentication and Authorization Service (JAAS)

Security for all components of the WebLogic Platform product are based on the WebLogic Security Service. For a complete overview of this service, see *Introduction to WebLogic Security*. We recommend that you read that document in addition to the sections that follow.

Securing a Platform-Based Application

This section introduces the following basic concepts related to securing a Platform-based application:

- The WebLogic Platform Domain
- Protecting Resources in a Platform-Based Application
- Using Web Service Security (WS-Security)
- Obtaining End-to-End Security

Wherever it is applicable, these sections show examples from the End-to-End WebLogic Platform Tour, to which we refer, in this document, simply as the *Tour*. The Tour includes basic security settings that demonstrate how resources in a Platform-based application can be protected. The sections that follow take a closer look at those security settings, showing how WebLogic Platform security works in such an application. These sections also explain how security settings can be extended—for example, in a production environment—to take advantage of additional features of WebLogic Platform security, resulting in a comprehensively secure platform application environment.

For complete details about the Tour, see the *WebLogic Platform Tour Guide*.

The WebLogic Platform Domain

When you perform a complete installation of WebLogic Platform, all WebLogic Platform components and sample application files are installed on your system in a set of WebLogic Server *domains*. A domain is the basic unit of administration for WebLogic Server: it consists of one or more WebLogic Server instances and logically related resources and services that are managed, collectively, as one unit.

After installation, you typically create a new domain using the Configuration Wizard. Administrators can define security settings that apply either domain-wide, or to a particular Platform component. In general, domain-wide security resources are shared by the applications running within a domain.

An example of the platform domain is the Tour domain, called `end2end`. You can locate the files for this domain in the following directory:

`BEAHOME``/weblogic81/samples/domains/end2end`

(The Tour application itself is in the *`BEAHOME`*`/weblogic81/samples/platform/end2end` directory.)

The Tour domain requires that the WebLogic Server, WebLogic Portal, WebLogic Integration, and WebLogic Workshop components be configured in that domain to run the Tour. You can refer to the Tour domain for examples of the following:

- Applying security to application resources
- Defining user information
- Creating security roles
- Using default security providers

For more information about WebLogic Server domains, see "Overview of WebLogic Server Domains" in *Configuring and Managing WebLogic Server*.

Protecting Resources in a Platform-Based Application

The goal of configuring security for a Platform-based application is to protect the application's resources. For example, the Tour includes two portals that require protection: the employee portal and the manager portal. To keep these resources secure, an administrator must ensure that:

- Only authorized employees have access to the portals.
- Only managers have access to the managers portal.

When only managers have access to the managers portal, back-end resources that are accessible only through that portal are protected by default.

The following sections discuss how these resources are protected:

- Configuring Users and Groups in a Platform-Based Application
- Configuring Security Roles

- Configuring Security Policies
- Using Declarative Security

Configuring Users and Groups in a Platform-Based Application

Users and groups are classifications designed to control access to application resources. A *user* is an entity that can be authenticated. It may be a person or a software entity, such as a Java client. Each user is given a unique identity within a security realm. For more efficient security management, BEA recommends further classifying users in groups. A *group* is a collection of users who usually have something in common, such as the department of the company in which they work.

BEA recommends the use of groups for the sake of efficiency: when users are assigned to groups, you can manage a larger number of users at any given time. For example, the Tour is preconfigured with the following users and groups:

- Four users: `john`, `scott`, `rachel`, and `portaladmin`
- Two groups:
  - `employee` group, consisting of `john` and `scott`
  - `manager` group, consisting of `rachel`

Note: User `portaladmin` belongs to the `Administrators` and `PortalSystemAdministrators` groups, which are preconfigured for all WebLogic Platform domains.

How to Add, Modify, or Delete User Information

WebLogic Platform offers a choice of several methods for adding, modifying, or deleting user information:

- When you create a domain for your application using the Configuration Wizard, the Configuration Wizard prompts you for the username and password of the administrator who can start the server. For more information, see "Configuring an Administrative Username and Password" in "Configuring Security" in *Creating WebLogic Configurations Using the Configuration Wizard*.

  When you add users and groups at the time you create the application domain, you have the opportunity to establish an initial set of users before the application is started. This capability can be very useful when propagating an existing application and user information to new domains. For more information, see "Configuring Users and Groups"

in "Configuring Security" in *Creating WebLogic Configurations Using the Configuration Wizard*.

- When your application is running, you can use the administration consoles to add, change, or delete user information.

  You configure users and groups, roles, or policies—and modify or delete that information—anytime the application is running.

  Use the WebLogic Server Administration Console for adding basic user information and configuring security for WebLogic Server resources. But if you want to configure security for resources specific to WebLogic Portal or WebLogic Integration—for example, by adding trading partners or specifying access to portal resources—you should use the administration consoles for those components, as appropriate. The security information you provide using these consoles is maintained in the default WebLogic Server security store.

Whenever you add a user through any of the consoles, that user becomes visible in all of the other consoles. Likewise, when you remove a user from any of the administration consoles in WebLogic Platform, that user is no longer visible from any other console.

The exception to this cross-component access to user information involves user profiles, which provide additional data for certain types of users. User profiles are used with WebLogic Integration and WebLogic Portal. For more information, see "Managing User Profile Information" on page 3-9.

All user and group information is maintained in the default security store, an LDAP security directory. For a description of this store, see "Where User Information Is Stored" on page 3-2.

For more information, see "Users and Groups" in *Securing WebLogic Resources*, which explains how to provide security for resources and to create, add, modify, and delete users and groups

Configuring Security Roles

A security role is a privilege granted to users or groups based on specific conditions. For example, an administrator may define a security role called `AppAdmin`, which affords access to a particular servlet. Any user or group granted the `AppAdmin` security role has access to that servlet. Multiple users or groups can be granted a single security role. Like groups, security roles allow you to restrict access to WebLogic Platform resources for several users at once. However, unlike groups, security roles:

- Are computed and granted to users or groups dynamically, based on conditions such as user name, group membership, or the time of day.

- Can be scoped to specific WebLogic resources within a single application in a WebLogic Server domain (unlike groups, which are always scoped to an entire WebLogic Server domain).

Granting a security role to a user or a group confers the defined access privileges to that user or group, as long as the user or group is in the security role. Multiple users or groups can be granted a single security role. A given role can be used to protect access to zero or more resources.

For example, the portal application of the Tour, `e2ePortal`, is configured with the following roles:

- `employee`
- `manager`

In the Tour, these roles are used to protect portal access so that only employees can access the Employee portal, and only managers can access the Manager portal. For more information about how this is done in the Tour, do the following:

1. Start WebLogic Workshop.
2. Open the file `e2ePortal.work` in the following directory:

   BEAHOME`/weblogic81/samples/platform/end2end/e2ePortal`
3. Display the `Controller.jpf` file in the `e2ePortalProject` folder.

The `Controller.jpf` file defines the page flow for the various JSPs in the Portal application. For more information about this Portal application, see "Reviewing the Avitek Intranet Portal" in "Logging In to the Avitek Corporate Intranet" in *WebLogic Platform Tour Guide*.

For more information about how to configure security roles, see the following:

- "Security Roles" in *Securing WebLogic Resources*
- "Delegated Administration Overview" in the *WebLogic Administration Portal Online Help*
- "User Management" in *Managing WebLogic Integration Solutions*

Configuring Security Policies

Access to a resource is always performed based on a *security policy*. A security policy is created when you define an association between a WebLogic resource and one or more users, groups, or security roles.

The WebLogic Platform Tour is configured with security policies that determine who has access to the employee and manager portals in the `e2ePortal` Web application. These security policies,

specified declaratively in the `web.xml` deployment descriptor file, ensure that the `e2ePortal` resources are protected as follows:

- Only users in the `employee` or `manager` role can access the login portal.
- Only users in the `employee` role can access the Employee portal.
- Only users in the `manager` role can access the Manager portal.

In the Tour, the `web.xml` file is located on the `WEB-INF` folder of the `e2ePortalProject` folder. You can also find more information about the Tour's `web.xml` file in "Configuring Security in Web Applications" in "Logging In to the Avitek Corporate Intranet" in the *WebLogic Platform Tour Guide*.

You can define security policies to protect access to particular application resources. These security policies are called *scoped security policies*. By default, resources are preconfigured with a security policy that grants access to everyone. You define scoped security policies via the administration consoles provided with WebLogic Platform. For example, to restrict access to a particular portlet, use the WebLogic Administration Portal.

For information about how to configure security policies, see the following topics:

- For information about configuring security policies on WebLogic Server resources, see "Security Policies" in *Securing WebLogic Resources*.
- For information about configuring security policies on portal resources, see "Delegated Administration Overview" and "Overview of Visitor Entitlements" in the *WebLogic Administration Portal Online Help*.
- For information about configuring security policies on WebLogic Integration resources, including business processes, see the following topics in *Managing WebLogic Integration Solutions*:
  - "User Management" for information about defining new users, groups, and roles
  - "Process Security Policies" in "Process Configuration" for information about restricting access to the different resources of a WebLogic Integration process
  - "Setting Channel Security Policies" in "Message Broker" for information about restricting access to resources used by the Message Broker

Using Declarative Security

The J2EE specification defines standard, portable deployment descriptors for J2EE application components, such as Web applications, EJBs, and Web services. You can define security

information declaratively in the deployment descriptors for those entities. The deployment descriptors map the application's logical security requirements to its run-time definitions. At run time, the application container uses the security definitions to enforce the requirements. An example of this declarative security is provided in the `web.xml` file of the `e2ePortal` application in the Tour (for more information, see "Configuring Security in Web Applications" in "Logging In to the Avitek Corporate Intranet" in the *WebLogic Platform Tour Guide*).

If you use WebLogic Workshop, you can also declaratively specify security constraints via annotations for Web applications, Web services, controls, business process, and EJBs. When WebLogic Workshop builds the application, it automatically generates the security information required in the corresponding J2EE security descriptors for that application.

For complete details about specifying security in deployment descriptors, see the following:

- For Web application security, see "Developing Secure Web Applications" in "Securing Web Applications" in *Programming WebLogic Security*.
- For EJB security, see "Using Declarative Security with EJBs" in "Securing Enterprise JavaBeans (EJBs)" in *Programming WebLogic Security*.
- For security of Web services created in WebLogic Workshop, see "Web Service Security (WS-Security)" in the *WebLogic Workshop Help*.
- For security of Web services created in the WebLogic Server J2EE programming environment, see "Configuring Security" in *Programming WebLogic Web Services*.
- For security of business processes created in WebLogic Workshop, see the "@common:security Annotation" in the *WebLogic Workshop Help*.

Using Web Service Security (WS-Security)

WebLogic Workshop provides message-level security for Web services through an implementation of the WS-Security OASIS Web service security standard. The WebLogic Workshop implementation of WS-Security lets you secure the Simple Object Access Protocol (SOAP) messages passed between Web services using:

- Security tokens
- Digital signatures
- Encryption

Programming Web services security in WebLogic Workshop primarily involves writing code that does the following:

- Authorizes clients
- Sends credentials with outgoing messages
- Associates WS-Security policies to Web services and Web services controls

The declarative security that you program in WebLogic Workshop uses the same authentication and authorization mechanisms provided for all of WebLogic Platform. The security programming mechanisms available in Workshop merely provide a way to specify policies in deployment descriptors for the run-time code that is generated by the Workshop framework.

For information about and examples of using WS-Security when implementing Web services in the Workshop IDE, see "Web Services Security (WS-Security)" in the *WebLogic Workshop Help*.

You can also implement WS-Security in Web services that you create in the WebLogic Server programming environment. For more information, see "Overview of Web Services Security" in "Configuring Security" in *Programming WebLogic Web Services*.

Note: BEA's current implementation of the Web Services Security Core Specification is based on Working Draft Version 1.0, dated April 5, 2002. Because this specification is not yet an OASIS Standard, BEA's implementation is subject to change in future versions of WebLogic Server. Customers using the message-level security features described in "Overview of Web Services Security" in "Configuring Security" in *Programming WebLogic Web Services* should do so with the understanding that this implementation may not be compatible with Web Services Security implementations based on other versions of the Core Specification.

Obtaining End-to-End Security

WebLogic Platform provides multiple technologies to allow you to define end-to-end security for your applications. These technologies involve:

- Authentication of users or participants in the application
- Authorization to protected resources
- Control over the definition of confidential communication.

The Tour shows an example of how the WebLogic Security Service works to provide end-to-end security for an application that spans multiple components of WebLogic Platform. Consider the scenario in which an employee, `john`, logs in to the Avitek Employee portal to do the following:

1. Verify that the Human Resources department has correctly entered his personal data

2. Submit a requisition order for a new laptop.

The following steps enable you to take a closer look at how the WebLogic Security Service protects all the application resources involved in servicing the two requests issued by `john`:

1. First, before any employees log in to the Avitek Employee portal, the portal application itself needs to be started. WebLogic Server always authenticates who can start a server. You can control who can start the server on which you deploy your applications by requiring that the server prompt for a username and password.

   You can also preconfigure this authorized user when you create a new domain via the Configuration Wizard. Because the Tour runs in "development mode," the startup scripts are preconfigured to start the server on behalf of the administrator username `weblogic`.

2. After the Tour has been started, the `e2ePortal` application is launched. This application provides protection to its resources by requiring users to log in.

   The `e2ePortal` uses FORM authentication, which is a style of authentication provided by the WebLogic Server servlet container to validate the username and password against the data stored in the WebLogic Server Authentication provider (in this case, the LDAP directory that is embedded in WebLogic Server). The definition of the FORM authentication is specified within the `<login-config>` section of the `web.xml` deployment descriptor for the `e2ePortal` application. A valid authentication associates the current thread with the Subject information of the authenticated users, and redirects the control to the `begin` action of the `Controller.jpf` file in the `e2ePortal` page flow. This action extracts the role from the Subject and, depending on this role, redirects the flow to either the `Employee.portal` or to the `Manager.portal` portals.

3. After `john` has been authenticated, the page flow controller, `Controller.jpf`, of the employee portal (found, in Workshop, under the `employee` directory of `e2ePortalProject`) uses the Subject to extract the appropriate username and retrieve information stored in the database via a database control. This information is then displayed in the corresponding portlet.

4. When `john` enters an order requisition for a new laptop, an XML form is created and sent to the Order Requisition business process via the `WorkflowInvoker` Web service control. (This control is found, in Workshop, in the `WorkflowInvoker` folder of the `e2ePortal` directory.) This `WorkflowInvoker` control uses the standard Web service-based invocation mechanism to start the Order Requisition process. Note that because JMS currently does not automatically propagate the credentials of the invoker to the receiver, and because HTTP-based SOAP requests cannot propagate the identity automatically, you should design the appropriate mechanisms to pass these credentials to the receiver, if needed (for example, by adding the credential to the XML message).

5. The Order Requisition process, defined in the `e2eWorkflow` application, receives the XML order and processes it via a sequence of steps. Two of these steps are:

   a. Assigning an approval task to `john`'s manager, `rachel`

      The task is generated via a task control that is protected so that only members of the `Administrators` group can generate tasks. The Tour uses `run as` to switch the identity (by default, anonymous) of the JMS receiver (the Order Requisition process) to the `installAdministrator` user, who is a member of the `Administrators` group. The task is directed to the user who is authorized to respond to the task: `rachel`. This validation is performed when `rachel` logs in via the Manager portal.

   b. Accessing the inventory system to verify that the laptop is in stock

      Access to the inventory system is implemented via a WebLogic Integration Application View that uses a J2EE JCA adapter to access a database containing the inventory. The database is accessed on behalf of the user `weblogic`, and the mapping from the `InstallAdministrator` user is automatically done to the user `weblogic` (note that both users are members of the `Administrators` group) by the WebLogic Server J2EE JCA mechanisms.

As you can see, security for your application must be designed and planned from the outset. WebLogic Platform provides multiple tools for configuring security, depending on the components that you want to use. You should become familiarized with these tools before you design the security of your application. For a set of useful links to the more detailed documentation provided by the different WebLogic Platform components, see Chapter 6, "Where to Find More Information," in this document.

CHAPTER 2

Managing WebLogic Platform Security

The following sections provide a high-level view of the tasks associated with managing WebLogic Platform security:

- Configuring WebLogic Platform Security
- Securing a Production Environment

Configuring WebLogic Platform Security

This section discusses the following topics:

- Upgrading the Security Configuration from a Previous Release
- Configuring Security Using the Configuration Wizard
- Configuring Security Using the Administration Consoles
- Configuring Your Platform Applications on Multiple Servers, Across Domains, and in Clusters
- Customizing the Security Configuration
- Backing Up User Information
- Migrating User Information to Another Security Realm or Domain

Upgrading the Security Configuration from a Previous Release

This section summarizes key points about upgrading your security configuration from previous releases of WebLogic Platform, and provides links to security upgrade topics throughout the WebLogic Platform documentation. For a comprehensive list of all upgrade topics in the WebLogic Platform documentation, see the *WebLogic Platform Upgrade Planning Guide*.

Note: Compatibility Security is not supported for WebLogic Platform. If you use only WebLogic Server in a domain, you can use Compatibility Security. However, if you add a WebLogic Portal or WebLogic Integration component to that domain, then Compatibility Security is not supported.

Upgrading to WebLogic Server 8.1 Security

If you are upgrading WebLogic Server 7.0 to 8.1, note the following changes regarding the security system:

- All security configuration data is now stored in the `config.xml` file. Existing security configuration data is written to the `config.xml` file when the WebLogic Server 8.1 server is initially booted.
- A new keystore implementation is available in WebLogic Server 8.1. The keystore retrieves trusted certificate authorities, private keys, and server certificates from JDK keystores. Any JDK-supported keystore provider can be used WebLogic Server 8.1; however, the custom WebLogic Keystore Provider introduced in 7.0 is deprecated.

For more information about upgrading to WebLogic Server 8.1 security, see "Security" in "Upgrading WebLogic Server 7.0 to Version 8.1" in *WebLogic Server 8.1 Upgrade Guide*.

Upgrading to WebLogic Integration 8.1 Security

The security model used by WebLogic Integration has changed significantly in WebLogic Platform 8.1. In version 7.0, WebLogic Integration used Compatibility Security, which is no longer supported in WebLogic Platform. In version 8.1, WebLogic Integration uses the new security model introduced in WebLogic Server 7.0.

See the *WebLogic Integration Upgrade Guide* for the following topics related to upgrading the security configuration from previous releases of WebLogic Integration:

- Business Process Management: migrating users and roles, organizations, calendars, email, and permissions

- B2B: migrating the keystore, trading partner security information, and the packaging of some Java classes you may be using
- Application Integration: upgrading EIS authentication and authorization and Application View Access Control
- Security API changes
- Mapping ACL information to security policies

Upgrading to WebLogic Portal 8.1 Security

The security model used by WebLogic Portal has changed significantly in WebLogic Platform 8.1. In version 7.0, WebLogic Portal used Compatibility Security, which is no longer supported in WebLogic Platform. In version 8.1, WebLogic Portal uses the new security model introduced in WebLogic Server 7.0.

See the *WebLogic Portal Upgrade Guide* for the following notes related to upgrading the security configuration from previous releases of WebLogic Portal:

- Existing WebLogic Portal 7.0 users will be able to continue to leverage their RDBMS user stores by leveraging an SSPI Authentication Provider that will be provided for connecting to a RDBMS.
- Entitlement segments on 7.0 cannot be upgraded because of the change to leveraging the WebLogic Server SSPI. Customers will need to create roles using the WebLogic Administration Portal for entitlements.
- The Unified User Profile is unchanged for 8.1.

Upgrading to WebLogic Workshop 8.1 Security

If you are upgrading from release 7.0 to 8.1, there are no security-related upgrade requirements. However, there are a number of new security tools and capabilities available, including the following:

- The annotation `@common:security`, which automatically writes role restrictions into the EJB configuration files, eliminating a tedious manual process. This annotation can be used in all Java-derived files (Web services, business process definitions, extensible Java controls, and so on), except Java page flows.

  For more information, see "An Overview of Role-Based Security" in the *WebLogic Workshop Help*.

- The Security Roles folder, which provides a convenient way to add test users and application-scoped roles for testing your security designs.

  For more information, see "Creating Principals and Role-Principal Mappings" in the *WebLogic Workshop Help*.

- A new Web Services Security implementation, described in "Web Service Security (WS-Security)" in the *WebLogic Workshop Help*.

Configuring Security Using the Configuration Wizard

When you create a new domain using the Configuration Wizard, you are prompted to specify an administrative username and password for the domain. You can also specify additional security for application resources by defining users, groups, and roles for that domain. For more information about configuring security using the Configuration Wizard, see "Configuring Security" in *Creating WebLogic Configurations Using the Configuration Wizard*.

Configuring Security Administratively Versus Programmatically

One of the most powerful features of WebLogic Platform Security is the flexibility it provides for supporting robustly secure deployments. One important choice you must make when planning your security configuration is whether to implement it administratively, programmatically, or as a combination of the two.

Administrators typically use the administration console to provide security for various resources. For example, instead of requiring developers to modify multiple deployment descriptors when organizational security requirements change, administrators can modify all security configurations from a centralized, graphical user interface. Users, groups, security roles, and security policies can all be defined, modified, and managed using an administration console. As a result, the process of making changes based on updated security requirements becomes more efficient.

Using this technique, you can configure security for all types of WebLogic Platform resources. Therefore, the recommendations in this document for providing security for WebLogic Platform resources are intended specifically for users of the administration consoles.

Developers typically define and enforce roles programmatically. Roles are associated with the deployment descriptor. Later, administrators can always add roles and security policies. The primary benefit of specifying the security for resources such as EJBs and Web applications in their deployment descriptors is that the technique is a widely known and standard method for

adding declarative security to those types of applications. This technique may also be the one with which most organizations are familiar. When this technique is used, security roles and policies are specified in the `web.xml`, `weblogic.xml`, `ejb-jar.xml`, and `weblogic-ejb-jar.xml` deployment descriptors.

When choosing the administrative model that works best for you, consider who is responsible for managing security in your organization. Developers are most familiar with the application components they build, but they might not necessarily be familiar with the deployment environment in which those components run. In addition, as security policies change, it is more economical to reconfigure security administratively instead of rebuilding, retesting, and redeploying applications.

Configuring Security Using the Administration Consoles

The security settings you configure in the WebLogic Server Administration Console are propagated by default to all the platform components. For example, the users and groups you add in the WebLogic Server Administration Console are users of WebLogic Integration, WebLogic Portal, and WebLogic Workshop applications and resources by default. Conversely, as you add users in the WebLogic Integration or WebLogic Portal administration consoles, those users become WebLogic Server users by default.

The purpose of the administration consoles for WebLogic Integration and WebLogic Portal, however, is to configure and manage entities that are specific to those components. The security that you configure for WebLogic Portal and WebLogic Integration resources is visible only from the administration consoles for those components.

Table 2-1 lists the resources you can secure via the administration consoles available in WebLogic Platform.

Table 2-1 Consoles for Configuring Security

Console	Resource Types
WebLogic Server	• Users, groups, roles (global and scoped), and security policies. • Security providers, such as those that provide authentication, authorization, adjudication, role mapping, credential mapping, and auditing services. • Security for WebLogic Server resources, such as Administrative resources, Application resources, EJB resources, URL resources, JDBC resources, JMS resources, and so on. • The embedded LDAP server (which serves as the registry, or store, for the authentication and authorization information needed by your application) For a complete list, see "Types of WebLogic Resources" in *Securing WebLogic Resources*.
WebLogic Integration	• Users, groups, and roles • Business processes and business process operations • Trading partners, including certificates and transports • Message broker channels • Process security policies • Application views, for application integration For a complete list, see "Using WebLogic Integration Security" in *Deploying WebLogic Integration Solutions*.
WebLogic Portal	• Users, groups, and roles • Delegated administration • Visitor entitlements • Portals, desktops, shells, books, pages, layouts, look & feels, and portlets • Web applications, JSPs, EJBs, and other J2EE resources • Content providers • Campaigns For a complete list, see "Securing Portal Applications" in the *WebLogic Workshop Help*.

Note: You can use any of the administration consoles in WebLogic Platform to define security policies for an application that has been developed in WebLogic Workshop. However, if you redeploy your application from Weblogic Workshop in a development mode server, those security policies will be reset to the policies specified in the deployment descriptors for that application. This does not happen for a production mode server.

Configuring Your Platform Applications on Multiple Servers, Across Domains, and in Clusters

You may want to configure the applications based on the WebLogic Platform components on different servers within a domain. For example, you might want to manage your WebLogic Portal application independently from your WebLogic Integration, Workshop, and EJB applications. By configuring your WebLogic Portal application in its own server, you can adjust quickly to the needs of your customers and still communicate with your applications.

If your application spans multiple domains, you need to set up a trust relationship among those domains. In a trust relationship, principals from one domain will be accepted as principals in another. The trust relationship is established when the `Credential` attribute of the `SecurityConfigurationMBean` (see the `config.xml` file) in one domain matches the `Credential` attribute on the `SecurityConfigurationMBean` in the other domain. If you boot an administration server for the first time, and the `Credential` attribute is not set, the administration server notices that this attribute is not set and generates a random credential that is then used to sign the principals created in that domain. Other servers in the domain retrieve the credential from the administration server and, therefore, will be able to establish the trust relationship within the domain. (Note that setting up a trust relationship is required only when multiple domains are involved; it is not necessary for multiple servers operating within the same domain.) For more information, see "Enabling Trust Between WebLogic Server Domains" in "Configuring Security for a WebLogic Domain" in *Managing WebLogic Security*.

If you want to configure your WebLogic Platform applications in a clustered environment, note that all the servers in a given cluster need to be identical. Therefore, if you want to deploy the servers for one WebLogic Platform component separately from the servers for other WebLogic Platform components, you would target one cluster for one WebLogic Platform component, and another cluster for a different component. For example, you could deploy a WebLogic Integration application in one cluster, and a WebLogic Portal application in another cluster. However, in a clustered domain, the security settings apply uniformly to the entire domain. The difference among the clusters is in how you deploy your application. For more information, see *Using WebLogic Server Clusters*.

Customizing the Security Configuration

WebLogic Platform gives you a great deal of flexibility in how you can customize your default security configuration. Some of the ways in which you can customize your security configuration include:

- Replacing the WebLogic security providers with your custom security providers, for example, the authentication provider, authorization provider, or auditing provider
- Changing the default attribute settings for a WebLogic Security provider
- Configuring additional security providers in the default security realm. (For example, if you want to use two Authentication providers—one that uses the embedded LDAP server and one that uses an existing store of users and groups—you would configure them in the default security realm.)

For information, see "Customizing the Default Security Configuration" and "Configuring Security Providers" in *Managing WebLogic Security*.

Backing Up User Information

BEA strongly recommends that you maintain an archive of your user information in the event of a machine failure or other problem. The following sections describe backing up user information:

- Backing Up User Information in the Embedded LDAP Server
- Backing Up Digital Certificates
- Backing Up Trading Partner Profiles
- Additional Resources on dev2dev

Backing Up User Information in the Embedded LDAP Server

If you store user information in the embedded LDAP server, WebLogic Platform provides built-in tools that can do the following to either an archive or to another embedded LDAP server:

- Export user information
- Back up the entire security content of an embedded LDAP server

For information about using these tools, refer to the following:

- "Managing the Embedded LDAP Server" in *Managing WebLogic Security*

- "Backing Up Security Data" in *Configuring and Managing WebLogic Server*

If you are using an external LDAP server, you must use the tools available with that server for backing up stored information about users. You can also use the tools available with your RDBMS to back up the system and user profiles stored by WebLogic Platform.

Backing Up Digital Certificates

Digital certificates are electronic documents used to identify principals and objects as unique entities over networks such as the Internet. A digital certificate securely binds the identity of a user or object, as verified by a trusted third party known as a certificate authority, to a particular public key. The combination of the public key and the private key provides a unique identity for the owner of the digital certificate.

Once you have obtained private keys, digital certificates, and trusted CA certificates, you need to store them so that WebLogic Server can use them to find and verify identity. Private keys, their associated digital certificates, and trusted CA certificates are stored in keystores. The keystores can be configured through the WebLogic Server Administration Console or specified on the command-line. Use the Keystore Configuration section of the Keystores and SSL page of the WebLogic Server Administration Console to configure Identity and Trust keystores for WebLogic Server.

Note: WebLogic Platform does not provide tools for archiving or restoring digital certificates, so you should make sure you keep backup copies of certificates in the event of a system failure.

Backing Up Trading Partner Profiles

WebLogic Integration provides a utility that allows you to export and import trading partner profiles, as described in the following topics in "Trading Partner Management" in *Managing WebLogic Integration Solutions*:

- "Exporting Management Data"
- "Importing Management Data"

Additional Resources on dev2dev

Additional resources for use with WebLogic Platform 8.1 can be found at the BEA dev2dev Web site, available at the following URL:

`http://dev2dev.bea.com/products/wlplatform81/index.jsp`

Migrating User Information to Another Security Realm or Domain

The WebLogic Security Service supports the ability to export users, groups, and security roles from one security realm, and import them into another—into the same or a new domain. You can migrate security data associated one security provider individually, or migrate all the security data associated with each security provider at once (that is, security data for an entire security realm). You migrate security data through the WebLogic Server Administration Console or by using the `weblogic.Admin` utility.

Migrating security data may be helpful when you must:

- Transition from development mode to production mode
- Move production-mode security configurations to new WebLogic Server domains
- Move data from one security realm to another, within a single WebLogic Server domain, and one or more of the WebLogic Platform security providers is replaced by custom or third-party security providers

If you are migrating security data to another security realm, in either the same or a different domain, note the following:

- The source and target realms must be alike with respect to the type of data store and security providers used. For example, if the source security realm uses the default Authentication provider, the target realm must as well.
- If your security realm uses a custom security provider or LDAP store, you must use migration tools specific to the provider or LDAP store. The built-in WebLogic Server migration tools for security information are supported for only the default set of security providers and the embedded LDAP server.

For more information, see "Migrating Security Data" in *Managing WebLogic Security*. For information about using the `weblogic.Admin` utility to migrate security information to another security realm, see "Using the weblogic.Admin Utility" in "Migrating Security Data" in *Managing WebLogic Security*.

Securing a Production Environment

This section discusses considerations for securing the production environment in which WebLogic Platform applications can run. It also provides links to helpful documents about

security in a production environment in the documentation for WebLogic Server, WebLogic Integration, WebLogic Portal, and WebLogic Workshop.

Securing the Production Environment

The WebLogic Server document, *Securing a Production Environment*, contains several basic tips for providing security in a WebLogic Platform production environment. Specifically, the document offers both general and detailed advice about securing the following key resources:

- WebLogic Server host
- Network connections
- Databases
- WebLogic Security Service
- Applications

Securing WebLogic Platform Resources

In addition to making your basic application environment secure, we recommend that you provide security for resources that are specific to WebLogic Platform, such as the following:

- Database tables and pools
- Configuration files on disk
- Keystores
- Security realms (embedded LDAP store)
- Thread pools
- J2EE connector resources from WebLogic Integration
- JNDI resources for platform components
- EJBs and Web applications generated through WebLogic Integration

For information about securing these and other WebLogic Platform resources in a production environment, see the following documents.

Table 2-2 Securing WebLogic Platform Resources in a Production Environment

For information about how to secure . . .	See the following documents . . .
WebLogic Server resources	*Securing WebLogic Resources*
WebLogic Integration resources	"Setting Up a Secure Deployment" in "Using WebLogic Integration Security" in *Deploying WebLogic Integration Solutions*
WebLogic Portal resources	"Securing Portal Applications" and "Implementing Authentication" in the *WebLogic Workshop Help*. See also "Preparing and Deploying the EAR File" in the WebLogic Portal *Production Operations User Guide*.
Web services created in WebLogic Workshop	"Web Service Security (WS-Security)" in the *WebLogic Workshop Help*

Configuring a Domain for a Production Environment

You should not deploy applications on the Administration Server or expose that server directly to the Internet. When setting up a production system, BEA recommends that you deploy your applications on a domain that is configured with an Administration Server and one or more Managed Servers. This type of domain configuration is recommended for production environments that require the improved availability and performance provided by the built-in failover and load balancing features in a WebLogic Server cluster.

Deploying your applications on Managed Servers gives the added security benefit of insulating the application and its users from the production environment infrastructure. It is also important to avoid making the Administration Server, or any components deployed on it, available outside the enterprise.

Note: The default setting of an option defined in the WebLogic Server Administration Console presents a potential security risk. The option is the "Anonymous Admin Lookup Enabled" option, and by default it is selected. To protect your application, we recommend that you deselect this option when you enter production mode. To find this option in the console, select Domain Wide Security Settings, Configuration and then select the General tab. Remove the checkmark from the box beside "Anonymous Admin Lookup Enabled." Keep in mind, however, that applications that perform anonymous lookup of the administration MBeanHome interface, intentionally or otherwise, might not work correctly when this feature is disabled.

For more information, see the following topics:

- "Deployment Targets" in "Overview of WebLogic Server Deployment" *Deploying WebLogic Server Applications*
- "The Administration Server and Managed Servers" in "Overview of WebLogic Server System Administration" *Configuring and Managing WebLogic Server*
- "Example: Creating a Domain with Administration Server and Clustered Managed Servers" in "Creating and Configuring Domains Using the Configuration Wizard" in *Configuring and Managing WebLogic Server*

Enabling Security Auditing

Auditing is the process whereby information about operating requests and the outcomes of those requests is collected, stored, and distributed in order to prevent repudiation. In other words, auditing provides an electronic trail of computer activity. In the WebLogic Server security architecture, an Auditing provider is used to provide auditing services.

If configured, the WebLogic Security Framework calls an Auditing provider before and after the performance of security operations (such as authentication or authorization). The decision to audit a particular event is made by the Auditing provider itself and can be based on specific audit criteria and/or severity levels.

You can use either the WebLogic Auditing provider or a custom Auditing provider in a security realm. Although an Auditing provider is configured per security realm, each server writes auditing data to its own log file in the server directory. By default, all auditing information recorded by the WebLogic Auditing provider is saved in the following file:

`WL_HOME\yourdomain\yourserver\DefaultAuditRecorder.log`.

By writing a custom Auditing provider, however, you can send the records containing audit information to any one of various output repositories, such as an LDAP server, database, or a simple file.

For information about configuring an Auditing provider and enabling auditing, see "Configuring a WebLogic Auditing Provider" in "Configuring Security Providers" in *Managing WebLogic Security*.

CHAPTER 3

Using an External Store for User Information

As an administrator, one of your basic tasks is to create and manage information about the users of your deployment. This chapter gives an overview of the following topics:

- Basic concepts about the WebLogic Security Service related to user information that is useful to understand
- Tasks you need to perform to create and manage user information, and to customize how it is stored and used
- Back up and restore user information, on either the same or a different configuration
- Work with user information that is stored in an external LDAP server

The following topics are included:

Note: User information is defined to encompass users, groups, roles, and security policies. However, this chapter focuses specifically on users, groups, and roles. For information about security policies, see "Security Policies" in *Securing WebLogic Resources*.

Where User Information Is Stored

User information consists of the following:

- User names and passwords
- Groups
- Security roles
- Security policies

By default, whenever you create a domain, user information (with one exception) is stored in the embedded WebLogic LDAP server.

Note: The exception is user profiles. For details, see "Managing User Profile Information" on page 3-9.

User information stored in the embedded LDAP server is always available to every component of WebLogic Platform; it can also be managed, modified, and deleted by any component, regardless of which administration console was used for the initial creation of the user information.

After you modify user information in the administration console for one component, you can display the modified information in the administration console for another component, as long as display of the latter administration console has been refreshed.

Figure 3-1 shows the embedded LDAP server that is shared by all component administration consoles.

Figure 3-1 Embedded LDAP Server Stores WebLogic Platform Security Information Centrally

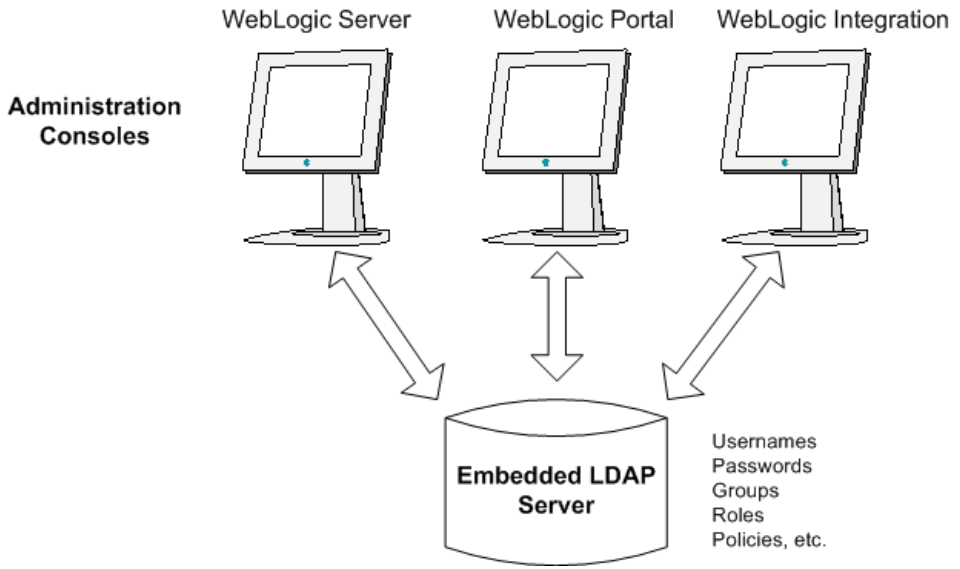

Table 3-1 summarizes the user information that you can create, modify, and remove in each administration console available in WebLogic Platform, and also lists the security provider and storage repository associated by default with each category of user information.

Table 3-1 also lists the default security provider and data store associated with each piece of user information. If later on you choose to customize the store used for user information, you need to customize the associated security provider. (Customizing the data store is described in "Using an External LDAP Server" on page 3-6.)

Table 3-1 User Information You Can Enter, Modify, and Remove in WebLogic Platform

User Information	Console Used to Create and Manage Information	Associated Security Provider	Default Storage Repository
Usernames and passwords	Administration console for any WebLogic Platform component	WebLogic Authentication provider	Embedded LDAP
Groups	Administration console for any WebLogic Platform component	WebLogic Authentication provider	Embedded LDAP

Table 3-1 User Information You Can Enter, Modify, and Remove in WebLogic Platform (Continued)

User Information	Console Used to Create and Manage Information	Associated Security Provider	Default Storage Repository
Security Roles	Administration console for any WebLogic Platform component	WebLogic Role Mapping provider	Embedded LDAP
Portal user profiles	WebLogic Administration Portal Console	N/A	RDBMS
Trading partner profiles	WebLogic Integration Administration Console	N/A	RDBMS

The sections that follow give additional information about the two security providers and the default storage repositories listed in Table 3-1.

Security Providers Associated with User Information

Table 3-1 lists the following security providers that coordinate security functions associated with users, passwords, groups, and roles:

- Authentication provider

  Authentication providers are used to prove the identity of users or system processes. Authentication providers also remember, transport, and make identity information available to various components of a system when needed.

- Role Mapping provider

  Role Mapping providers obtain a set of security roles granted to a requestor for a given resource at run time, and then supply Authorization providers with this role information so that the Authorization provider can determine if access is allowed to the requested WebLogic resource.

The WebLogic Security Service also provides additional security providers that work with user information, but these two providers are of special importance when backing up or migrating user information, or for customizing how user information is stored.

Default Repositories for User Information

As shown in Table 3-1, the following user information is maintained by default in the embedded LDAP server:

- User names
- Passwords
- Groups
- Security roles

User profiles are created in the RDBMS that is used for storing WebLogic Platform data in a given domain. By default, WebLogic Platform domains use the PointBase RDBMS. For more information about user profiles, see "Managing User Profile Information" on page 3-9.

Customizing the User Information Data Stores

Table 3-2 shows a sample configuration that uses Netscape iPlanet 4.1.3 LDAP server for storing user and group information, and Oracle 8.1.7 as an RDBMS that is used with WebLogic Platform and WebLogic Portal.

Table 3-2 Using External Data Stores for User Information

User Information	Console Used to Create and Manage Information	Associated Security Provider	External Storage Repository
Usernames and passwords	Netscape iPlanet administration server	Netscape iPlanet Authentication provider[1]	Netscape iPlanet 4.1.3 LDAP server
Groups	Netscape iPlanet administration server	Netscape iPlanet Authentication provider[1]	Netscape iPlanet 4.1.3 LDAP server
Security Roles	Administration console for any WebLogic Platform component	WebLogic Role Mapping provider	Embedded LDAP

Table 3-2 Using External Data Stores for User Information (Continued)

User Information	Console Used to Create and Manage Information	Associated Security Provider	External Storage Repository
Portal user profiles	WebLogic Administration Portal Console	N/A	Oracle 8.1.7
Trading partner profiles	WebLogic Integration Administration Console	N/A	Oracle 8.1.7

1. The WebLogic Platform distribution includes an out-of-the-box authentication provider that can be used with the Netscape iPlanet LDAP server.

Using an External LDAP Server

WebLogic Platform provides tools that make it easy to configure Authentication and Role Mapping providers to work with user information stored in an external LDAP server. This section provides high-level notes about using an external LDAP server and provides links to appropriate topics in the WebLogic Server documentation set that describe the tools and give the steps for migrating user information to and using an external LDAP server.

Note: When you use an external LDAP server, you can use that server to store user and group information. You use the console for that LDAP server to manage the user information stored on it. However, roles, policies, and additional security information remain stored in the embedded LDAP server and are managed from the WebLogic administration consoles.

LDAP Servers You Can Use with WebLogic Platform

WebLogic Platform can be used with any LDAP server that works with WebLogic Server. The following external LDAP servers have been tested with WebLogic Server:

- Netscape iPlanet version 4.1.3
- Active Directory shipped as part of Windows 2000
- Open LDAP version 2.0.7
- Novell NDS version 8.5.1

Note that if you do use an external LDAP server:

- You manage user names, passwords, and groups from the console for that LDAP server.
- From any WebLogic Platform administration console, you can view the user name, password, and group information stored in the external LDAP server, but the access is read-only if you use an out-of-the-box Authentication provider with that server. (You do have the option of creating a custom Authentication provider that has write-access to that information, however.)

For more information about external LDAP server support, see the following topics in *Managing WebLogic Security*:

- "Configuring an LDAP Authentication Provider" in "Configuring Security Providers"
- "Managing the Embedded LDAP Server"

Using a Custom or Third Party Authentication Provider

If you choose to store user information in an external LDAP server, you need to configure an Authentication provider that works with that server. WebLogic Platform includes out-of-the-box Authentication providers that have been tested with the LDAP servers listed in the preceding section.

You can also create a custom Authentication provider. Any Authentication provider compatible with the WebLogic Server Security SPI can interact with the users, groups, roles, entitlements, and resources configured for WebLogic Platform. In addition, any such Authentication provider is not restricted in the number of groups and users you configure.

Note: Currently, WebLogic Platform does not offer an authentication provider that can work with user information stored in an RDBMS.

Using Custom or Third-Party Authentication Providers with WebLogic Portal or WebLogic Integration

If you are using a custom or third-party authentication provider with WebLogic Portal or WebLogic Integration, note the following:

- WebLogic Portal and WebLogic Integration support multiple authentication providers, including all the standard providers supported by WebLogic Server.
- WebLogic Portal and WebLogic Integration support the management of users and groups from custom or third-party authentication providers. Note, however, the following limitation: Even though you can use custom or third-party authentication providers, such as

the out-of-the-box third-party authentication providers included with WebLogic Platform, only read access to those providers is available from the WebLogic Platform administration consoles by default. Therefore, the WebLogic Portal and WebLogic Integration administration consoles and tools do not support create, delete, and modify operations for user and group management in custom or third-party authentication providers by default. Although the default implementation of the out-of-the-box third-party providers included with WebLogic Platform is read-only, you can configure the provider's MBean to support full create, delete, and modify operations. (Full management access to users and groups is provided by WebLogic Server's embedded LDAP server.)

- New in WebLogic Platform 8.1 Service Pack 3, WebLogic Portal supports multiple authentication providers. You configure, deploy, and undeploy authentication providers via the WebLogic Server Administration Console; and you can use those authentication providers for the management of users, groups, personalization, Delegated Administration, and visitor entitlements directly from the WebLogic Administration Portal.

  Note the following regarding the use of multiple authentication providers in WebLogic Portal:

  – The WebLogic Administration Portal now includes new pages for building the User Group Resource tree for each configured authentication provider and viewing the read and write permissions for that provider.

  – The amount and granularity of information you can manage using a particular custom or third-party authentication provider depends on the set of optional security MBeans that are implemented in the provider.

  – The Workshop Portal Extensions have been enhanced to let you specify the authentication provider to use in any development task involving users and groups.

  Note that in Service Pack 3, the WebLogic Administration Portal has added the Authentication Hierarchy Service, which you can use for building in-memory group resource trees for each configured authentication provider. This new service gives portal administrators a visual representation of the groups associated with each authentication provider, providing a convenient visual mode for user and group management and increasing the access speed to the users and groups in those providers.

  For more information about managing multiple authentication providers in the WebLogic Administration Portal, see "Using Multiple Authentication Providers with WebLogic Portal."

  For more information about using multiple authentication providers in WebLogic Portal application development, see "Using Multiple Authentication Providers in Portal Development."

For More Information About Custom Authentication Providers

If you are using a custom or third-party authentication provider with WebLogic Server, see the following topics in *Developing Security Providers for WebLogic Server*:

- "Configuring the Custom Security Provider" in "Introduction to Developing Security Providers for WebLogic Server"
- "Configure the Custom Authentication Provider Using the Administration Console" in "How to Develop a Custom Authentication Provider" in "Authentication Providers"

If you want to configure an out-of-the-box authentication provider for an external LDAP server, see "Configuring an LDAP Authentication Provider" in "Configuring Security Providers" in *Managing WebLogic Security*.

Using a Custom Role Mapping Provider

The Role Mapping provider included with WebLogic Platform supports the deployment and undeployment of security roles within the system. This provider uses the same security policy engine as the WebLogic Authorization provider. If you want to use a role mapping mechanism that already exists within your organization, however, you can create a custom Role Mapping provider to tie into that system.

Consider an environment that uses a large project database that contains the information required to determine role information. Because the WebLogic Role Mapping provider works only with role information stored in the embedded LDAP server, a custom Role Mapping provider would need to be created that can work with this externally-stored role information.

In WebLogic Platform 8.1 Service Pack 3, the WebLogic Portal administration tools now support the federating of roles from custom or third-party Role Mapping providers. This enables you to create visitor entitlements that are based on roles that are maintained by Role Mapping providers that have been configured in addition to the WebLogic Role Mapping provider.

For information about how to create a custom Role Mapping provider, see "How to Develop a Custom Role Mapping Provider" in "Role Mapping Providers" in *Developing Security Providers for WebLogic Server*.

Managing User Profile Information

WebLogic Integration and WebLogic Portal also allow you to add profiles that can be associated with users. By default, user profiles are visible only on the administration console from which they are created. WebLogic Integration and WebLogic Portal store profiles in the WebLogic

Integration and WebLogic Portal repositories, respectively, which exist on the RDBMS that has been configured for the WebLogic Platform domain. (By default, this RDBMS is PointBase.)

For example, in WebLogic Integration you can create trading partner profiles. A trading partner profile consists of a user name, a password, and other data that is specific to B2B applications, such as the trading partner's address, business, and other relevant data. The user name associated with a trading partner profile is stored and maintained as a regular WebLogic user in the configured Authentication provider. However, the additional trading partner profile data is stored in the WebLogic Integration repository. WebLogic Integration has an internal mechanism that maps each trading partner profile in its repository with the corresponding user name.

WebLogic Portal also has the notion of a user profile. Like a trading partner profile, a WebLogic Portal user profile:

- Associates a user with the profile, and maintains the corresponding user name and password in the configured Authentication provider. The data in a user profile includes data that is collected about the associated user, such as portal preferences.
- Maintains the additional data in the user profile in the WebLogic Portal repository

By default, trading partner profile information can be viewed only from the WebLogic Integration Administration Console, and Portal user profiles can be viewed only from the WebLogic Administration Portal. However, the users associated with trading partner profiles and Portal user profiles can be viewed in any WebLogic administration console.

Note: Profiles of users created in the WebLogic Integration and WebLogic Portal administration consoles are stored on an RDBMS. Usernames and passwords, however, are maintained by the authentication provider configured for your application domain. They are stored in the LDAP server configured with that provider.

Removing User Profiles

Note the following about removing user profiles:

- When you remove a portal user from any of the WebLogic Platform administration consoles, the profile information associated with that user is no longer accessible from the WebLogic Administration Portal.
- Removing a trading partner from the WebLogic Integration is a two step process:

  a. Remove the trading partner from the WebLogic Integration Administration Console. This removes the corresponding user profile from the WebLogic Integration repository.

b. Remove the corresponding WebLogic Server user from the WebLogic Server Administration Console.

If you perform step a without performing step b, the WebLogic Server user remains in the environment. Likewise, simply removing the WebLogic Server user does not cause the corresponding user profile to be removed from the WebLogic Integration repository.

Users, Groups, and Roles Preconfigured in a Platform Domain

This section lists and describes the users, groups, and roles that are preconfigured when you create a platform domain using the Configuration Wizard. The user information listed in this section is provided for informational purposes only. You may find it useful for the purposes of tracking the users, groups, and security roles that need to be protected, backed up, deleted, or migrated, depending on your software environment and the preconfigured user information that you use.

Default Users Created in a Platform Domain

Table 3-3 lists and describes the users that are created by default in a platform domain.

Table 3-3 Default WebLogic Platform Users

User Name	Description
`weblogic`	Default username for the administrator of a domain. This user has system-administrator privileges. Note that `weblogic` is the default password for this username in the sample application domains provided out-of-the-box for all WebLogic Platform components.

Table 3-3 Default WebLogic Platform Users

User Name	Description
`portaladmin`	Default username for the Portal administrator. This user belongs to the `Administrators` and `PortalSystemAdministrators` groups. By default, the password for this user is `portaladmin`. If you are not using the Administration Portal, it is safe to remove this user from any WebLogic Platform administration console. **Note:** We strongly recommend that you change this password after you create your domain, especially if the domain is meant to be used in a production environment.
`yahooadmin`	Default username for the administrator for the My Yahoo! Enterprise Edition portlet. This user belongs to the `Administrators` group. The `yahooadmin` name activates support for anonymous users of the My Yahoo! Enterprise Edition portlets. By default, the password for this user is `yahooadmin`. If you are not using the Administration Portal, it is safe to remove this user from any WebLogic Platform administration console. **Note:** We strongly recommend that you change this password after you create your domain, especially if the domain is meant to be used in a production environment.

Default WebLogic Server Roles and Groups

Table 3-4 and Table 3-5 list and describe the default WebLogic Server roles and groups created in a platform domain.

Table 3-4 Default Roles in WebLogic Server

Role	Description
`Anonymous`	All users (the group everyone) are granted this global role.
`Admin`	Has the privilege to: • View the server configuration, *including* the encrypted value 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.

Table 3-4 Default Roles in WebLogic Server (Continued)

Role	Description
`Deployer`	Has the privilege to: • View the server configuration, *except* for encrypted attributes. • Deploy enterprise applications, startup and shutdown classes, and Web Application, EJB, J2EE Connector, and Web Service modules.
`Operator`	Has the privilege to: • View the server configuration, *except* for encrypted attributes. • Start, resume, and stop servers.
`Monitor`	Has the privilege to view the server configuration, except for encrypted attributes.

Table 3-5 Default Groups in WebLogic Server

Group	Description
`users`	Users, when they log in (for example, through a Web page).
`everyone`	Every user is a member of this group.
`Administrators`	By default, this group contains: • The user information entered as part of the installation process (that is, through the Configuration Wizard) • The system user if the WebLogic Server instance is running Compatibility Security. Any user assigned to the `Administrators` group is granted the `Administrator` security role by default and has full administrator privileges for all WebLogic Platform components, including WebLogic Integration and WebLogic Portal.
`Deployers`	By default, this group is empty. Any user assigned to the `Deployers` group is granted the `Deployer` security role by default.
`Operators`	By default, this group is empty. Any user assigned to the `Operators` group is granted the `Operator` security role by default.
`Monitors`	By default, this group is empty. Any user assigned to the `Monitors` group is granted the `Monitor` security role by default.

Default WebLogic Integration Security Roles and Groups

Table 3-6 and Table 3-7 lists and describes the default WebLogic Integration roles and groups created in a platform domain.

Table 3-6 Default WebLogic Integration Security Roles

Role	Description
`IntegrationAdmin`	WebLogic Integration administrator role. This role has full privileges to all servers in the cluster. This role can create additional roles using the administration console.
`IntegrationDeployer`	WebLogic Integration deployer role. This role has full privileges to all servers in the cluster. This role can create additional roles using the administration console.
`IntegrationOperator`	The WebLogic Integration operator role. This role has nearly all the privileges of the `IntegrationAdministrator` role. For example, a user in the `IntegrationOperator` role cannot configure certain security properties, but can otherwise modify resources.
`IntegrationMonitor`	The WebLogic Integration monitor role. This role has read-only access to the WebLogic Integration Administration Console.
`IntegrationUser`	The default WebLogic Integration user role. When first created, all users are assigned to the `IntegrationUser` role.
`TaskCreationRole`	Optional role that you can authorize to create Worklist Tasks, as described in "Configuring the Worklist Task Creation Role" in "System Configuration" in *Managing WebLogic Integration Solutions*. If you configure this role to be authorized to create Worklist Tasks, any user assigned to the `TaskCreationGroup` would have this privilege. However, by default, the `TaskCreationRole` does not have any special privileges until explicitly configured to have them.

Table 3-7 Default Groups in WebLogic Integration

Group	Description
`IntegrationAdministrators`	The WebLogic Integration administrator group. This group is assigned to the role `IntegrationAdmin`, and all members inherit the that role.
`IntegrationDeployers`	The WebLogic Integration deployer group. This group is assigned to the role `IntegrationDeployer`, and all members inherit the that role.
`IntegrationUsers`	The WebLogic Integration user group. This group is assigned to the role `IntegrationUser`, and all members inherit the that role.
`IntegrationMonitors`	The WebLogic Integration monitor group. This group is assigned to the role `IntegrationMonitor`, and all members inherit the that role.
`IntegrationOperators`	The WebLogic Integration operator group. This group is assigned to the role `IntegrationOperator`, and all members inherit the that role.
`TaskCreationGroup`	The WebLogic Integration group containing users, in addition to Integration Administrators, that are authorized to create new Worklist Tasks. This group is assigned to the role `TaskCreationRole`.

Default WebLogic Portal Security Roles and Groups

Table 3-8 and Table 3-9 list and describe the default WebLogic Portal roles and groups created in a platform domain.

Table 3-8 Default Security Roles in WebLogic Portal

Role	Description
`CustomerRole`	Role associated with the `wlcs_customer` group, which is used with commerce services. You can safely remove this role if your portal applications do not use commerce services.
`PortalSystemAdministrator`	The default WebLogic Portal system administrator role. This role has full privileges to all servers in the cluster. This role can create additional roles using the administration console.
`PortalSystemDelegator`	Top-level role for establishing delegated administration. All users in the `Administrators` group are assigned to this role by default.

Table 3-9 Default Groups in WebLogic Portal

Group	Description
`PortalSystemAdministrators`	The WebLogic Portal administrator group. This group is assigned to the role `PortalSystemAdministrator`, and all members inherit the that role.
`wlcs_customer`	Group used in commerce services for distinguishing portal customers from portal users. This group exists for compatibility purposes with WebLogic Portal 7.0. You can safely remove this group if your portal applications do not use commerce services.

CHAPTER 4

Using BEA WebLogic Enterprise Security

BEA provides a standalone security product, called BEA WebLogic Enterprise Security, that you can use with WebLogic Platform to administer application security for your whole environment. Whether this product is appropriate for your WebLogic Platform configuration depends on the the type of application platforms used in your enterprise environment, and your specific security requirements.

BEA WebLogic Enterprise Security is an infrastructure designed to provide security for multiple applications across an enterprise. These applications may be hosted on heterogeneous platforms, such as Netscape Web servers, Sun ONE Web servers, and Java applications, in addition to WebLogic Platform.

WebLogic Enterprise Security consists of the following:

- An administrative application that includes a central administration console
- A family of distributed service modules that interoperate with each of the supported environments. WebLogic Enterprise Security Version 4.1 includes service modules for WebLogic Platform 8.1, Netscape and Sun ONE web servers, and Java applications.

WebLogic Enterprise Security includes the following key features:

- Centralized policy administration with rule-based delegation

  Security administrators can define and deploy security policies without writing new code or redeploying applications. Rules and policies are all managed through a central administration console. In addition, policy verification and policy inquiry functions allow the administrator to validate security policy implementations prior to deployment.

- A rules-based policy model with integrated authentication, authorization, and auditing

  Security policies can be designed to model your business policies, and then implemented, tested, and distributed through a central administration application.

- A comprehensive and standards-based design that leverages the Lightweight Directory Access Protocol (LDAP), Security Assertion Markup Language (SAML), and Java

  Additional support for standard security technologies is provided. These technologies include J2EE security technologies, such as the Java Authentication and Authorization Service (JAAS), Java Secure Sockets Extensions (JSSE), and Java Cryptography Extensions (JCE).

- A set of Security Service Provider Interfaces (SSPI) to enable the development of custom security services

When you use WebLogic Enterprise Security with WebLogic Platform, the WebLogic Server 8.1 Security Service Module (SSM) replaces the WebLogic Platform security framework. This SSM ties all WebLogic Platform applications into a single WebLogic Enterprise Security administration application so that all WebLogic Platform administrative activities are performed through one administration application.

WebLogic Enterprise Security offers the following features that can supplement WebLogic Platform security:

- Multidomain Single Sign-On

  This feature allows user identity to be propagated from an application to WebLogic Server so that users are not required to authenticate themselves multiple times as they access WebLogic Server resources, including across domains.

- Ability to authenticate users from additional external stores

  WebLogic Enterprise includes Authentication providers that can work with user information stored in Windows NT realms and RDBMSs.

- Metadirectory feature for extended user profiles

  This feature allows you to create user profiles based on information that is distributed across multiple sources, and to create role and authorization policies based on the additional information contained in those profiles.

- Public API

  The Java Security Service Module provides a public application programming interface (API) that allows security developers to insert security services into their applications.

- Security policy segmentation

  WebLogic Enterprise Security allows you to apply security policies within components of an application as well as across domains.

- Enhanced auditing capabilities

  WebLogic Enterprise Security provides an implementation of log4j, which provides a wide range of flexibility in configuring the output of Auditing providers.

For more information, see the BEA WebLogic Enterprise Security 4.1 Technical Resource Center at the following Web site:

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

CHAPTER 5

Security Advisories and Online Support

The following sections provide information about and links to WebLogic Platform security advisories and other online support resources for security:

- BEA Security Advisories
- Reporting Security Issues
- dev2dev Security Resources

BEA Security Advisories

To keep you informed of the latest security advisories and to make sure you have access to security-related patches as soon as they become available, BEA maintains the BEA Advisories & Notifications page at:

`http://dev2dev.bea.com/advisories`

Reporting Security Issues

If you discover a possible security issue with WebLogic Platform 8.1 or any of its components, BEA recommends that you report it to `secalert@bea.com`.

dev2dev Security Resources

The BEA dev2dev Web site includes a Web page that provides links to several security-specific resources, including:

- Security code samples
- Training
- Newsgroups

For the list of security topics recommended on the dev2dev Web site, see:

http://dev2dev.bea.com/products/wlplatform81/security.jsp

The dev2dev site also hosts a Web page that provides answers to frequently asked questions (FAQ) about BEA WebLogic Platform and other BEA products. The site is updated monthly. To visit the dev2dev FAQ page, see:

http://dev2dev.bea.com/topitems/topsolutions/index.jsp

CHAPTER 6

Where to Find More Information

You can obtain more information about how to configure the security attributes used by the different WebLogic Platform components in the following guides:

Table 6-1 WebLogic Platform Security Documentation Topics

To learn about . . .	Refer to the following . . .
The hardware, operating systems, SDKs, database systems, Web Servers, and browsers that can be used with WebLogic Platform software	*Supported Configurations for WebLogic Platform 8.1*
WebLogic Server domains	"Overview of WebLogic Server System Administration" in *Configuring and Managing WebLogic Server*
Creating domains using the Configuration Wizard	*Creating WebLogic Configurations Using the Configuration Wizard*
Specifying the username and password of the administrator who can start a Platform server	"Configuring an Administrative Username and Password" in "Configuring Security" in *Creating WebLogic Configurations Using the Configuration Wizard*
Adding users and groups at the time you create the application domain	"Configuring Users and Groups" in "Configuring Security" in *Creating WebLogic Configurations Using the Configuration Wizard*
Using single-sign on	"Single Sign-On with Enterprise Information Systems" in *Managing WebLogic Security*

Table 6-1 WebLogic Platform Security Documentation Topics (Continued)

To learn about . . .	Refer to the following . . .
Web Services Security (WS-Security)	"Web Service Security (WS-Security)" in the *WebLogic Workshop Help* "Overview of Web Services Security" in "Configuring Security" in *Programming WebLogic Web Services*
WebLogic Server Security	
About WebLogic Server security	*Introduction to WebLogic Security*
All WebLogic Server security topics	Security in the WebLogic Server documentation Web site
Configuring security for WebLogic Server resources, including users, groups, roles, and application resources	*Securing WebLogic Resources*
WebLogic Server frequently asked questions	*WebLogic Server Frequently Asked Questions*
Upgrading security configuration from a previous WebLogic Server release	*WebLogic Server 8.1 Upgrade Guide*
Using the default security store	"Managing the Embedded LDAP Server" in *Managing WebLogic Security*
Securing resources and creating, adding, modifying, and deleting users and groups	"Users and Groups" in *Securing WebLogic Resources*
Configuring security policies on WebLogic Server resources	"Security Policies" in *Securing WebLogic Resources*
Configuring EJB security	"Using Declarative Security with EJBs" in "Securing Enterprise JavaBeans (EJBs)" in *Programming WebLogic Security*
Configuring Web service security	"Configuring Security" in *Programming WebLogic Web Services*
Configuring security providers	"Configuring Security Providers" in *Managing WebLogic Security*
Configuring security attributes for domains and servers	"Configuring Security for a WebLogic Domain" in *Managing WebLogic Security*

Table 6-1 WebLogic Platform Security Documentation Topics (Continued)

To learn about . . .	Refer to the following . . .
Using the WebLogic security APIs	*Programming WebLogic Security*
Security considerations for a WebLogic Server production environment	*Securing a Production Environment*
• Customizing WebLogic Server security • Configuring an LDAP authentication provider • Configuring and customizing WebLogic security providers	"Customizing the Default Security Configuration" in *Managing WebLogic Security*
Configuring the SSL protocol	"Configuring SSL" in *Managing WebLogic Security*
WebLogic Integration Security	
WebLogic Integration security	"Using WebLogic Integration Security" in *Deploying WebLogic Integration Solutions*
Upgrading a security configuration from a previous release	*WebLogic Integration Upgrade Guide*
Adding users, groups, and roles	"User Management" in *Managing WebLogic Integration Solutions*
Setting up security for trading partner integration business protocols	"Trading Partner Integration Security" in *Introducing Trading Partner Integration*
Restricting access to the different resources of a WebLogic Integration process	"Process Security Policies" in "Process Configuration" in *Managing WebLogic Integration Solutions*
Restricting access to resources used by the Message Broker	"Setting Channel Security Policies" in "Message Broker" in *Managing WebLogic Integration Solutions*
Adding security to business processes	"@common:security Annotation" in the *WebLogic Workshop Help*
WebLogic Portal Security	
Securing Portal applications	"Securing Portal Applications" in the *WebLogic Workshop Help*

Table 6-1 WebLogic Platform Security Documentation Topics (Continued)

To learn about . . .	Refer to the following . . .
Upgrading a security configuration from a previous release	*WebLogic Portal Upgrade Guide*
Adding users and groups	"Overview of Users and Groups" in the *WebLogic Administration Portal Online Help*
Configuring security roles	"Delegated Administration Overview" in the *WebLogic Administration Portal Online Help*
Configuring security policies on portal resources	"Delegated Administration Overview" and "Overview of Visitor Entitlements" in the *WebLogic Administration Portal Online Help*
WebLogic Workshop Security	
Security overview	WebLogic Workshop Security Overview in the *WebLogic Workshop Help*
Web service security	Web Service Security (WS-Security) in the *WebLogic Workshop Help*
Transport security	Transport Security in the *WebLogic Workshop Help*
Role-based security	An Overview of Role-Based Security in the *WebLogic Workshop Help*