



BEA WebLogic Server™

Securing WebLogic Resources

Copyright

Copyright © 2004 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, 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 Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Contents

About This Document

Audience	viii
e-docs Web Site	viii
How to Print the Document	viii
Related Information	viii
Contact Us!	ix
Documentation Conventions	x

1. Introduction to Securing WebLogic Resources

Overview of Securing WebLogic Resources	1-1
Audience for This Guide	1-3
Terms and Concepts	1-3
Securing WebLogic Resources: Main Steps	1-3
Best Practices: Configure Entitlements Caching When Using WebLogic Providers	1-5

2. Types of WebLogic Resources

Overview of WebLogic Resource Types	2-1
Administrative Resources	2-2
Application Resources	2-2
Enterprise Information Systems (EIS) Resources	2-2
COM Resources	2-3
Java DataBase Connectivity (JDBC) Resources	2-3
Java Messaging Service (JMS) Resources	2-4

Java Naming and Directory Interface (JNDI) Resources	2-5
Server Resources	2-5
Layered Security Scheme for Server Resources	2-6
Security Policies for Server Resources	2-7
MBean Protections	2-7
How the WebLogic Security Service Verifies Layered Protections	2-8
Example of Layered Protection for a Server Resource	2-8
Maintaining a Consistent Security Scheme	2-10
Permissions for Starting and Shutting Down Servers	2-11
Permissions for Using the weblogic.Server Command	2-11
Permissions for Using the Node Manager	2-12
Shutting Down a WebLogic Server Instance	2-12
URL (Web) and EJB (Enterprise JavaBean) Resources	2-12
Techniques for Securing URL and EJB Resources	2-13
Using the WebLogic Server Administration Console	2-13
Using Deployment Descriptors	2-13
Using the Administration Console and Deployment Descriptors	2-14
Prerequisites for Securing URL and EJB Resources	2-14
Understanding How to Check Security Roles and Security Policies	2-15
Understanding What to Do on Future Redeploys of the WebLogic Resource	2-17
How to Change the Check Roles and Policies and Future Redeploys Settings	2-17
Understanding How These Settings Interact	2-18
Using the Combined Technique to Secure Your URL and EJB Resources	2-19
Copying Security Configurations	2-20
Reinitializing Security Configurations	2-27
Web Service Resources	2-29

3. Users and Groups

Overview of Users and Groups	3-2
Creating Users	3-2
Adding Users to Groups	3-3
Modifying Users	3-4
Deleting Users	3-4
Default Groups.	3-5
Creating Groups.	3-6
Nesting Groups	3-7
Modifying Groups	3-7
Deleting Groups.	3-8

4. Security Roles

Overview of Security Roles.	4-1
Dynamic Role Mapping.	4-2
Types of Security Roles: Global Roles and Scoped Roles	4-3
Using the Administration Console to Create Security Roles.	4-3
Default Global Roles	4-5
Protected MBean Attributes and Operations	4-6
Default Group Associations.	4-11
Components of a Security Role: Role Conditions, Expressions, and Role Statements.	4-12
Working with Global Roles.	4-13
Creating Global Roles	4-14
Modifying Global Roles.	4-17
Deleting Global Roles	4-18
Working with Scoped Roles	4-18
Creating Scoped Roles	4-19
Step 1: Select the WebLogic Resource	4-19

Step 2: Create the Scoped Role.	4-27
Step 3: Create the Role Conditions.	4-27
Modifying Scoped Roles	4-30
Deleting Scoped Roles	4-31

5. Security Policies

Overview of Security Policies.	5-1
Security Policy Granularity and Inheritance	5-2
Security Policy Storage and Prerequisites for Use	5-2
Default Security Policies.	5-3
Protected Public Interfaces	5-4
Components of a Security Policy: Policy Conditions, Expressions, and Policy Statements	5-5
Working with Security Policies.	5-7
Creating Security Policies	5-7
Step 1: Select the WebLogic Resource	5-7
Step 2: Create the Policy Conditions	5-18
Modifying Security Policies.	5-21
Deleting Security Policies	5-21

Index

About This Document

This document introduces the various types of WebLogic resources, and provides information that allows you to secure these resources using WebLogic Server.

The document is organized as follows:

- [Chapter 1, “Introduction to Securing WebLogic Resources,”](#) is an overview of this document (such as its intended audience), an overview of securing WebLogic resources, and a “main steps” section for those wanting to skip directly to instructional sections.
- [Chapter 2, “Types of WebLogic Resources,”](#) describes the different types of WebLogic resources and provides important information about securing some of the more complex and common WebLogic resources.
- [Chapter 3, “Users and Groups,”](#) describes users and groups, including WebLogic Server default groups. This section also includes step-by-step instructions that tell you how to work with users and groups in the WebLogic Server Administration Console.
- [Chapter 4, “Security Roles,”](#) describes security roles, including WebLogic Server default global roles. This section also explains the difference between global and scoped roles and describes the components of a security role. Last, this section includes step-by-step instructions that tell you how to work with global and scoped roles in the Administration Console.
- [Chapter 5, “Security Policies,”](#) describes security policies, including WebLogic Server default security policies. This section also describes the components of a security policy, and provides step-by-step instructions that tell you how to work with security policies in the Administration Console.

Audience

This document is written primarily for Server Administrators. **Server Administrators** work closely with Application Architects to design a security scheme for the server and the applications running on the server, to identify potential security risks, and to propose configurations that prevent security problems. Related responsibilities may include maintaining critical production systems; configuring and managing security realms; implementing authentication and authorization schemes for server and application resources; upgrading security features; and maintaining security provider databases. Server Administrators have in-depth knowledge of the Java security architecture, including Enterprise Application, Web Application and EJB security, Public Key security, and SSL.

This document is written for Server Administrators who use the WebLogic Server Administration Console, and should be used in conjunction with *Managing WebLogic Security* to ensure that security is completely configured for a WebLogic Server deployment.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation.

How to Print the Document

You can print a copy of this document from a Web browser, one main topic at a time, by using the File→Print option on your Web browser.

A PDF version of this document is available on the WebLogic Server documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Server documentation Home page, click Download Documentation, and select the document you want to print.

Adobe Acrobat Reader is available at no charge from the Adobe Web site at <http://www.adobe.com>.

Related Information

The BEA corporate Web site provides all documentation for WebLogic Server. Other WebLogic Server documents that may be of interest to Server Administrators wanting to secure WebLogic resources are:

- [Managing WebLogic Security](#)
- “Securing Web Applications,” “Securing Enterprise JavaBeans (EJBs),” and “Using Java Security to Protect WebLogic Resources” in *Programming WebLogic Security*.
- “Configure Access Control” in *Programming WebLogic jCOM* (COM resources).
- “Security” in *Programming WebLogic J2EE Connectors* (EIS resources).
- “Configuring Security” in *Programming WebLogic Web Services* (Web Services resources).

The following Avitek Medical Records Tutorials illustrate the concepts and procedures explained throughout this document, and may also prove helpful:

- [Tutorial 14: Securing Application and URL \(Web\) Resources Using the Administration Console](#)
- [Tutorial 15: Securing Enterprise JavaBean \(EJB\) Resources Using the Administration Console](#)
- [Tutorial 16: Copying and Reinitializing Security Configurations](#)

Additional security documents are listed on the [Security page](#).

Contact Us!

Your feedback on BEA 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 documentation.

In your e-mail message, please indicate the software name and version you are using, as well as the title and document date of your documentation. If you have any questions about this version of BEA WebLogic Server, or if you have problems installing and running BEA WebLogic Server, contact BEA Customer Support through BEA WebSupport at <http://www.bea.com>. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

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

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using

- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Usage
Ctrl+Tab	Keys you press simultaneously.
<i>italics</i>	Emphasis and book titles.
monospace text	Code samples, commands and their options, Java classes, data types, directories, and file names and their extensions. Monospace text also indicates text that the user is told to enter from the keyboard. <i>Examples:</i> <pre>import java.util.Enumeration; chmod u+w * config/examples/applications .java config.xml float</pre>
<i>monospace italic text</i>	Placeholders. <i>Example:</i> <pre>String CustomerName;</pre>
UPPERCASE MONOSPACE TEXT	Device names, environment variables, and logical operators. <i>Examples:</i> <pre>LPT1 BEA_HOME OR</pre>
{ }	A set of choices in a syntax line.
[]	Optional items in a syntax line. <i>Example:</i> <pre>java utils.MulticastTest -n name -a address [-p portnumber] [-t timeout] [-s send]</pre>

Convention	Usage
	<p>Separates mutually exclusive choices in a syntax line. <i>Example:</i></p> <pre>java weblogic.deploy [list deploy undeploy update] password {application} {source}</pre>
...	<p>Indicates one of the following in a command line:</p> <ul style="list-style-type: none"> • An argument can be repeated several times in the 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.

About This Document

Introduction to Securing WebLogic Resources

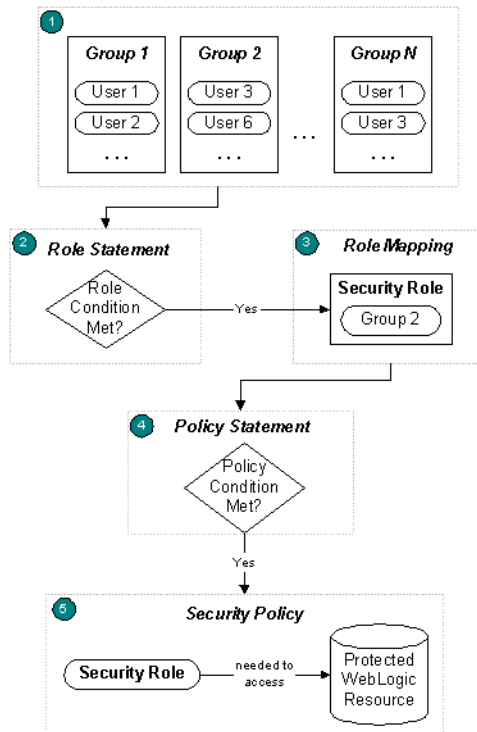
The following sections describe key concepts and tasks for securing WebLogic resources:

- [“Overview of Securing WebLogic Resources” on page 1-1](#)
- [“Audience for This Guide” on page 1-3](#)
- [“Terms and Concepts” on page 1-3](#)
- [“Securing WebLogic Resources: Main Steps” on page 1-3](#)
- [“Best Practices: Configure Entitlements Caching When Using WebLogic Providers” on page 1-5](#)

Overview of Securing WebLogic Resources

A **WebLogic resource** represents an underlying WebLogic Server entity that can be protected from unauthorized access using security roles and security policies. Examples of WebLogic resources include Enterprise Applications (EARs), EJBs (JARs), and Web Applications (WARs). For more information about the different types of WebLogic resources, see [Chapter 2, “Types of WebLogic Resources.”](#)

[Figure 1-1](#) illustrates the overall process for securing WebLogic resources, and a brief explanation follows.

Figure 1-1 Securing WebLogic Resources

- Administrators statically assign users to groups, which can represent organizational boundaries. The same user can be a member of multiple groups. [Figure 1-1](#) shows three groups with two users each. User 1 and User 3 are members of multiple groups.

BEA recommends assigning users to groups because doing so increases efficiency for administrators who work with many users.
- Administrators create a security role based on their organization's established business procedures. The security role consists of one or more role statements, each of which include a role condition. The role condition specifies the circumstances under which a particular group should be granted the security role.
- At runtime, the WebLogic Security Service compares the groups against the role condition(s) to determine whether users in the group should be dynamically granted a security role. This process is referred to as **role mapping**. In [Figure 1-1](#), Group 2 is the only group that is granted a security role.

Individual users can also be granted a security role, but this is a less typical practice.

4. Administrators create a security policy based on their organization's established business procedures. The security policy consists of one or more policy statements, each of which include a policy condition. The policy condition specifies the circumstances under which a particular security role should be granted access to a protected WebLogic resource.
5. At runtime, the WebLogic Security Service uses the security policy and the WebLogic resource itself to determine whether access to the protected WebLogic resource should be granted. Only users who are members of the group that is granted the security role can access the WebLogic resource. In [Figure 1-1](#), User 3 and User 6 can access the protected WebLogic resource because they are members of Group 2, and Group 2 is granted the necessary security role.

Audience for This Guide

This document is written primarily for Server Administrators. **Server Administrators** work closely with Application Architects to design a security scheme for the server and the applications running on the server, to identify potential security risks, and to propose security configurations that prevent security problems. Related responsibilities can include maintaining critical production systems; configuring and managing security realms; implementing authentication and authorization schemes for server and application resources; upgrading security features; and maintaining security provider databases. Server Administrators have in-depth knowledge of the Java security architecture, including Enterprise Application, Web Application and EJB security, Public Key security, and SSL.

This document written for Server Administrators who use the WebLogic Server Administration Console, and should be used in conjunction with [Managing WebLogic Security](#) to ensure that security is completely configured for a WebLogic Server deployment.

Terms and Concepts

WebLogic Server security includes many unique terms and concepts that you need to understand. These terms and concepts—which you will encounter throughout the WebLogic Server security documentation—are defined in the “[Terminology](#)” section and the “[Security Fundamentals](#)” section of *Introduction to WebLogic Security*, respectively.

Securing WebLogic Resources: Main Steps

The main steps for securing a WebLogic resource are:

1. Determine which WebLogic resource to secure. See [Chapter 2, “Types of WebLogic Resources.”](#)

2. If you want to secure a URL (Web) or EJB (Enterprise JavaBean) resource:
 - a. Decide which technique you will use. See [“Techniques for Securing URL and EJB Resources” on page 2-13](#).
 - b. Review important information about securing URL and EJB resources to prevent overriding security configurations. See [“Prerequisites for Securing URL and EJB Resources” on page 2-14](#).
 - c. If you want to use the WebLogic Server Administration Console to secure your URL or EJB resource, follow the instructions in step 3.
 - d. If you want to use deployment descriptors to secure your URL or EJB resource, see [“Adding Declarative Security to Web Applications”](#) or [“Adding Declarative Security to EJBs”](#) in *Programming WebLogic Security*, respectively.
 - e. If you want to copy security configurations from existing deployment descriptors upon the initial deployment of URL or EJB resources, or reinitialize the security configuration for URL or EJB resources to their original state (as specified in the deployment descriptors), follow the instructions in [“Using the Combined Technique to Secure Your URL and EJB Resources” on page 2-19](#).
3. Use the Administration Console to secure your WebLogic resource:
 - a. Create users and groups—representations of individuals and collections of individuals—who may be granted a security role. See [“Creating Users” on page 3-2](#) and [“Creating Groups” on page 3-6](#) for step-by-step instructions.
 - b. Create security roles—dynamically computed privileges granted to users or groups based on specific conditions—which are used to restrict access to WebLogic resources. See [“Working with Scoped Roles” on page 4-18](#) for step-by-step instructions.

BEA recommends creating security roles and using them (rather than users or groups) to secure WebLogic resources, because doing so increases efficiency for administrators who work with many users.
 - c. Create a security policy—an association between the WebLogic resource and a user, group, or security role—that specifies who has access to the WebLogic resource. See [“Working with Security Policies” on page 5-7](#).

Best Practices: Configure Entitlements Caching When Using WebLogic Providers

The WebLogic Authorization provider (`DefaultAuthorizer`) and the WebLogic Role Mapping provider (`DefaultRoleMapper`) improve performance by caching the roles, predicates, and resource data that they look up. If you use these WebLogic providers, you can configure the maximum number of items that they store in the caches.

By default, the WebLogic Authorization and Role Mapping providers store the following number of items in each cache:

- 2000 items in the roles cache

This cache contains the name of each role that has been looked up and the policy that protects it.

- 200 items in the predicates cache

This cache contains each predicate that the WebLogic entitlements engine has looked up.

- 5000 items in the resources cache

This cache contains the name of each resource that has been looked up and the policy that protects it.

If a cache exceeds its maximum size, the WebLogic entitlements engine removes the least recently used (LRU) item from the cache.

If the applications on a WebLogic Server instance use more than 2000 roles or 5000 resources, consider increasing the cache sizes. (The WebLogic providers include less than 50 predicates, so there is no need to increase the size of this cache.)

To change the maximum number of items that a cache contains, pass one of the following system properties in the `java` startup command for a WebLogic Server instance:

- `-Dweblogic.entitlement.engine.cache.max_role_count=max-roles`

where *max-roles* is the maximum number of roles that you want to cache.

- `-Dweblogic.entitlement.engine.cache.max_predicate_count=max-predicates`

where *max-predicates* is the maximum number of predicates that you want to cache.

- `-Dweblogic.entitlement.engine.cache.max_resource_count=max-resources`

where *max_resource_count* is the maximum number of resources that you want to cache.

By default, the WebLogic providers add items to the cache as they use them. With this configuration, the initial lookup of entitlement data takes longer than subsequent lookups. You can, however, decrease the amount of time needed for an initial lookup by configuring a WebLogic Server instance to load the caches during its startup cycle. To do so, pass the following system property to the server's `java` startup command:

- `-Dweblogic.entitlement.engine.cache.preload=true`

For example:

```
java -Dweblogic.entitlement.engine.cache.max_role_count=6001
     -Dweblogic.entitlement.engine.cache.max_resource_count=3001
     -Dweblogic.entitlement.engine.cache.preload=true
     weblogic.Server
```

Types of WebLogic Resources

The following sections describe the types of resources included in WebLogic Server:

- “Overview of WebLogic Resource Types” on page 2-1
- “Administrative Resources” on page 2-2
- “Application Resources” on page 2-2
- “Enterprise Information Systems (EIS) Resources” on page 2-2
- “COM Resources” on page 2-3
- “Java DataBase Connectivity (JDBC) Resources” on page 2-3
- “Java Messaging Service (JMS) Resources” on page 2-4
- “Java Naming and Directory Interface (JNDI) Resources” on page 2-5
- “Server Resources” on page 2-5
- “URL (Web) and EJB (Enterprise JavaBean) Resources” on page 2-12
- “Web Service Resources” on page 2-29

Overview of WebLogic Resource Types

WebLogic resources are hierarchical. Therefore, the level at which you define security roles and security policies is up to you. For example, you can define security roles and security policies for

an entire Enterprise Application (EAR), an Enterprise JavaBean (EJB) JAR containing multiple EJBs, a particular EJB within that JAR, or a single method within that EJB.

Administrative Resources

An **Administrative resource** is a type of WebLogic resource that allows users to perform administrative tasks. Examples of Administrative resources include the WebLogic Server Administration Console, the `weblogic.Admin` tool, and MBean APIs.

Administrative resources are limited in scope. Currently, you can only secure the User Lockout operation on an Administrative resource using the WebLogic Server Administration Console. This operation provides compatibility with WebLogic Server 6.x., and allows users who meet the security requirements to unlock users who have been locked out of their accounts. For more information about user lockout, see [“Protecting User Accounts”](#) in *Managing WebLogic Security*.

Application Resources

An **Application resource** is a type of WebLogic resource that represents an Enterprise Application, packaged as an EAR (Enterprise Application aRchive) file. Unlike the other types of WebLogic resources, the hierarchy of an Application resource is a mechanism for containment, rather than a type hierarchy. You secure an Application resource when you want to protect multiple WebLogic resources that *constitute* the Enterprise Application (for example, EJB resources, URL resources, and Web Service resources). In other words, securing an Enterprise Application will cause all the WebLogic resources within that application to inherit its security configuration.

You can also secure, on an individual basis, the WebLogic resources that constitute an Enterprise Application (EAR). Securing a resource by both means causes the individual security configuration to override the security configuration inherited from the Enterprise Application for that WebLogic resource.

Enterprise Information Systems (EIS) Resources

A J2EE Connector is a system-level software driver used by an application server such as WebLogic Server to connect to an Enterprise Information System (EIS). BEA supports Connectors developed by EIS vendors and third-party application developers that can be deployed in any application server supporting the Sun Microsystems J2EE Platform Specification, Version 1.3. Connectors, also known as **Resource Adapters**, contain the Java, and if necessary, the native components required to interact with the EIS.

An **Enterprise Information System (EIS) resource** is a specific type of WebLogic resource that is designed as a Connector. To secure access to an EIS, you create security policies and security roles for all Connectors as a group, or for individual Connectors.

Information about securing EIS resources can be found both in this document, and in the “[Security](#)” section of *Programming WebLogic J2EE Connectors*. Instructions for creating the credential maps for use with EIS resources are available in the “[Single Sign-On with Enterprise Information Systems](#)” section of *Managing WebLogic Security*.

COM Resources

WebLogic jCOM is a software bridge that allows bidirectional access between Java/J2EE objects deployed in WebLogic Server, and Microsoft ActiveX components available within the Microsoft Office family of products, Visual Basic and C++ objects, and other Component Object Model/Distributed Component Object Model (COM/DCOM) environments.

A **COM resource** is a specific type of WebLogic resource that is designed as a program component object according to Microsoft's framework. To secure COM components accessed through BEA's bi-directional COM-Java (jCOM) bridging tool, you create security policies and security roles for packages containing multiple COM classes, or for individual COM classes.

Information about securing COM resources can be found both in this document and in the “[Configuring Access Control](#)” section of *Programming WebLogic jCOM*.

Java DataBase Connectivity (JDBC) Resources

A **Java DataBase Connectivity (JDBC) resource** is a specific type of WebLogic resource that is related to JDBC. To secure JDBC database access, you can create security policies and security roles for all connection pools as a group, individual connection pools, and MultiPools. When you secure individual connection pools, you can choose whether to protect all operations on the connection pool, or protect one of the following operations:

- **admin**—The following methods on the `JDBCConnectionPoolRuntimeMBean` are invoked as admin operations: `clearStatementCache`, `destroy`, `disableDroppingUsers`, `disableFreezingUsers`, `enable`, `forceDestroy`, `forceShutdown`, `forceSuspend`, `getProperties`, `poolExists`, `resume`, `shutdown`, `shutdownHard`, `shutdownSoft`, and `suspend`.
- **reserve**—Applications reserve a connection in the connection pool by looking up the data source that points to the connection pool and then calling `getConnection`.

Note: Giving a user the `reserve` permission enables them to execute vendor-specific operations on the connection. Depending on the database vendor, some of these operations may have database security implications.

- `shrink`—Shrinks the connection pool to the maximum of the currently reserved connections or the initial size.
- `reset`—Resets the database connection pool by shutting down and re-establishing all physical database connections. This also clears the statement cache for each connection in the connection pool. You can only reset a normally running connection pool.

Note: If a security policy controls access to a connection pool that is in a MultiPool, access checks are performed at both levels of the JDBC resource hierarchy (once at the MultiPool level, and again at the individual connection pool level). As with all types of WebLogic resources, this double-checking ensures that the most restrictive security policy controls access.

Note: If you are an Oracle user, you can also control access to JDBC resources using an Oracle Virtual Private Database (VPD). For more information, see “[Programming with Oracle Private Virtual Databases](#)” in *Using Third-Party Drivers with WebLogic Server*.

Java Messaging Service (JMS) Resources

A **Java Messaging Service (JMS) resource** is a specific type of WebLogic resource that is related to JMS. To secure JMS destinations, you create security policies and security roles for all destinations (JMS queues and JMS topics) as a group, or an individual destination (JMS queue or JMS topic) on a JMS server. When you secure a particular destination on a JMS server, you can protect all operations on the destination, or protect one of the following operations:

- `send`—Required to send a message to a queue or a topic. This includes calls to the `MessageProducer.send()`, `QueueSender.send()`, and `TopicPublisher.publish()` methods, as well as the Messaging Bridge.
- `receive`—Required to create a consumer on a queue or a topic. This includes calls to the `Session.createConsumer()`, `Session.createDurableSubscriber()`, `QueueSession.createReceiver()`, `TopicSession.createSubscriber()`, `TopicSession.createDurableSubscriber()`, `Connection.createConnectionConsumer()`, `Connection.createDurableConnectionConsumer()`, `QueueConnection.createConnectionConsumer()`, `TopicConnection.createConnectionConsumer()`, and `TopicConnection.createDurableConnectionConsumer()` methods, as well as the Messaging Bridge and message-driven beans.

- **browse**—Required to view the messages on a queue using the `QueueBrowser` interface.

Java Naming and Directory Interface (JNDI) Resources

JNDI provides a common-denominator interface to many existing naming services, such as Lightweight Directory Access Protocol (LDAP) and Domain Name System (DNS). These naming services maintain a set of bindings, which relate names to objects and provide the ability to look up objects by name. JNDI allows the components in distributed applications to locate each other.

JNDI is independent of any specific naming or directory service implementation. It supports the use of a number of methods for accessing various new and existing services. This support allows any service-provider implementation to be plugged into the JNDI framework using the standard service provider interface (SPI) conventions.

A **Java Naming and Directory Interface (JNDI) resource** is a specific type of WebLogic resource that uses the industry-standard JNDI SPI to enable connectivity to heterogeneous enterprise naming and directory services. To secure access to the JNDI tree, you create security policies and security roles for the entire JNDI tree, or for an individual branch of that tree. Regardless, you can protect all operations, or protect one of the following operations:

- **modify**—Whenever an application modifies the JNDI tree in any way (that is, adding, removing, changing) the current user must have permission to make the modification. This includes the `bind()`, `rebind()`, `createSubContext()`, `destroySubContext()`, and `unbind()` methods.
- **lookup**—Whenever an application looks up an object in the JNDI tree, the current user must have permission to perform the lookup. This includes the `lookup()` and `lookupLink()` methods.
- **list**—Whenever an application lists the contents of a context in JNDI, the current user must have permission to perform the list. This includes the `list()` and `listBindings()` methods.

Server Resources

Many engineering teams divide administration responsibilities into distinct roles. Each project might give only one or two team members permission to deploy applications or modules, but allow all team members to view the server configuration. As described in [Chapter 5, “Security Policies,”](#) WebLogic Server supports this division of responsibility by allowing administrators to

secure WebLogic resources with security policies. Typically, these security policies are based on whether users or groups of users are granted a particular security role.

A **Server resource** is a specific type of WebLogic resource that is used to protect activities that control the running state of a WebLogic Server instance. To secure Server resources, you create security policies and security roles for all WebLogic Server instances (servers) as a group, or individual servers. When you secure a particular server, you can protect all operations on the server, or protect one of the following operations:

- **boot**—A user who tries to start a WebLogic Server instance, either an Administration Server or Managed Server, must have permission to do so. This action is typically initiated through a call to the `java weblogic.Server` command on the command line, by a configured start script (which in turn calls the `java weblogic.Server` command), or through the Node Manager capabilities that allow for remote start of WebLogic Server instances.
- **shutdown**—A user who tries to shut down a running WebLogic Server instance, either an Administration Server or Managed Server, must have permission to do so. This action is typically initiated through the WebLogic Server Administration Console or the `weblogic.Admin SHUTDOWN` or `FORCESHUTDOWN` commands.
- **lock**—A user who tries to prohibit additional logins (logins other than for privileged administrative actions) to a running WebLogic Server instance, either an Administration Server or Managed Server, must have permission to do so. This action is typically initiated through the Administration Console or the `weblogic.Admin LOCK` commands.
- **unlock**—A user who tries to re-enable non-privileged logins to a running WebLogic Server instance, either an Administration Server or Managed Server, must have permission to do so. This action is typically initiated through the Administration Console or the `weblogic.Admin UNLOCK` commands.

Like other types of WebLogic resources, a Server resource and its operations are secured with security policies. However, because the configuration of a server is exposed through a set of MBeans, a Server resource also has additional protections that affect how administrators access MBean operations.

Layered Security Scheme for Server Resources

The following sections provide more information about the layered security scheme for Server resources:

- [“Security Policies for Server Resources” on page 2-7](#)

- [“MBean Protections” on page 2-7](#)
- [“How the WebLogic Security Service Verifies Layered Protections” on page 2-8](#)
- [“Example of Layered Protection for a Server Resource” on page 2-8](#)
- [“Maintaining a Consistent Security Scheme” on page 2-10](#)

Security Policies for Server Resources

Like other types of WebLogic resources, a Server resource is secured with security policies through the WebLogic Server Administration Console.

More specifically, all server resources inherit a default security policy that is based on the `Admin` and `Operator` default global security roles. As described in [“Default Global Roles” on page 4-5](#), the `Admin` and `Operator` global roles are given specific privileges that are required in order for administrators to interact with administrative interfaces like the Administration Console or the `weblogic.Admin` command. These default global roles are based on the default groups (described in [“Default Group Associations” on page 4-11](#)). Therefore, administrators who need access to Server resources need to be members of either the `Administrators` or `Operators` default groups.

Note: Because WebLogic Server grants the four default global roles to four default groups, adding a user to one of these groups automatically grants the user the global role.

Caution: Do not modify the default security policies for Server resources to make them more restrictive. Eliminating some of the existing security roles might negatively affect the functioning of WebLogic Server. However, if you like, you can make the default security policies more inclusive (for example, by adding new security roles).

MBean Protections

Each type of WebLogic resource (including a Server resource) exposes a set of its operations through its own implementation of the `weblogic.security.spi.Resource` interface (the `weblogic.security.service.ServerResource` class for Server resources). Therefore, the `ServerResource` class is the entity that is actually secured by the security policy described in [“Security Policies for Server Resources” on page 2-7](#).

In WebLogic Server, the configuration of a Server resource is exposed through a set of MBeans. As such, the actions that the `ServerResource` class protects correspond to underlying MBean attributes and operations. For example, the `Resource` interface’s `start()` method maps directly to the `start` operation of the `ServerRuntime` MBean.

The MBeans that expose the configuration of a Server resource are protected using one of the four default global roles. This protection is *in addition to* the security policy on the Server resource and is currently an unconfigurable protection supplied by the WebLogic Security Service. Therefore, although you can create your own global roles for securing Server resources, only users granted one of the default global roles can view or change the configuration of a server.

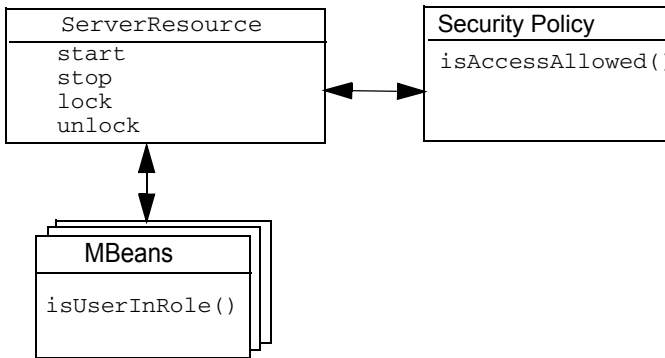
How the WebLogic Security Service Verifies Layered Protections

When an administrator tries to interact with a Server resource, the WebLogic Security Service:

- Determines whether the user is granted one of the default global roles permitted to change the attributes of the MBean or invoke the operations of the MBean
- Checks the default security policy for the Server resource to verify that the user meets the requirement defined by that security scheme

Therefore, a user must satisfy both security schemes for the request to be successful. [Figure 2-1](#) shows how a security policy on the Server resource interacts with the security role-based protections on the underlying MBeans.

Figure 2-1 Layered Protections for Server Resources



Because the privileges given by the MBean protections are immutable, it is necessary to maintain security policies in a way that ensures consistency. (For more information, see [“Maintaining a Consistent Security Scheme”](#) on page 2-10.)

Example of Layered Protection for a Server Resource

This example illustrates how one Server resource is protected by the layered security scheme.

An administrator with the user name `JDoe` wants to start the server called `myserver`. This administrative user (`JDoe`) is a member of the default group `Administrators`, which by default is granted the `Admin` global security role. You set up user-to-group and group-to-security role configuration through the WebLogic Server Administration Console, as described in other sections of this guide.

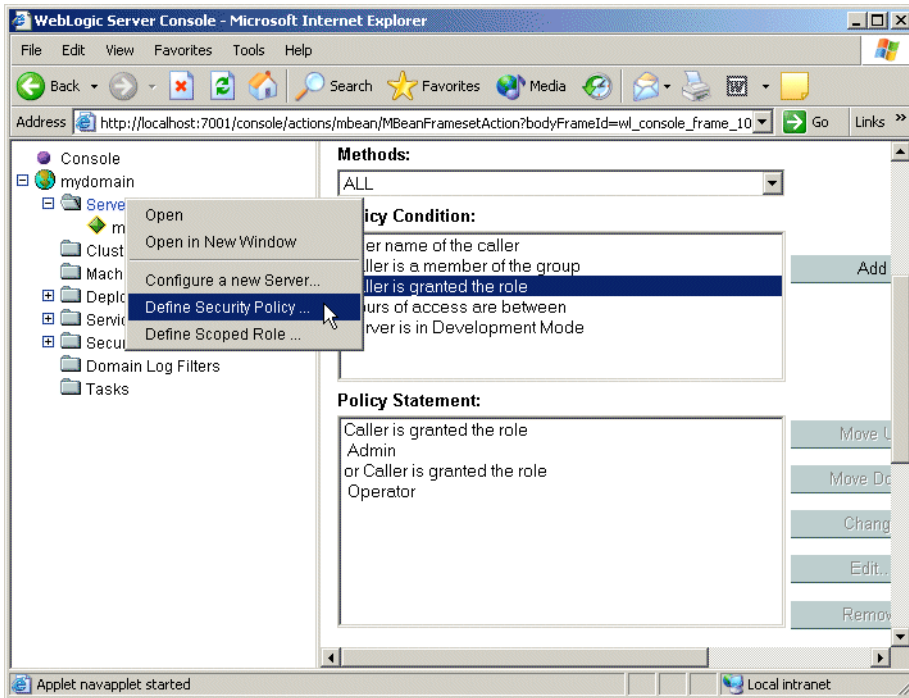
Part 1: MBean Protections

Because starting a server requires interactions with various MBeans, and because MBean protections are an unconfigurable protection supplied by the WebLogic Security Service, a user wanting to perform such an operation must be in the `Admin` or `Operator` default global roles. For example, [Table 4-5, “Privileges for the Admin or Operator Default Global Roles,” on page 4-11](#) shows that access to the `Server` and `ServerRuntime` MBeans (MBeans with `start` operations) is a privilege given only to users in these default security roles. Because the administrative user `JDoe` is a member of the default group `Administrators`, he is also granted the `Admin` global security role, and therefore fulfills the first part of the dual security scheme for Server resources.

Part 2: Security Policy on the Server Resource

As the Policy Editor page in [Figure 2-2](#) shows, the default security policy for `myserver` (viewed by right-clicking on `myserver` in the navigation tree and selecting the Define Security Policy... option) allows users granted the `Admin` or `Operator` global roles to interact with this Server resource. Because the administrative user `JDoe` is a member of the default group `Administrators`, he is also granted the `Admin` global security role, and therefore fulfills the second part of the layered security scheme for Server resources.

Figure 2-2 Default Security Policy for the myserver Server



Notes: Had the administrative user JDoe been a member of the `Operators` group (and therefore granted the `Operator` default global role), he would have still fulfilled both parts of the dual security scheme.

For a detailed explanation of the Policy Editor page shown in [Figure 2-2, “Default Security Policy for the myserver Server,”](#) on page 2-10, see “Components of a Security Policy: Policy Conditions, Expressions, and Policy Statements” on page 5-5.

Maintaining a Consistent Security Scheme

The WebLogic Server default configuration of groups, global roles, security policies on Server resources, and MBean protections work together to create a consistent security scheme. You can, however, make modifications that limit access in ways that you do not intend. Be certain that any modifications you make to the default security settings do not prevent a user from being authorized by both the MBean protections and the security policy on the Server resource.

For example, if you use the WebLogic Server Administration Console to add a user to the `Operator` global role, but fail to use the `Operator` global role in the security policy defined for

a Server resource, the user can call MBean operations that are used in the startup and shutdown sequence, but cannot use any Server resource operations to start or stop a server. Similarly, if you use the Administration Console to remove the `Operator` global role from a security policy on the Server resource, a user granted the `Operator` global role can still call MBean operations but cannot call the Server resource. This result occurs because MBean protections for the default global roles are part of the WebLogic Security Service and are not currently configurable.

To keep MBean protections synchronized with security policies, consider taking the following actions when you create or modify a security policy:

- Always give the `Admin` global role access to a Server resource.
- For a security policy on a server, use the `Operator` global role.
Failure to use the `Operator` global role or a security role nested within this default global role may result in problems with the WebLogic Security Service.
- For a security policy on a deployable resource (such as an application, EJB module, Web Application module, Connector module, or startup/shutdown class), use the `Deployer` global role.

For more information about security policies, see [“Working with Security Policies” on page 5-7](#).

Permissions for Starting and Shutting Down Servers

WebLogic Server provides two ways to start and shut down WebLogic Server instances (servers): the `weblogic.Server` command and the Node Manager. Because the underlying components for the `weblogic.Server` command and the Node Manager are different, the two commands use different authorization methods.

The following sections provide more information about the permissions for starting and shutting down servers:

- [Permissions for Using the `weblogic.Server` Command](#)
- [Permissions for Using the Node Manager](#)
- [Shutting Down a WebLogic Server Instance](#)

Permissions for Using the `weblogic.Server` Command

The `weblogic.Server` command, which you can use to start both Administration and Managed Servers, calls methods that are protected by a security policy on the Server resource. To use this command, you must satisfy the requirements of the security policy on the Server resource.

Some `weblogic.Server` arguments set attributes for MBeans. However, because these arguments modify an MBean before the server is in the `RUNNING` state, the security policy on the Server resource, not the protection on the MBean, is the authorizer. For example, a user in the `Operator` global role can use the `-Dweblogic.ListenPort` argument to change a server's default listen port, but once the WebLogic Server instance is running, this user cannot change the listen port value.

For more information about `weblogic.Server`, see [“weblogic.Server Command-Line Reference”](#) in the *WebLogic Server Command Reference*.

Permissions for Using the Node Manager

The Node Manager uses both MBeans and the security policy on the Server resource to start a remote server.

If you configure a Node Manager on the host machine of a remote WebLogic Server instance, by default a user in the `Admin` or `Operator` global role can use the Node Manager to start the remote server.

For more information about the Node Manager, see [“Configuring, Starting, and Stopping Node Manager”](#) in *Configuring and Managing WebLogic Server*.

Shutting Down a WebLogic Server Instance

Shutting down a WebLogic Server instance involves both MBeans and the security policy on the Server resource. When a user issues a shutdown command, the server first determines whether that user is granted the `Admin` or `Operator` global role (per the MBean protection). Then, after the MBean operations run, the server determines whether the security policy on the Server resource authorizes the user to shut down the server.

For more information about shutting down a WebLogic Server instance, see [“Starting and Stopping Servers: Quick Reference”](#) in *Configuring and Managing WebLogic Server*.

URL (Web) and EJB (Enterprise JavaBean) Resources

A **URL (Web) resource** is a specific type of WebLogic resource that is related to Web Applications. To secure Web Applications, you create security policies and security roles for a WAR (Web Application Archive) file or for individual components of a Web Application (such as servlets and JSPs). An **EJB (Enterprise JavaBean) resource** is a specific type of WebLogic resource that is related to EJBs. To secure EJBs, you create security policies and security roles for EJB JARs, individual EJBs within an EJB JAR, or for individual methods on an EJB.

Because the Java 2 Enterprise Edition (J2EE) platform standardizes Web Application and EJB security in deployment descriptors, WebLogic Server integrates this standard mechanism with its Security Service to give you a choice of techniques for securing URL and EJB resources. The technique you choose will affect the procedure you will follow, and will require different prerequisite settings in the WebLogic Server Administration Console. For more information, see [“Techniques for Securing URL and EJB Resources” on page 2-13](#) and [“Prerequisites for Securing URL and EJB Resources” on page 2-14](#), respectively.

Note: The instructions for EJB resources provided in this document also apply to Message-Driven Beans (MDBs).

Techniques for Securing URL and EJB Resources

The following sections describe the different techniques for securing URL (Web) and EJB (Enterprise JavaBean) resources in more detail:

- [“Using the WebLogic Server Administration Console” on page 2-13](#)
- [“Using Deployment Descriptors” on page 2-13](#)
- [“Using the Administration Console and Deployment Descriptors” on page 2-14](#)

Using the WebLogic Server Administration Console

The primary benefit of using the Administration Console to secure URL and EJB resources is unified security management. 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 using the Administration Console. As a result, the process of making changes based on updated security requirements becomes more efficient.

You can secure all types of WebLogic resources using this technique. Therefore, the instructions for securing WebLogic resources contained within this document are written specifically for users of the Administration Console.

Using Deployment Descriptors

The primary benefit of securing your URL and EJB resources through J2EE and WebLogic deployment descriptors is that it is a widely known, standard technique for adding declarative security to Web Applications and EJBs. It may also be the technique with which most organizations are familiar. When using this technique, security roles and security policies are

specified in the `web.xml`, `weblogic.xml` and `ejb-jar.xml`, `weblogic-ejb-jar.xml` deployment descriptors.

Note: In WebLogic Server 7.0 SP02, a special tag called `<externally-defined>` was introduced. This tag allows you to specify on a role-by-role basis whether a security role mapping is defined in the deployment descriptors or in the Administration Console. This tag replaces the `<global-role>` tag, which has been deprecated. For more information about using this tag with URL or EJB resources, see [“Using the `<externally-defined>` Tag with Web Applications”](#) or [“Using the `<externally-defined>` Tag with EJBs”](#) in *Programming WebLogic Security*, respectively.

You can secure only URL and EJB resources through use of deployment descriptors. The instructions for using deployment descriptors are available in the [“Using Declarative Security With Web Applications”](#) and [“Using Declarative Security With EJBs”](#) sections of *Programming WebLogic Security*, respectively.

Using the Administration Console and Deployment Descriptors

Some organizations currently using deployment descriptors to secure their URL and EJB resources may want to take advantage of the unified security management capabilities that the WebLogic Server Administration Console provides. In a situation like this, the Administration Console can be instructed to copy security configurations from existing deployment descriptors upon the initial deployment of a Web Application or EJB module. Once these security configurations are copied into the Role Mapping and Authorization providers’ databases, the Administration Console must be used for subsequent updates to security roles and security policies. It is also possible to use this combined technique to reinitialize security configurations for URL and EJB resources to the state specified in the deployment descriptors.

Caution: When using the combined technique, it is possible to override security configurations for URL and EJB resources. Therefore, administrators using the combined technique need to take extra care to ensure that the appropriate security configuration is in place for their Web Applications and EJBs. Read [“Prerequisites for Securing URL and EJB Resources”](#) on page 2-14 for important information.

For more information about using the combined technique, see [“Tutorial 16: Copying and Reinitializing Security Configurations.”](#)

Prerequisites for Securing URL and EJB Resources

Whether using the WebLogic Server Administration Console or deployment descriptors to secure your URL and EJB resources, there are two important settings in WebLogic Server that you need

to understand: the Check Security Roles and Policies setting and the On Future Redeploys setting. Failure to understand these settings could result in incorrect or lost security configurations.

Before attempting to secure URL or EJB resources, read the following sections:

- “Understanding How to Check Security Roles and Security Policies” on page 2-15
- “Understanding What to Do on Future Redeploys of the WebLogic Resource” on page 2-17
- “How to Change the Check Roles and Policies and Future Redeploys Settings” on page 2-17
- “Understanding How These Settings Interact” on page 2-18

Understanding How to Check Security Roles and Security Policies

To give you control over performance, the WebLogic Server Administration Console requires you to specify how the WebLogic Security Service should perform security checks. You specify this preference through the Check Roles and Policies drop-down menu highlighted in [Figure 2-3](#).

Figure 2-3 Administration Console: Security > Realms > myRealm > General

The screenshot shows the 'General' tab of the 'myrealm' configuration in the WebLogic Administration Console. The 'Name' is 'myrealm'. Below it, a description states: 'The name of this configuration.' The 'Check roles and policies for:' dropdown menu is set to 'web applications and EJBs protected in DD'. The 'On future redeploys:' dropdown menu is set to 'initialize roles and policies from DD'. Below these, there are two unchecked checkboxes: 'Ignore Deploy Credential Mapping' and 'Use Deprecated Web Resource'. Each checkbox has a description. The 'Apply' button is at the bottom right.

When the value of the Check Roles and Policies setting is Web Applications and EJBs Protected in DD, the WebLogic Security Service *only* performs security checks on URL (Web) and EJB resources that have security specified in their associated deployment descriptors (DDs). This is the default Check Roles and Policies setting.

When the value of the Check Roles and Policies setting is All Web Applications and EJBs, the WebLogic Security Service performs security checks on *all* URL (Web) and EJB resources, regardless of whether there are any security settings in the deployment descriptors (DDs) for these WebLogic resources. If you change the value of the Check Roles and Policies drop-down menu to All Web Applications and EJBs, you also need to specify what the WebLogic Security Service should do when the Web Application or EJB module is redeployed. For more information, see [“Understanding What to Do on Future Redeploys of the WebLogic Resource” on page 2-17](#).

Note: The Check Roles and Policies setting affects all the WebLogic Server instances (servers) in the WebLogic Server domain in which it is set.

Using the fullyDelegateAuthorization Flag

You can also specify how to perform security checks on URL (Web) and EJB resources using the `fullyDelegateAuthorization` flag, a command-line argument that you set when you start WebLogic Server. When the value of the `fullyDelegateAuthorization` flag is false, the WebLogic Security Service only performs security checks on URL and EJB resources that have security specified in their associated deployment descriptors.

You can set the `fullyDelegateAuthorization` flag in one of three ways:

- Type:

```
-Dweblogic.security.fullyDelegateAuthorization = boolean_value
```

where *boolean_value* is true or false on the command line each time you start a WebLogic Server instance.

Note: You should ensure that the `fullyDelegateAuthorization` flag is set the same way for both your Administration and Managed Servers.

- Edit the `startWLS` script (located in the `WL_HOME\server\bin` directory) to include the following:

```
-Dweblogic.security.fullyDelegateAuthorization = boolean_value
```

where *boolean_value* is true or false. This is more efficient because it will save you from having to set the flag each time you start a WebLogic Server instance. However, keep in mind that this setting will apply to all WebLogic Server domains in the installation.

- Edit the `startWebLogic` script (located in the `WL_HOME\user_projects\domains\mydomain` directory, where *mydomain* is the name of a WebLogic Server domain you created) to include the following in the `JAVA_OPTIONS` section:

```
set JAVA_OPTIONS=...
-Dweblogic.security.fullyDelegateAuthorization=true
```

This method allows you to set the `fullyDelegateAuthorization` flag for each WebLogic Server domain, rather than all the domains in the installation.

Note: If you are using Node Manager to start your Managed Servers, the start scripts previously described are not used. Therefore, you will need to set the `fullyDelegateAuthorization` flag using the WebLogic Server Administration Console.

Understanding What to Do on Future Redeploys of the WebLogic Resource

If you decide that the WebLogic Security Service should perform security checks on All Web Applications and EJBs in the Check Roles and Policies drop-down menu, you also need to tell WebLogic Server which technique you want to use to secure these URL (Web) and EJB resources. (See [“Techniques for Securing URL and EJB Resources” on page 2-13](#) for more information.) You specify this preference through the Future Redeploys drop-down menu highlighted in [Figure 2-3](#).

Set the value of the Future Redeploys drop-down menu as follows:

- To secure your URL (Web) and EJB resources using *only* the WebLogic Server Administration Console, select Ignore Roles and Policies From DD (Deployment Descriptors) and follow steps 3a - 3c in [“Securing WebLogic Resources: Main Steps” on page 1-3](#).
- To secure your URL (Web) and EJB resources using *only* the deployment descriptors (that is, the `web.xml`, `weblogic.xml`, `ejb-jar.xml`, and `weblogic-ejb-jar.xml` files), select Initialize Roles and Policies from DD and refer to the [“Adding Declarative Security to Web Applications”](#) and [“Adding Declarative Security to EJBs”](#) sections of *Programming WebLogic Security*, respectively.

Warning: Switching the value of the Future Redeploys setting is risky and can lead to incorrect or lost security configurations. If you need to switch between these settings (specifically for the situations described in [“Using the Administration Console and Deployment Descriptors” on page 2-14](#)), you can *carefully* follow the instructions in [“Using the Combined Technique to Secure Your URL and EJB Resources” on page 2-19](#).

How to Change the Check Roles and Policies and Future Redeploys Settings

To change the values of the Check Roles and Policies and Future Redeploys settings:

1. In the left pane of the WebLogic Server Administration Console, expand Security →Realms.
2. Click the name of a security realm for which you are setting this option (for example, myrealm).
3. On the General tab, change the value of the Check Roles and Policies drop-down menu.
4. If you changed the value of the Check Roles and Policies drop-down menu to All Web Applications and EJBs, change the value of the Future Redeploys drop-down menu.
5. Click Apply to save your changes.
6. If you changed the value of the Check Roles and Policies drop-down menu, restart the server.

Understanding How These Settings Interact

Table 2-1 shows how to achieve the behavior you want from the WebLogic Security Service using different combinations of the Check Roles and Policies and Future Redeploys settings.

Table 2-1 Interaction Between Check Roles and Policies Setting and On Future Redeploys Setting

If you want to perform security checks on...	and set security for URL (Web) and EJB resources...	then set Check Roles and Policies to...	and set Future Redeploys to...
All URL (Web) and EJB resources	using <i>only</i> the Administration Console	All Web Applications and EJBs	Ignore Roles and Policies from DD

Table 2-1 Interaction Between Check Roles and Policies Setting and On Future Redeploys Setting

If you want to perform security checks on...	and set security for URL (Web) and EJB resources...	then set Check Roles and Policies to...	and set Future Redeploys to...
<i>All</i> URL (Web) and EJB resources	by <i>copying</i> or <i>reinitializing</i> security data from the deployment descriptors into the configured Authorization and Role Mapping providers' databases when the Web Application or EJB module is deployed, then using one of the other techniques to modify security roles and security policies Note: Security data will be copied/reinitialized <i>each time</i> the Web Application or EJB module is deployed.	All Web Applications and EJBs	Initialize Roles and Policies from DD
<i>Only</i> on URIs and EJB methods that are specified in the deployment descriptors (default configuration)	using <i>only</i> the deployment descriptors	Web Applications and EJBs Protected in DD	--

Using the Combined Technique to Secure Your URL and EJB Resources

As described in [“Techniques for Securing URL and EJB Resources”](#) on page 2-13, you can combine the use of the WebLogic Server Administration Console and J2EE/WebLogic deployment descriptor techniques, and would typically do so for two reasons:

- To copy security configurations from deployment descriptors into the configured Authorization and Role Mapping providers' databases, upon initial deployment of Web Application and EJB modules. This process enables you to use the Administration Console for subsequent modifications to security roles and security policies.
- To reinitialize security configurations for URL and EJB resources to their original state, as specified in the deployment descriptors.

Use of the combined technique for other purposes is not recommended. Before continuing, be sure you have read [“Prerequisites for Securing URL and EJB Resources” on page 2-14](#).

The following sections provide step-by-step instructions for using the combined technique to secure your URL and EJB resources:

- [“Copying Security Configurations” on page 2-20](#)
- [“Reinitializing Security Configurations” on page 2-27](#)

You may also want to review [“Tutorial 16: Copying and Reinitializing Security Configurations”](#) before performing these tasks.

Copying Security Configurations

These instructions are intended for administrators who presently secure URL (Web) and EJB (Enterprise JavaBean) resources using J2EE and WebLogic deployment descriptors, but want to exclusively use the WebLogic Server Administration Console from this point forward. Note that BEA does **not** recommend maintaining security configurations in both the deployment descriptors and the Administration Console.

Caution: When using the combined technique, it is possible to override security configurations for URL and EJB resources. Therefore, you must take extra care to ensure that the appropriate security configuration is in place. Follow these instructions carefully to prevent data loss and to ensure that your URL and EJB resources are secured properly.

To copy security configurations for a URL or EJB resource so that you can use the WebLogic Server Administration Console for subsequent modifications, follow these steps:

- [“Step 1: Modify the Security Realm Settings and Deploy the Resource” on page 2-20](#)
- [“Step 2: Verify the Copied Security Policies \(Optional\)” on page 2-21](#)
- [“Step 3: Verify the Copied Security Roles \(Optional\)” on page 2-23](#)
- [“Step 4: Revert the On Future Redeploys Setting” on page 2-26](#)
- [“Step 5: Modify Security Roles and Security Policies Using the Administration Console \(Optional\)” on page 2-27](#)

Step 1: Modify the Security Realm Settings and Deploy the Resource

1. In the left pane of the Administration Console, expand Security → Realms.

2. Click the name of your security realm (for example, `myrealm`).
3. On the General tab, select All Web Applications and EJBs as the value for the Check Roles and Policies drop-down menu.

You are telling WebLogic Server that you want the WebLogic Security Service to perform security checks on *all* URL (Web) and EJB resources. See [“Understanding How to Check Security Roles and Security Policies” on page 2-15](#).

Note: If All Web Applications and EJBs was already selected as the value of the Check Roles and Policies drop-down menu, just continue to step 4.

4. Select Initialize Roles and Policies From DD as the value for the On Future Redeploys drop-down menu.

You are telling WebLogic Server to *copy* security for URL (Web) and EJB resources from the deployment descriptors into the configured Authorization and Role Mapping providers’ databases *each time you deploy the resource*. See [“Understanding What to Do on Future Redeploys of the WebLogic Resource” on page 2-17](#).

5. Click Apply to save your changes.
6. If you had to set the Check Roles and Policies drop-down menu to All Web Applications and EJBs in step 2 (that is, it was *not* already set this way), restart the server. (For help, see [“Starting and Stopping WebLogic Servers: Quick Reference”](#) in the *WebLogic Server Administration Guide*.)

Note: If you did not have to modify the value of the Check Role and Policies drop-down menu in step 3, continue to step 7 *without restarting the server*.

7. Deploy the Web Application or EJB module whose security configuration you want to copy, targeting it to the appropriate server.

For instructions about how to deploy Web Application and EJB modules, see [Deploying WebLogic Server Applications](#).

Step 2: Verify the Copied Security Policies (Optional)

To verify the copied security policies, follow the instructions shown in the appropriate column of [Table 2-2](#).

Table 2-2 Verifying Copied Security Policies, Based on Resource Type

Step	URL (Web) Resources	EJB Resources
1	Open the <code>web.xml</code> deployment descriptor for the Web Application, and record the content of any <code><url-pattern></code> and <code><http-method></code> elements, as well as any <code><role-name></code> subelements of the <code><auth-constraint></code> element.	Open the <code>ejb-jar.xml</code> deployment descriptor for the EJB, and record the content of any <code><method-permission></code> elements, specifically focusing on the <code><role-name></code> , <code><ejb-name></code> , and <code><method-name></code> subelements. Note: If the deployment descriptor uses the <code><unchecked /></code> element where you would normally find a <code><role-name></code> element, security checks will not be performed on that method; therefore, no security data for that method will be copied.
2	Using the navigation tree at the left side of the Administration Console, right-click the name of the deployed Web Application module.	Using the navigation tree at the left side of the Administration Console, right-click the name of the deployed EJB module.
3	Select the Define Security Policy... option from the menu.	Select the Define Roles and Policies for Individual Beans... option from the menu. A table listing all the EJBs that are in the JAR file appears.
4	On the General tab, click the hyperlinked URL pattern that corresponds to the content of a single <code><url-pattern></code> element you recorded in step 1.	Click the [Define Security Policies] link for the EJB that corresponds to the <code><ejb-name></code> element you recorded in step 1.

Table 2-2 Verifying Copied Security Policies, Based on Resource Type (Continued)

Step	URL (Web) Resources	EJB Resources
5	<p>On the Policy Editor page that appears, select a method from the Methods drop-down menu that corresponds to the content of a <code><http-method></code> element you recorded in step 1.</p> <p>The Caller is Granted the Role condition in the Policy Condition list box is highlighted, and the content of the Policy Statement list box corresponds to the content of the appropriate <code><role-name></code> element that you recorded in step 1.</p>	<p>On the Policy Editor page that appears, select a method from the Methods drop-down menu that corresponds to the content of a <code><method-name></code> element you recorded in step 1.</p> <p>The Caller is Granted the Role condition in the Policy Condition list box is highlighted, and the content of the Policy Statement list box corresponds to the content of the corresponding <code><role-name></code> element that you recorded in step 1.</p>
6	Repeat steps 1- 5 to verify multiple security policies.	Repeat steps 1- 5 to verify multiple security policies.

Step 3: Verify the Copied Security Roles (Optional)

To verify the copied security policies, follow the instructions shown in the appropriate column of [Table 2-3](#).

Table 2-3 Verifying Copied Security Roles, Based on Resource Type

Step	URL (Web) Resources	EJB Resources
1	<p>Open the <code>weblogic.xml</code> deployment descriptor for the Web Application, and record the content of any <code><security-role-assignment></code> elements, specifically focusing on the <code><role-name></code> and <code><principal-name></code> subelements.</p> <p>Note: If the deployment descriptor uses the <code><externally-defined></code> element for a Web Application, no scoped roles are actually defined; therefore no scoped roles for the EJB can be copied.</p>	<p>Open the <code>weblogic-ejb-jar.xml</code> deployment descriptor for the EJB, and record the content of any <code><security-role-assignment></code> elements, specifically focusing on the <code><role-name></code> and <code><principal-name></code> subelements.</p> <p>Note: If the deployment descriptor uses the <code><externally-defined></code> element for an EJB, no scoped roles are actually defined; therefore no scoped roles for the EJB can be copied.</p>
2	Using the navigation tree at the left side of the Administration Console, right-click the name of the deployed Web Application module.	Using the navigation tree at the left side of the Administration Console, right-click the name of the deployed EJB module.
3	Select the Define Scoped Role... option from the menu.	<p>Select the Define Scoped Role... option from the menu.</p> <p>The Select Roles page displays all the scoped roles for this EJB that are currently defined in the WebLogic Role Mapping provider's database, including the ones from your deployment descriptor's <code><role-name></code> element.</p>

Table 2-3 Verifying Copied Security Roles, Based on Resource Type (Continued)

Step	URL (Web) Resources	EJB Resources
4	<p>On the General tab, click the hyperlinked URL pattern <code>/*</code>.</p> <p>Note: Security roles obtained from deployment descriptors are always copied into the configured Role Mapping provider's database as scoped roles, with an URL pattern of <code>/*</code>.</p> <p>The Select Roles page displays all the scoped roles for this Web Application that are currently defined in the WebLogic Role Mapping provider's database, including the ones from your deployment descriptor's <code><role-name></code> element.</p>	<p>Click the hyperlinked name of the scoped role.</p>
5	<p>Click the hyperlinked name of the scoped role.</p>	<p>Select the Conditions tab.</p> <p>The Role Statement list box contains a Role Statement based on the content of your deployment descriptor's corresponding <code><principal-name></code> element.</p> <p>Note: Because principals can be users or groups, the Role Statement list box will show two expressions: one using the contents of the <code><principal-name></code> element in the User Name of the Caller Role Condition, the other using it in a Caller is a Member of the Group Role Condition, linked by an <code>or</code> statement. The Administration Console presumes that a user or group of the name used in the deployment descriptor already exists. If they do not, you will need to create them.</p>

Table 2-3 Verifying Copied Security Roles, Based on Resource Type (Continued)

Step	URL (Web) Resources	EJB Resources
6	<p>Select the Conditions tab.</p> <p>The Role Statement list box contains a Role Statement based on the content of your deployment descriptor's corresponding <principal-name> element.</p> <p>Note: Because principals can be users or groups, the Role Statement list box will show two expressions: one using the contents of the <principal-name> element in the User Name of the Caller Role Condition, the other using it in a Caller is a Member of the Group Role Condition, linked by an or statement. The Administration Console presumes that a user or group of the name used in the deployment descriptor already exists. If they do not, you will need to create them.</p>	<p>Repeat steps 1- 5 to verify multiple scoped roles.</p>
7.	<p>Repeat steps 1- 6 to verify multiple scoped roles.</p>	--

Step 4: Revert the On Future Redeploys Setting

Caution: You must perform this step. Failure to revert this setting may result in inconsistent security configurations when your Web Application and EJB modules are redeployed. Therefore, be sure to perform this step *before* you restart your server. If you do not perform this step or perform this step incorrectly, you will see the following message the next time you load the Policy Editor page:

The information presented below may not be accurate. To ensure that you are viewing accurate information, you may need to delete and redeploy your WebLogic resources.

1. In the left pane of the Administration Console, expand Security →Realms.
2. Click the name of your security realm (for example, myrealm).

3. On the General tab, select Ignore Roles and Policies From DD as the value for the On Future Redeploys drop-down menu.

You are telling WebLogic Server that you will set security for URL (Web) and EJB resources using the Administration Console, not deployment descriptors. See [“Understanding What to Do on Future Redeploys of the WebLogic Resource” on page 2-17](#).

4. Click Apply to save your changes.

Step 5: Modify Security Roles and Security Policies Using the Administration Console (Optional)

Follow the instructions in [“Modifying Global Roles” on page 4-17](#) and [“Working with Security Policies” on page 5-7](#) to modify your URL or EJB resource’s security roles and security policies.

Reinitializing Security Configurations

To reinitialize security configurations for URL (Web) and EJB (Enterprise JavaBean) resources to their original state as specified in their deployment descriptors, follow these steps:

- [“Step 1: Modify the Security Realm Settings and Redeploy the WebLogic Resource” on page 2-27](#)
- [“Step 2: Verify the Reinitialized Security Policies and Security Roles \(Optional\)” on page 2-28](#)
- [“Step 3: Revert the On Future Redeploys Setting” on page 2-29](#)
- [“Step 4: Modify Security Roles and Security Policies Using the Administration Console \(Optional\)” on page 2-29](#)

Step 1: Modify the Security Realm Settings and Redeploy the WebLogic Resource

1. In the left pane of the Administration Console, expand Security → Realms.
2. Click the name of your security realm (for example, `myrealm`).
3. If All Web Applications and EJBs was already selected as the value of the Check Roles and Policies drop-down menu, skip step 4.
4. On the General tab, select All Web Applications and EJBs as the value for the Check Roles and Policies drop-down menu.

You are telling WebLogic Server that you want the WebLogic Security Service to perform security checks on *all* URL (Web) and EJB resources. See [“Understanding How to Check Security Roles and Security Policies” on page 2-15](#).

5. Select Initialize Roles and Policies From DD as the value for the On Future Redeploys drop-down menu.

You are telling WebLogic Server to *copy* security for URL (Web) and EJB resources from the deployment descriptors into the configured Authorization and Role Mapping providers’ databases *each time you deploy the resource*. See [“Understanding What to Do on Future Redeploys of the WebLogic Resource” on page 2-17](#).

6. Click Apply to save your changes.
7. In the left pane of the Administration Console, expand Deployments, then click either:
 - Web Application Modules—for URL (Web) resources, or
 - EJB Modules—for EJB (Enterprise JavaBean) resources.

8. Click the name of a Web Application or EJB module.

A table that lists all the Web Application or EJB modules appears in the right pane.

9. Click the trash can icon that is located in the same row as the Web Application or EJB module for which you want to reinitialize a security configuration.

10. Click Yes, then the Continue link to delete the Web Application or EJB module.

The deleted Web Application or EJB module no longer appears in the table.

11. Re-deploy the Web Application or EJB module whose security configuration you want to initialize, targeting it to the appropriate server.

For instructions about how to deploy Web Application and EJB modules, see [Deploying WebLogic Server Applications](#).

Step 2: Verify the Reinitialized Security Policies and Security Roles (Optional)

To verify the reinitialized security policies and security roles, follow the instructions shown in the appropriate column of [Table 2-2, “Verifying Copied Security Policies, Based on Resource Type,” on page 2-22](#) and [Table 2-3, “Verifying Copied Security Roles, Based on Resource Type,” on page 2-24](#), respectively.

Step 3: Revert the On Future Redeploys Setting

Caution: You must perform this step. Failure to revert this setting may result in inconsistent security configurations when your Web Application and EJB modules are redeployed. Therefore, be sure to perform this step *before* you restart your server. If you do not perform this step or perform this step incorrectly, you will see the following message the next time you load the Policy Editor page:

The information presented below may not be accurate. To ensure that you are viewing accurate information, you may need to delete and redeploy your WebLogic resources.

1. In the left pane of the Administration Console, expand Security → Realms.
2. Click the name of your security realm (for example, `myrealm`).
3. On the General tab, select Ignore Roles and Policies From DD as the value for the On Future Redeploys drop-down menu.

You are telling WebLogic Server that you will set security for URL (Web) and EJB resources using the Administration Console, not deployment descriptors. See [“Understanding How to Check Security Roles and Security Policies” on page 2-15](#).

4. Click Apply to save your changes.

Step 4: Modify Security Roles and Security Policies Using the Administration Console (Optional)

Follow the instructions in [“Modifying Global Roles” on page 4-17](#) and [“Working with Security Policies” on page 5-7](#) to modify your URL (Web) or EJB resource’s security roles and security policies.

Web Service Resources

A WebLogic Web Service is typically packaged as an Enterprise Application that contains a special type of Web Application that includes an additional deployment descriptor called `web-services.xml`. If your Web Service is implemented with a Java class, then the Web Application WAR file contains the Java class files. If the Web Service is implemented with a stateless session EJB, then the Enterprise Application EAR file contains the corresponding EJB JAR file.

Note: A Web Service can also be packaged as a stand-alone Web Application WAR file if the Web Service is implemented with just a Java class. This type of packaging for a Web Service is uncommon, however; typically Web Services are packaged as EAR files.

A **Web Service resource** is a specific type of WebLogic resource that is related to Web Services. To secure Web Services, you can create security policies and security roles for:

- The entire Web Service
- A subset of the operations of the Web Service
- The Web Service URL
- The stateless session EJB that implements the Web Service
- A subset of the methods of the stateless session EJB
- The WSDL and Home Page of the Web Service

For more information about WebLogic Web Services, see [Programming WebLogic Web Services](#).

Users and Groups

The following sections describe the features and functions of users and groups:

- [“Overview of Users and Groups” on page 3-2](#)
- [“Creating Users” on page 3-2](#)
- [“Adding Users to Groups” on page 3-3](#)
- [“Modifying Users” on page 3-4](#)
- [“Deleting Users” on page 3-4](#)
- [“Default Groups” on page 3-5](#)
- [“Creating Groups” on page 3-6](#)
- [“Nesting Groups” on page 3-7](#)
- [“Modifying Groups” on page 3-7](#)
- [“Deleting Groups” on page 3-8](#)

Note: For information about how to perform administrative tasks related to users and groups using the `weblogic.Admin` command-line utility (rather than the WebLogic Server Administration Console GUI), see [“Using weblogic.Admin Commands to Manage Users and Groups”](#) in *WebLogic Server Command Reference*.

Overview of Users and Groups

A **user** is an entity that can be authenticated. A user can 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 adding users to groups. A **group** is a collection of users who usually have something in common, such as working in the same department in a company.

Creating Users

Notes: The instructions in this section apply to the WebLogic Authentication provider only. If you customize the default security configuration to use a custom Authentication provider, you must use the administration tools supplied by that security provider to create a user.

When upgrading to the WebLogic Authentication provider, you cannot automatically load existing users into the WebLogic Authentication provider's database. For this release of WebLogic Server, adding existing users is a manual step. If you have many existing users, consider using the Realm Adapter Authentication provider. See [“Configuring a Realm Adapter Authentication Provider”](#) in *Managing WebLogic Security*.

To create a new user:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm for which you are creating a user (for example, myrealm).
3. Click Users.

The Users page displays all the users currently defined in the WebLogic Authentication provider's database.

4. Click the Configure a new User... link to display the Create User page.

Note: If multiple WebLogic Authentication providers are configured in the security realm, an intermediate page will list them in a table. From the table, select which WebLogic Authentication provider's database should store information for the new user before performing step 5.

5. On the General tab, enter the name of the user in the Name field.

Do not use commas or any other characters in this comma-separated list: \t, <, >, #, |, &, ?, (), { }. User names are case sensitive.

6. Optionally, enter a description of the user (such as their full name) in the Description field.

7. Enter a password for the user in the Password field.

The minimum password length for a user defined in the WebLogic Authentication provider is 8 characters. Do not use the user name/password combination `weblogic/weblogic` in a production environment.

8. Re-enter the password for the user in the Confirm Password field.
9. Click Apply to save your changes.

Adding Users to Groups

BEA recommends adding users to groups because groups allow you to manage a number of users at the same time. This is generally more efficient than managing each user individually.

In the procedure that follows, it is assumed that you have already created groups as described in [“Creating Groups” on page 3-6](#), or that you will use the default groups described in [“Default Groups” on page 3-5](#).

To add a user to a group:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm for which you are adding a user to a group (for example, `myrealm`).
3. Click Users.

The Users page displays all the users currently defined in the WebLogic Authentication provider’s database.

4. Click the hyperlinked name of the user that you want to add to a group.

If you have many users, use the Filter By field at the top of the page to retrieve and list only the users that match your search criteria, then click the hyperlinked name. The Filter By field uses the asterisk (*) as the wildcard character.

5. Select the Groups tab.

All the groups available in the WebLogic Authentication provider’s database appear in the Possible Groups list box. All the groups to which the user belongs appear in the Current Groups list box.

6. In the Possible Groups list box, highlight the name of a group.

7. Click the highlighted arrow to move the group from the Possible Groups list box to the Current Groups list box.
8. If desired, repeat steps 6 and 7 to add the user to multiple groups.
9. Click Apply to save your changes.

Modifying Users

To modify an existing user:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm for which you are modifying a user (for example, `myrealm`).
3. Click Users.

The Users page displays all the users currently defined in the WebLogic Authentication provider's database.

4. Click the hyperlinked name of the user that you want to modify.

If you have many users, use the Filter By field at the top of the page to retrieve and list only the users that match your search criteria, then click the hyperlinked name. The Filter By field uses the asterisk (*) as the wildcard character.

5. Use the General tab to modify the user's description or password, and the Groups tab to modify the user's membership in one or more groups. (See [“Creating Users” on page 3-2](#) and [“Adding Users to Groups” on page 3-3](#) for specific instructions.)

Note: On both tabs, click Apply to save your changes.

Deleting Users

To delete an existing user:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm from which you are deleting a user (for example, `myrealm`).
3. Click Users.

The Users page displays all the users currently defined in the WebLogic Authentication provider's database.

4. Click the trash can icon that is located in the same row as the user you want to delete.

If you have many users, use the Filter By field at the top of the page to retrieve and list only the users that match your search criteria, then click the trash can icon. The Filter By field uses the asterisk (*) as the wildcard character.

5. Click Yes to confirm the deletion.
6. Click Continue.

The Users page no longer shows the deleted user in the table.

Default Groups

By default, WebLogic Server defines the groups shown in [Table 3-1](#).

Table 3-1 Default Groups

Group Name	Membership
users	Users, when they log in (for example, through a Web page). The <code>users</code> group includes all users except the <code><anonymous></code> user. See “ Guest and <anonymous> Users ” in the <i>WebLogic Server 8.1 Upgrade Guide</i> .
everyone	Every user is a member of this group. The <code>users</code> group is nested within the <code>everyone</code> group.
Administrators	By default, this group contains the user information entered as part of the installation process (that is, the Configuration Wizard), and the <code>system</code> user if the WebLogic Server instance is running Compatibility security. Any user assigned to the <code>Administrators</code> group is granted the <code>Admin</code> security role by default.
Deployers	By default, this group is empty. Any user assigned to the <code>Deployers</code> group is granted the <code>Deployer</code> security role by default.
Operators	By default, this group is empty. Any user assigned to the <code>Operators</code> group is granted the <code>Operator</code> security role by default.
Monitors	By default, this group is empty. Any user assigned to the <code>Monitors</code> group is granted the <code>Monitor</code> security role by default.

For more information about the default security roles, see “[Default Global Roles](#)” on page 4-5.

You can add to the default groups by creating your own, as described in [“Creating Groups” on page 3-6](#).

Creating Groups

Notes: The instructions in this section apply to the WebLogic Authentication provider only. If you customize the default security configuration to use a custom Authentication provider, you must use the administration tools supplied by that security provider to create a group.

When upgrading to the WebLogic Authentication provider, you cannot automatically load existing groups into the WebLogic Authentication provider’s database. For this release of WebLogic Server, adding existing groups is a manual step. If you have many existing groups, consider using the Realm Adapter Authentication provider. See [“Configuring a Realm Adapter Authentication Provider”](#) in *Managing WebLogic Security*.

To create a new group:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm for which you are creating a group (for example, myrealm).
3. Click Groups.

The Groups page displays all the groups currently defined in the WebLogic Authentication provider’s database.

4. Click the Configure a new Group... link to display the Create Group page.

Note: If multiple WebLogic Authentication providers are configured in the security realm, an intermediate page will list them in a table. From the table, select which WebLogic Authentication provider’s database should store information for the new group before performing step 5.

5. On the General tab, enter the name of the group in the Name field.

Do not use commas or any other characters in this comma-separated list: \t, <>, #, |, &, ?, (), { }. Group names are case sensitive. Group names are plural, according to the BEA convention.

6. Optionally, enter a description of the group in the Description field.
7. Click Apply to save your changes.

Nesting Groups

Optionally, you can nest groups within other groups.

Note: In the procedure that follows, it is assumed that you have already created groups as described in [“Creating Groups” on page 3-6](#) *or* that you will use the default groups described in [“Default Groups” on page 3-5](#).

To nest a group within another group:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm for which you are nesting a group (for example, `myrealm`).
3. Click Groups.

The Groups page displays all the groups currently defined in the WebLogic Authentication provider's database.

4. Click the hyperlinked name of the group that you want to nest within another group.

If you have many groups, use the Filter By field at the top of the page to retrieve and list only the groups that match your search criteria, then click the hyperlinked name. The Filter By field uses the asterisk (*) as the wildcard character.

5. Select the Membership tab.

All the groups available in the WebLogic Authentication provider's database appear in the Possible Groups list box. All the groups in which the group is nested appear in the Current Groups list box.

6. In the Possible Groups list box, highlight the name of a group.
7. Click the highlighted arrow to move the group from the Possible Groups list box to the Current Groups list box.
8. If desired, repeat steps 6 and 7 to nest the group within multiple groups.
9. Click Apply to save your changes.

Modifying Groups

To modify an existing group:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm for which you are modifying a group (for example, `myrealm`).

3. Click Groups.

The Groups page displays all the groups currently defined in the WebLogic Authentication provider's database.

4. Click the hyperlinked name of the group that you want to modify.

If you have many groups, use the Filter By field at the top of the page to retrieve and list only the groups that match your search criteria, then click the hyperlinked name. The Filter By field uses the asterisk (*) as the wildcard character.

5. Use the General tab to modify the group's description, and the Membership tab to modify the group's membership in one or more other groups. (See [“Creating Groups” on page 3-6](#) and [“Nesting Groups” on page 3-7](#) for specific instructions.)

Note: On both tabs, click Apply to save your changes.

Deleting Groups

To delete an existing group:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm from which you are deleting a group (for example, myrealm).
3. Click Groups.

The Groups page displays all the groups currently defined in the WebLogic Authentication provider's database.

4. Click the trash can icon that is located in the same row as the group you want to delete.

If you have many groups, use the Filter By field at the top of the page to retrieve and list only the users that match your search criteria, then click the trash can icon. The Filter By field uses the asterisk (*) as the wildcard character.

5. Click Yes to confirm the deletion.
6. Click Continue.

The Groups page no longer shows the deleted group in the table.

Security Roles

The following sections describe the features and functions of security roles:

- [“Overview of Security Roles” on page 4-1](#)
- [“Dynamic Role Mapping” on page 4-2](#)
- [“Types of Security Roles: Global Roles and Scoped Roles” on page 4-3](#)
- [“Using the Administration Console to Create Security Roles” on page 4-3](#)
- [“Default Global Roles” on page 4-5](#)
- [“Default Group Associations” on page 4-11](#)
- [“Components of a Security Role: Role Conditions, Expressions, and Role Statements” on page 4-12](#)
- [“Working with Global Roles” on page 4-13](#)
- [“Working with Scoped Roles” on page 4-18](#)

Overview of Security Roles

A **security role** is a privilege granted to users or groups based on specific conditions. Like groups, security roles allow you to restrict access to WebLogic resources for several users at once. Security roles differ from groups as follows:

- 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. Groups are static.

- Security roles 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. For example, an administrator may define a security role called `AppAdmin`, which has write access to a particular Web Application's resources. Any user or group granted the `AppAdmin` security role would then have write access to that URL (Web) resource. Multiple users or groups can be granted a single security role. (For more information about users and groups, see [Chapter 3, “Users and Groups.”](#))

Note: In WebLogic Server 6.x, security roles applied to Web Applications and EJBs (Enterprise JavaBeans) only. This version of WebLogic Server expands the use of security roles to all of the defined WebLogic resources. For more information, see [Chapter 2, “Types of WebLogic Resources.”](#)

Dynamic Role Mapping

At runtime, the WebLogic Security Service compares users or groups against a role condition to determine whether they should be dynamically granted a security role. This process is referred to as **role mapping**, and occurs just prior to when the WebLogic Security Service renders an access decision for a protected WebLogic resource. An access decision is the component of an Authorization provider that determines whether a subject has permission to perform a given operation on a WebLogic resource.

Note: For more information about role conditions and access decisions, see “[Components of a Security Role: Role Conditions, Expressions, and Role Statements](#)” on page 4-12 and “[Access Decisions](#)” in *Developing Security Providers for WebLogic Server*, respectively.

This dynamic mapping of security roles to users or groups provides a very important benefit: users or groups can be granted a security role based on business rules, or the context of the request. For example, a user may be allowed to be in a `Manager` security role only while the actual manager is away. Dynamically granting this security role means that you do not need to change or redeploy your application to allow for such a temporarily arrangement. You simply specify the hours between which the temporary manager should have special privileges. Further, you do not need to remember to revoke these special privileges when the actual manager returns, as you would if you temporarily added the user to a management group.

Types of Security Roles: Global Roles and Scoped Roles

There are two types of security roles in WebLogic Server: global roles and scoped roles. A security role that applies to all WebLogic resources deployed within a security realm (and thus the entire WebLogic Server domain) is called a **global role**. A security role that applies to a specific instance of a WebLogic resource deployed in a security realm (such as a method on an EJB or a branch of a JNDI tree) is called a **scoped role**. Multiple roles (either global or scoped) can be used to create a security policy for a WebLogic resource. (For more information, see [Chapter 5, “Security Policies.”](#))

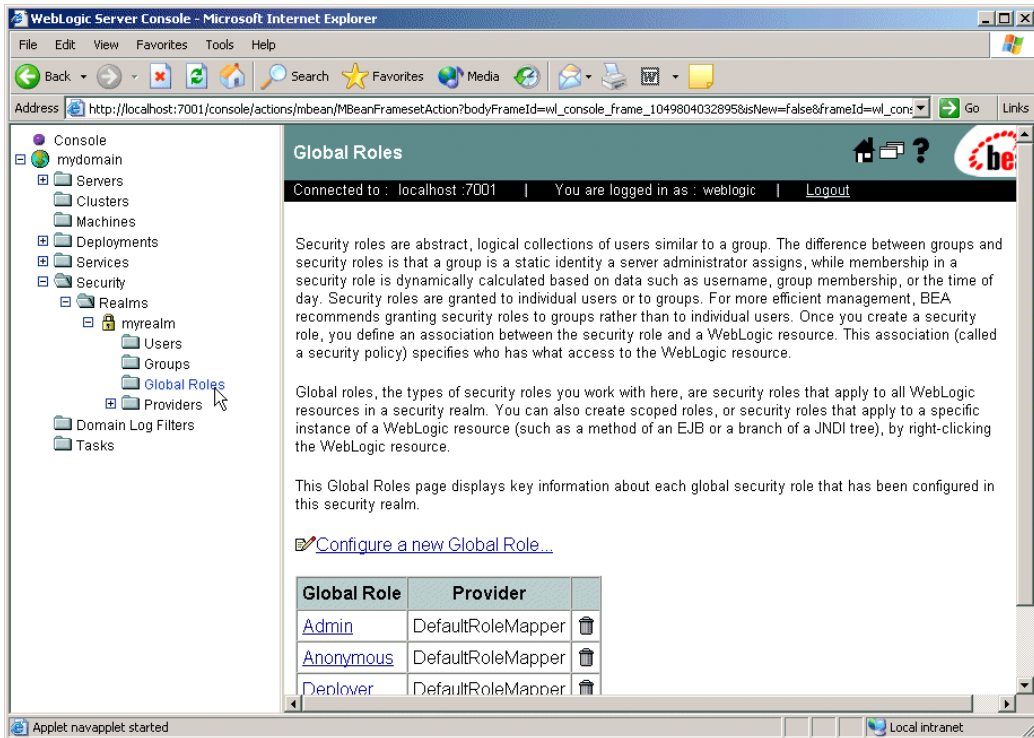
While BEA recommends creating security roles and using them (rather than users or groups) to secure WebLogic resources, you do not need to use a particular type of security role. BEA provides several default global roles that you can use out of the box to secure your WebLogic resources; these are described in [“Default Global Roles” on page 4-5](#). You may never need to use scoped roles. (Scoped roles are provided for their flexibility and are an extra feature for advanced customers.)

Using the Administration Console to Create Security Roles

The way you use the WebLogic Server Administration Console to create security roles differs, depending on whether you want to create a global role or a scoped role.

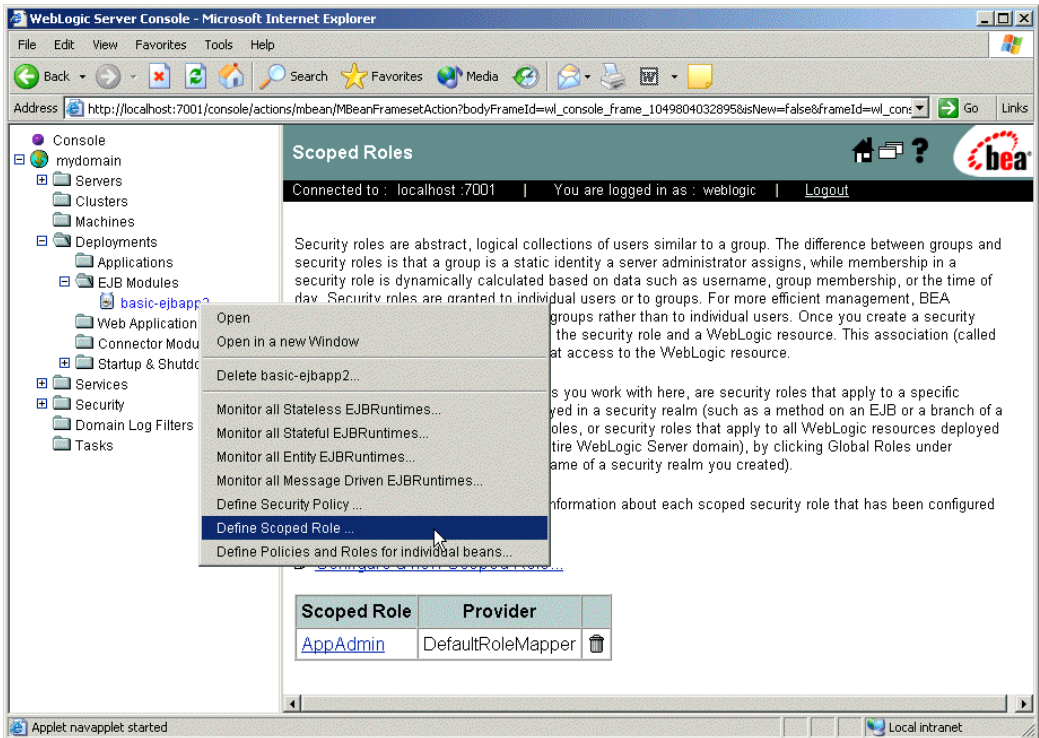
Because global roles apply to all WebLogic resources in a security realm, you create global roles at the security realm level. In the left pane of the Administration Console, expand Security → Realms → `myrealm` (or the name of a security realm you created). Then click Global Roles to display the page that allows you to create a global role. This navigation path is shown on the left side of [Figure 4-1](#), and the resulting page is shown on the right.

Figure 4-1 Creating a Global Role



Because they apply only to a particular WebLogic resource in a security realm, you create scoped roles at the WebLogic resource level. Deployed modules (that is, Web Applications, EJBs, and so on) for which you can create scoped roles show a Define Scoped Role... option when you right-click on them in the Administration Console's navigation tree. Select the Define Scoped Role... option to display the page that allows you to create a scoped role. This navigation path—using the `basic-ejbapp` JAR as the WebLogic resource—is shown on the left side of Figure 4-2, and the resulting page is shown on the right.

Figure 4-2 Creating a Scoped Role



Default Global Roles

By default, WebLogic Server defines the global roles shown in [Table 4-1](#). The table also lists the privileges that users or groups in these security roles are granted.

The default global roles are used in the default security policies that protect most types of WebLogic resources. In addition, the default global roles are used to provide additional security for Server resources that are exposed as MBeans. For more information, see [Chapter 5, “Security Policies,”](#) and [“MBean Protections”](#) on page 2-7.

Table 4-1 Default Global Roles and Their Privileges

Global Role	Privileges
Anonymous	<p>All users (the group <code>everyone</code>) are granted this global role.</p> <p>Note: This global role is provided as a convenience, and can be specified in the <code>weblogic.xml</code> and <code>weblogic-ejb-jar.xml</code> deployment descriptors.</p>
Admin	<ul style="list-style-type: none"> • View the server configuration, <i>including</i> the encrypted value of encrypted attributes. • Modify the entire server configuration. • Deploy enterprise applications, startup and shutdown classes, and Web application, EJB, J2EE Connector, Web Service, and WebLogic Tuxedo Connector components. If applicable, edit deployment descriptors. • Start, resume, and stop servers by default.
Deployer	<ul style="list-style-type: none"> • View the server configuration, <i>except</i> for encrypted attributes. • Change startup and shutdown classes, Web applications, JDBC data pool connections, EJB, J2EE Connector, Web Service, and WebLogic Tuxedo Connector components. If applicable, edit deployment descriptors.
Operator	<ul style="list-style-type: none"> • View the server configuration, <i>except</i> for encrypted attributes. • Start, resume, and stop servers by default.
Monitor	<p>View the server configuration, <i>except</i> for encrypted attributes.</p> <p>This security role effectively provides read-only access to the WebLogic Server Administration Console, <code>weblogic.Admin</code> utility and MBean APIs.</p>

Note: If you are working directly with WebLogic Server MBeans and want more detailed information about the global roles and their privileges than is shown in [Table 4-1](#), see [“Protected MBean Attributes and Operations”](#) on page 4-6.

You can add to the default global roles by creating your own security roles (global or scoped) as described in [“Creating Global Roles”](#) on page 4-14.

Protected MBean Attributes and Operations

[Table 4-2](#) lists the immutable privileges given to users or groups who are granted the `Admin` default global role, for various WebLogic Server MBeans. In other words, users or groups who

are granted the `Admin` default global role have permission to access the MBean attributes listed in [Table 4-2](#).

Note: Users or groups who are granted the `Admin` default global role are also given the privileges described in [Table 4-3](#) through [Table 4-5](#).

Table 4-2 MBean Privileges for the Admin Default Global Role

MBeans	Accessible Attributes
BridgeDestinationCommonMBean	UserPassword
BridgeDestinationMBean	UserPassword
JDBCCConnectionPoolMBean	Password, XAPassword
JDBCDataSourceFactoryMBean	Password
JMSBridgeDestinationMBean	UserPassword
NetworkChannelMBean	DefaultIIOPPassword
NodeManagerMBean	CertificatePassword
SecurityConfigurationMBean	Credential, EncryptedSecretKey, Salt
SecurityMBean	Salt, EncryptedSecretKey
ServerMBean	SystemPassword, DefaultIIOPPassword, DefaultTGIOPPassword, CustomIdentityKeyStorePassPhrase, CustomTrustKeyStorePassPhrase, JavaStandardTrustKeyStorePassPhrase
ServerStartMBean	Password
SSLMBean	ServerPrivateKeyPassPhrase
WLECCConnectionPoolMBean	UserPassword, ApplicationPassword

The MBeans shown in [Table 4-2](#) are all in the `weblogic.management.configuration` package. For more information on MBeans used to configure WebLogic Server, see “[System Administration Infrastructure](#)” in the *WebLogic Server Administration Guide*.

[Table 4-3](#) lists the immutable privileges given to users or groups who are granted the `Admin` or `Deployer` default global roles, for various WebLogic Server MBeans. In other words, users or

groups who are granted the `Admin` or `Deployer` default global roles have permission to access the MBean operations listed in [Table 4-3](#).

Table 4-3 Privileges for the Admin or Deployer Default Global Roles

MBeans	Accessible Operations
Application, ApplicationConfig	All
ConnectorComponent, ConnectorComponentConfig	All
DeployerRuntime, DeploymentTaskRuntime	All
EJBComponent, EJBComponentConfig	All
WebAppComponent, WebAppComponentConfig	All
WebServiceComponent, WebServiceComponentConfig	All
WebServer, WebServerConfig	All
JDBCConnectionPool, JDBCConnectionPoolConfig	All
JDBCDataSourceFactory, JDBCDataSourceFactoryConfig	All
JDBCMultiPool, JDBCMultipoolConfig	All
JDBCDataSource, JDBCDataSourceConfig	All
JDBCTxDataSource, JDBCTxDataSourceConfig	All
JDBCPoolComponent, JDBCPoolComponentConfig	All
JMSBridgeDestination, JMSBridgeDestinationConfig	All
JMSConnectionConsumer, JMSConnectionConsumerConfig	All

Table 4-3 Privileges for the Admin or Deployer Default Global Roles (Continued)

MBeans	Accessible Operations
JMSConnectionFactory, JMSConnectionFactoryConfig	All
JMSDestination, JMSDestinationConfig	All
JMSDistributedDestination, JMSDistributedDestinationConfig	All
JMSDistributedDestinationMember, JMSDistributedDestinationMemberConfig	All
JMSDistributedTopic, JMSDistributedTopicConfig	All
JMSDistributedTopicMember, JMSDistributedTopicMemberConfig	All
JMSDistributedQueue, JMSDistributedQueueConfig	All
JMSDistributedQueueMember, JMSDistributedQueueMemberConfig	All
JMSFileStore, JMSFileStoreConfig	All
JMSDestinationKey, JMSDestinationKeyConfig	All
JMSServer, JMSServerConfig	All
JMSStore, JMSStoreConfig	All
JMSSessionPool, JMSSessionPoolConfig	All
JMSTemplate, JMSTemplateConfig	All
JMSQueue, JMSQueueConfig	All
JMSTopic, JMSTopicConfig	All
JMSJDBCStore, JMSJDBCStoreConfig	All
WTCServer, WTCServerConfig	All

Table 4-3 Privileges for the Admin or Deployer Default Global Roles (Continued)

MBeans	Accessible Operations
WTCBridgeGlobal, WTCBridgeGlobalConfig	All
WTCResources, WTCResourcesConfig	All
WTCEXport, WTCEXportConfig	All
WTCImport, WTCImportConfig	All
WTCLocalTuxDom, WTCLocalTuxDomConfig	All
WTCRemoteTuxDom, WTCRemoteTuxDomConfig	All
WTCPassword, WTCPasswordConfig	All
WTCtBridgeGlobal, WTCtBridgeGlobalConfig	All
WTCtBridgeRedirect, WTCtBridgeRedirectConfig	All
EJBDescriptor, ConnectorDescriptor, WebDescriptor	All
Server	addDeployment, lookupServerLifecycleRuntime, lookupServerRuntime, removeDeployment, sendNotification
ServerConfig	addDeployment, lookupServerLifecycleRuntime, removeDeployment, sendNotification

[Table 4-4](#) lists the immutable privileges given to users or groups who are granted the `Admin` or `Monitor` default global roles, for various WebLogic Server MBeans. In other words, users or groups who are granted the `Admin` or `Monitor` default global roles have permission to access the MBean operations listed in [Table 4-4](#).

Table 4-4 Privileges for the Admin or Monitor Default Global Roles

MBeans	Accessible Operations
Machine	lookupNodeManagerRuntime
NodeManagerRuntime	getStateForAll, register
Server	lookupServerLifeCycleRuntime, lookupServerRuntime

[Table 4-5](#) lists the immutable privileges given to users or groups who are granted the `Admin` or `Operator` default global roles, for various WebLogic Server MBeans. In other words, users or groups who are granted the `Admin` or `Operator` default global roles have permission to access the MBean operations listed in [Table 4-5](#).

Table 4-5 Privileges for the Admin or Operator Default Global Roles

MBeans	Accessible Operations
ServerLifeCycleRuntime	All
ServerLifeCycleTaskRuntime	All
ServerStart	All
Server	ExpectedToRun, lookupServerLifeCycleRuntime, lookupServerRuntime, sendNotification, start, suspend
ServerConfig	ExpectedToRun, lookupServerLifeCycleRuntime, sendNotification
ServerRuntime	forceShutdown, resume, shutdown, start, stop

Default Group Associations

By default, WebLogic Server grants four default global roles to four default groups. When you add a user to one of these groups, the user is automatically granted the global role. These default group associations are shown in [Table 4-6](#).

Table 4-6 Default Group Associations

Members of This Group	Are In This Global Role
Administrators	Admin
Deployers	Deployer
Operators	Operator
Monitors	Monitor

Components of a Security Role: Role Conditions, Expressions, and Role Statements

A **role condition** is a condition under which a security role (global or scoped) will be granted to a user or group. The role conditions that are available in this release of WebLogic Server are:

- **User Name of the Caller**—Creates a condition for a security role based on a user name. For example, you might create a condition indicating that only the user `John` can be granted the `BankTeller` security role.
- **Caller is a Member of the Group**—Creates a condition for a security role based on a group. For example, you might create a condition indicating that only users in the group `FullTimeBankEmployees` can be granted the `BankTeller` security role. BEA recommends this role condition for more efficient security management.
- **Hours of Access are Between**—Creates a condition for a security role based on a specified time period. For example, you might create a condition indicating that the `BankTeller` security role can only be granted to users when the bank is open.

When you use the `Hours of Access are Between` role condition, the security role will be granted to *all users* during the hours you specify, unless you further restrict the users by adding one of the other role conditions.

These role conditions, along with the specific information you supply for the condition (such as an actual user name, group, or start/stop times), are called **expressions**. An example of an expression that you may see in the WebLogic Server Administration Console is shown in [Figure 4-3](#).

Figure 4-3 Expression Example

```
Caller is a member of the group
FullTimeBankEmployees
```

In this expression example, the first line is the role condition, the second line is the specific information you supply for the condition—in this case, a group called `FullTimeBankEmployees`.

A **role statement** is a collection of expressions that define how a security role is granted, and is therefore the main part of any security role you create. The ability to use multiple expressions means that you can create complex security roles that meet your organization's security requirements. The use of `and` and `or` between these expressions, as well as the ordering of the expressions, is also an important feature:

- `And` is used to specify that all the expressions must be true in order for the security role to be granted.
- `Or` is used to specify that at least one of the expressions must be true in order for the security role to be granted.

The entire role statement must be true in order for a user or group to be granted the security role. More restrictive expressions should come later in a role statement.

An example of a role statement that you may see in the Administration Console is shown in [Figure 4-4](#).

Figure 4-4 Role Statement Example

```
Caller is a member of the group
FullTimeBankEmployees
and Hours of access are between
08:00:00 and 19:00:00
```

In this role statement example, there are two expressions: the first and second lines contain an expression based on the `Caller is a Member of the Group` role condition, and the third and fourth lines contain another expression based on the `Hours of Access are Between` role condition.

Working with Global Roles

The following sections provide instructions for working with global roles:

- [“Creating Global Roles” on page 4-14](#)
- [“Modifying Global Roles” on page 4-17](#)
- [“Deleting Global Roles” on page 4-18](#)

Note: This section describes how to create, modify, and delete global roles. Because they are always scoped to a WebLogic resource, instructions for creating, modifying, and deleting scoped roles are provided under [“Working with Scoped Roles” on page 4-18](#).

Creating Global Roles

Notes: The section [“Using the Administration Console to Create Security Roles” on page 4-3](#) may also be helpful to review before creating security roles. If you are creating global roles that will be used to secure Server resources, be sure to adhere to the advice given in [“Maintaining a Consistent Security Scheme” on page 2-10](#).

To create a new global role:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm for which you are creating a global role (for example, `myrealm`).
3. Click Global Roles to display the Global Roles page and table of currently defined global roles.
4. Click Configure a new Global Role....

Note: If multiple WebLogic Role Mapping providers are configured in the security realm, an intermediate page will list them in a table. From the table, select which WebLogic Role Mapping provider’s database should store information for the new global role before performing step 5.

5. On General tab, enter the name of the global role in the Name field.

Do not use blank spaces, commas, hyphens, or any characters in this comma-separated list: `\t, <, >, #, |, &, ~, ?, (,), {, }`. Security role names are case sensitive. All security role names are singular and the first letter is capitalized, according to the BEA convention.

The proper syntax for a security role name is as defined for an `Nmtoken` in the [Extensible Markup Language \(XML\) Recommendation](#).

6. Click Apply to save your changes.
7. Select the Conditions tab to display the Role Editor page (see [Figure 4-5](#)).

Figure 4-5 Role Editor Page

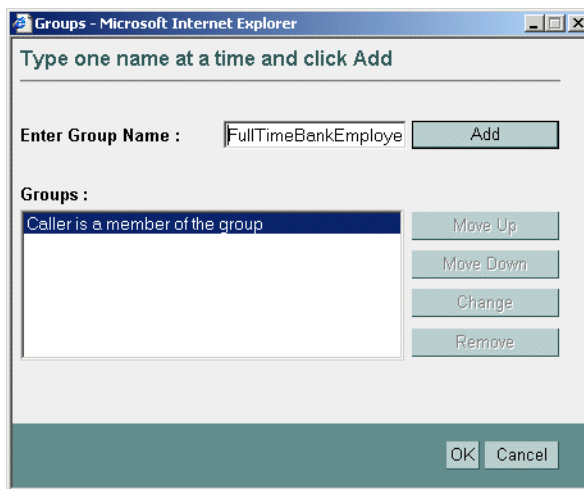
The screenshot displays the Role Editor interface. It features two main sections: 'Role Condition' and 'Role Statement'. The 'Role Condition' section contains a list box with three items: 'User name of the caller', 'Caller is a member of the group', and 'Hours of access are between'. To the right of this list box is an 'Add' button. The 'Role Statement' section contains a large text area for entering statements. To the right of this text area are five buttons: 'Move Up', 'Move Down', 'Change', 'Edit', and 'Remove'.

8. In the Role Condition list box, click one of the conditions. (For more information about the different role conditions, see [“Components of a Security Role: Role Conditions, Expressions, and Role Statements”](#) on page 4-12.)

BEA recommends that you create expressions using the `Caller is a Member of the Group` condition where possible. When a group is used to create a security role, the security role can be granted to all members of the group (that is, multiple users).

9. Click Add to display a customized window. (See [Figure 4-6](#).)

Figure 4-6 Customized Window for Caller is a Member of the Group Condition



10. If you selected the **Hours of Access are Between** condition, use the **Time Constraint** window to select start and end times, then click **OK**. The window closes and an expression appears in the **Role Statement** list box. (See [Figure 4-7](#) for an example.)

If you selected one of the other conditions, follow these steps:

- a. Use the **Users or Groups** window to enter the name of a user or group, then click **Add**. An expression appears in the list box.

You can repeat this step multiple times to add more than one user or group.

- b. If necessary, use the buttons located to the right of the list box to modify the expressions.

Move Up and **Move Down** change the ordering of the highlighted user or group name, and therefore the order in which they are evaluated. **Change** switches the highlighted **and** and **or** statements between expressions. **Remove** deletes the highlighted user or group name.

- c. Click **OK** to add the expression to the role statement. The window closes and the expression appears in the **Role Statement** list box. (See [Figure 4-7](#).)

Figure 4-7 Example Expression in Role Statement List Box

The screenshot shows two list boxes. The top list box is titled 'Role Condition:' and contains three items: 'User name of the caller', 'Caller is a member of the group' (which is highlighted), and 'Hours of access are between'. To the right of this list box is a single button labeled 'Add'. The bottom list box is titled 'Role Statement:' and contains two items: 'Caller is a member of the group' and 'FullTimeBankEmployees'. To the right of this list box are five buttons: 'Move Up', 'Move Down', 'Change', 'Edit...', and 'Remove'.

11. If desired, repeat steps 8-10 to add expressions based on different role conditions.
12. If necessary, use the buttons located to the right of the Role Statement list box to modify the expressions:
 - Move Up and Move Down change the ordering of the highlighted expression, and therefore the order in which they are evaluated.
 - Change switches the highlighted **and** and **or** statements between expressions.
 - Edit... reopens the customized window for the highlighted expression and allows you to modify the expression.
 - Remove deletes the highlighted expression.
13. When all the expressions in the Role Statement list box are correct, click Apply.

The General tab appears.

Note: You can also click Reset at the bottom of the Role Editor page to restore the page to its original state (that is, to undo any of your changes).

Modifying Global Roles

The procedure for modifying global roles is, for the most part, the same as the procedure for creating a new global role.

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.

2. Expand the security realm from which you are modifying a global role (for example, `myrealm`).
3. Click Global Roles.
The Global Roles page displays all the global roles currently defined in the WebLogic Role Mapping provider's database.
4. From the table, select the global role that you want to modify.
A table that lists all the scoped roles for the WebLogic resource appears in the right pane.
5. Select the Conditions tab to display the Role Editor page.
6. Make your changes, using steps 8- 12 in [“Creating Global Roles” on page 4-14](#) as a guide.
7. Click Apply to save your changes.

Deleting Global Roles

To delete a global role:

1. In the left pane of the WebLogic Server Administration Console, expand Security → Realms.
2. Expand the security realm from which you are deleting a global role (for example, `myrealm`).
3. Click Global Roles.
The Global Roles page displays all the global roles currently defined in the WebLogic Role Mapping provider's database.
4. Click the trash can icon that is located in the same row as the global role you want to delete.
5. Click Yes to confirm the deletion.
6. Click Continue.

The Global Roles page no longer shows the deleted global role in the table.

Working with Scoped Roles

The following sections provide instructions for working with scoped roles for the various types of WebLogic resources:

- [“Creating Scoped Roles” on page 4-19](#)

- [“Modifying Scoped Roles” on page 4-30](#)
- [“Deleting Scoped Roles” on page 4-31](#)

Creating Scoped Roles

To create a scoped role for a WebLogic resource:

- [“Step 1: Select the WebLogic Resource” on page 4-19](#)
- [“Step 2: Create the Scoped Role” on page 4-27](#)
- [“Step 3: Create the Role Conditions” on page 4-27](#)

Note: The instructions for working with scoped roles vary slightly from WebLogic resource to WebLogic resource. Be sure to follow any variations noted in this procedure that pertain to the type of WebLogic resource with which you are working. For more information, see [Chapter 2, “Types of WebLogic Resources.”](#)

Step 1: Select the WebLogic Resource

Follow the instructions in the appropriate section to select the type of WebLogic resource for which you will be creating a scoped role:

- [“Administrative Resources” on page 4-20](#)
- [“Application Resources” on page 4-20](#)
- [“COM Resources” on page 4-20](#)
- [“EIS Resources” on page 4-21](#)
- [“EJB Resources” on page 4-21](#)
- [“JDBC Resources” on page 4-22](#)
- [“JNDI Resources” on page 4-24](#)
- [“Server Resources” on page 4-25](#)
- [“URL Resources” on page 4-25](#)
- [“Web Service Resources” on page 4-26](#)

Administrative Resources

In the left pane of the WebLogic Server Administration Console, right-click the name of the WebLogic Server domain (for example, `examples`), and choose Define Scoped Role... to display the Scoped Roles page.

If available, a table of currently defined scoped roles appears in the right pane.

Application Resources

1. In the left pane of the WebLogic Server Administration Console, expand Deployments → Applications.

Optionally expand the Enterprise Application (EAR) for which you are creating a scoped role to see the different types of WebLogic resources it contains.

2. Right-click the name of an Enterprise Application (EAR) and choose Define Scoped Role... to display the Scoped Roles page.

If available, a table of currently defined scoped roles appears in the right pane.

COM Resources

If a package of EJB classes (such as `ejb20.basic.beanManaged.*`) will be accessed by a COM client:

1. In the left pane of the WebLogic Server Administration Console, expand Deployments, then EJB.

The EJB node expands to show the EJB JARs that are currently deployed.

2. Right-click the name of an EJB JAR containing the EJB that will be used to access the package, and choose Define Policies and Roles for Individual Beans... to display a list of EJBs.

3. Click the [Define JCOM Roles] link that is located in the same row as the EJB that will be used to access the package.

The General tab's COM Class field already shows the name of the package to which you want to scope the security role.

The value in the COM class field is a Java class or package name that is exposed to COM via the jCOM bridge.

4. Click the Define Role... button to display the Select Roles page.

If available, a table of currently defined scoped roles appears in the right pane.

If a package of Java classes (such as `java.util.*`) or individual classes (such as `java.util.Collection`) will be accessed by a COM client:

1. In the left pane of the WebLogic Server Administration Console, expand Services.
2. Right-click the JCOM node and choose Define Role....
3. On the General tab, in the COM class field, enter the name of the Java class or package to which you want to scope the security role.

The value you enter in the COM class field is a Java class or package name that is exposed to COM via the jCOM bridge.

4. Click the Define Role... button to display the Select Roles page.

If available, a table of currently defined scoped roles appears in the right pane.

EIS Resources

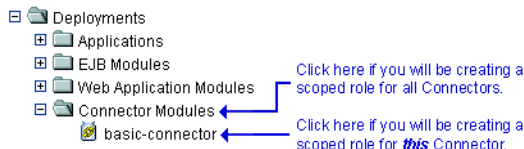
1. In the left pane of the WebLogic Server Administration Console, expand Deployments.

The Deployments node expands to show the types of WebLogic resources that can be deployed.

2. Right-click at the level of the EIS resource for which you want to create the scoped role, and choose Define Scoped Role... to display the Scoped Roles page.

If you will be creating a scoped role for *all* Connectors, right-click Connector Modules in the navigation tree. If you will be creating a scoped role for a *particular* Connector, expand Connector Modules, then right-click the name of a Connector. Figure 4-8 illustrates where you might click, using the `basic-connector` as an example.

Figure 4-8 Deployments Portion of the Administration Console Navigation Tree



If available, a table of currently defined scoped roles appears in the right pane.

EJB Resources

Note: These instructions also apply to Message-driven Beans (MDBs).

1. In the left pane of the WebLogic Server Administration Console, expand Deployments.

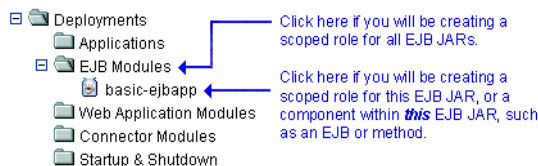
The Deployments node expands to show the types of WebLogic resources that can be deployed.

2. Right-click at the level of the EJB resource for which you want to create the scoped role.

If you will be creating a scoped role for *all* EJB JARs, right-click EJB Modules in the navigation tree. If you will be creating a scoped role for a *particular* EJB JAR, or for an EJB within a JAR, expand EJB Modules, then right-click the name of an EJB JAR.

Figure 4-9 illustrates where you might click, using the `basic-ejbapp` JAR as an example.

Figure 4-9 Deployments Portion of the Administration Console Navigation Tree



3. If you will be creating the scoped role for all EJB JARs or for a particular EJB JAR (that is, for *all* the EJBs in the JAR), choose Define Scoped Role... to display the Scoped Roles page.

If you will be creating the scoped for a *particular* EJB in an EJB JAR, follow these steps:

- a. Choose Define Policies and Roles for Individual Beans... to display a list of EJBs.
- b. Click the [Define Scoped Roles] link that is located in the same row as the EJB for which you want to create the scoped role.

If available, a table of currently defined scoped roles appears in the right pane.

JDBC Resources

1. In the left pane of the WebLogic Server Administration Console, expand Services → JDBC.

The JDBC node expands to show nodes for various JDBC components (connection pools, MultiPools, and data sources).

2. Right-click at the level of the JDBC resource for which you want to create the scoped role, and choose Define Scoped Role... to display the Scoped Roles page.

If you will be creating a scoped role for *all* connection pools, right-click Connection Pools in the navigation tree. If you will be creating a scoped role for a *particular* connection pool, expand Connection Pools, then right-click the name of a connection pool. To create a scoped role for an individual MultiPool, expand MultiPools, then right-click the name of the MultiPool.

Note: You cannot create a scoped role that encompasses all MultiPools.

Figure 4-10 illustrates where you might click, using various connection pools and a MultiPool as an example.

Figure 4-10 Services Portion of the Administration Console Navigation Tree



If available, a table of currently defined scoped roles appears in the right pane.

JMS Resources

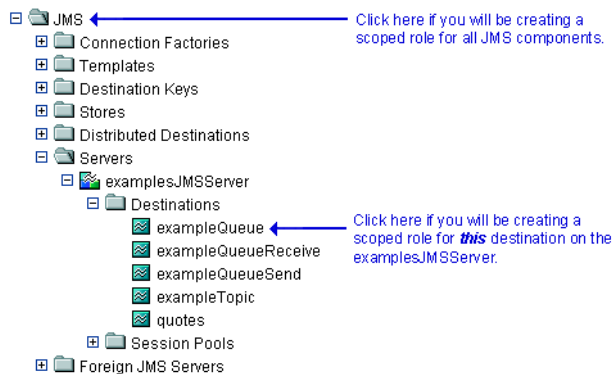
1. In the left pane of the WebLogic Server Administration Console, expand Services →JMS.

The JMS node expands to show nodes for various JMS components (connection factories, templates, destination keys, and so on).

2. Right-click at the level of the JMS resource for which you want to create the scoped role, and choose Define Scoped Role... to display the Scoped Roles page.

If you will be creating a scoped role for *all* JMS components, right-click JMS in the navigation tree. If you will be creating a scoped role for a *particular* destination on a JMS server, expand Servers, then the JMS server and the Destinations node, then right-click the name of a destination. Figure 4-11 illustrates where you might click, using various destinations on the `examplesJMS`Server as an example.

Figure 4-11 Services Portion of the Administration Console Navigation Tree



If available, a table of currently defined scoped roles appears in the right pane.

JNDI Resources

1. In the left pane of the WebLogic Server Administration Console, expand Servers.
The Servers node expands to show the servers available in the current WebLogic Server domain.
2. Right-click the name of a server that contains the JNDI resource for which you want to create the scoped role. (For example, `myserver`.)
3. From the menu that appears, select the View JNDI Tree option.
The JNDI tree for the server appears in a new Administration Console window.
4. In the new Administration Console window, right-click at the level of the JNDI tree at which you want to create the scoped role, and choose Define Scoped Role... to display the Scoped Roles page.

If you will be creating a scoped role for *a group of* objects, right-click the node in the navigation tree that represents that object type. If you will be creating a scoped role for a *particular* object, expand the node that represents that object, then right-click the name of an object.

Figure 4-12 illustrates where you might click, using the `examplesServer` JNDI tree as an example.

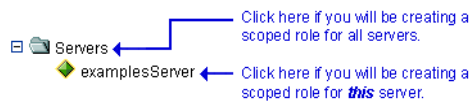
Figure 4-12 New Administration Console Window for examplesServer JNDI Tree

If available, a table of currently defined scoped roles appears in the right pane.

Server Resources

1. In the left pane of the WebLogic Server Administration Console, expand Servers.
The Servers node expands to show the different Server resources for which a scoped role can be created.
2. Right-click at the level of the Server resource at which you want to create the scoped role, and choose Defined Scoped Role... to display the Scoped Roles page.

If you will be creating a scoped role for *all* servers, right-click Servers in the navigation tree. If you will be creating a scoped role for a *particular* server, expand Servers, then right-click the name of a server. [Figure 4-13](#) illustrates where you might click, using the **examplesServer** as an example.

Figure 4-13 Servers Portion of the Administration Console Navigation Tree

If available, a table of currently defined scoped roles appears in the right pane.

URL Resources

1. In the left pane of the WebLogic Server Administration Console, expand Deployments.

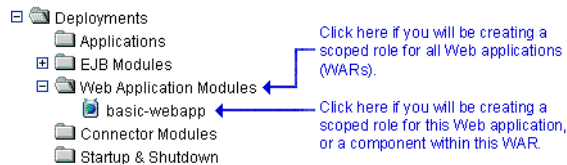
The Deployments node expands to show the types of WebLogic resources that can be deployed.

2. Right-click at the level of the URL (Web) resource at which you want to create the scoped role.

If you will be creating a scoped role for *all* Web Applications (WARs), right-click Web Application Modules in the navigation tree. If you will be creating a scoped role for a *particular* WAR or a component in a WAR (for example, a specific servlet or JSP), expand Web Application Modules, then right-click the name of a Web Application (WAR).

Figure 4-14 illustrates where you might click, using the `basic-webapp` WAR as an example.

Figure 4-14 Deployments Portion of the Administration Console Navigation Tree



3. If you will be creating the scoped role for all Web Applications (WARs), choose Define Scoped Role... to display the Scoped Roles page.

If you will be creating the scoped role for a particular WAR, or a component within a WAR, follow these steps:

- a. Choose Define Scoped Role... to display the General tab.
- b. Enter a URL pattern in the text field.

A URL pattern is a path to a specific component within a Web Application. Or, you can use `/*` to associate the scoped role with all components (servlets, JSPs, and so on) within the Web Application.


- c. Click the Define Scoped Role... button to display the Scoped Roles page.

If available, a table of currently defined scoped roles appears in the right pane.

Web Service Resources

1. In the left pane of the WebLogic Server Administration Console, expand Deployments.
2. If the Web Service is packaged as an EAR file, expand Applications, then expand the name of the application that contains the Web Service. The application expands to show the components that make up the application, including the Web Service Web Application.

If the Web Service is packaged as a stand-alone Web Application, expand Web Application Modules.

Note: The  icon designates a Web Service.

3. Right-click the name of the Web Service and choose Define Scoped Role... to display the Scoped Roles page.

If available, a table of currently defined scoped roles appears in the right pane.

Step 2: Create the Scoped Role

1. On the Scoped Roles page, click Configure a new Scoped Role....

Note: If multiple WebLogic Role Mapping providers are configured in the security realm, an intermediate page will list them in a table. From the table, select which WebLogic Role Mapping provider's database should store information for the new scoped role before performing step 5.

2. On the General tab, enter the name of the scoped role in the Name field.

Do not use blank spaces, commas, hyphens, or any characters in this comma-separated list: \t, <, >, #, |, &, ~, ?, (,), {, }. Security role names are case sensitive. All security role names are singular and the first letter is capitalized, according to BEA convention.

The proper syntax for a security role name is as defined for an Nmtoken in the [Extensible Markup Language \(XML\) Recommendation](#).

Warning: If you create a scoped role with the same name as a global role, the scoped role takes precedence over the global role.

3. Click Apply to save your changes.

Step 3: Create the Role Conditions

1. Select the Conditions tab to display the Role Editor page (see [Figure 4-15](#)).

Figure 4-15 Role Editor Page

The screenshot shows the Role Editor Page with two main sections:

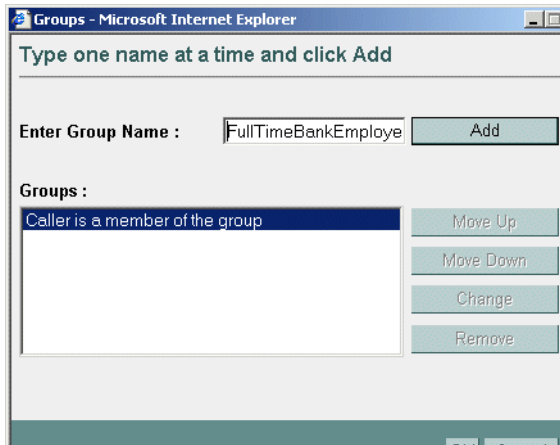
- Role Condition:** A list box containing three conditions: "User name of the caller", "Caller is a member of the group", and "Hours of access are between". To the right of this list box is an "Add" button.
- Role Statement:** A large text area for entering the role statement. To the right of this text area are five buttons: "Move Up", "Move Down", "Change", "Edit", and "Remove".

2. In the Role Condition list box, select one of the conditions. (For more information about the different role conditions, see [“Components of a Security Role: Role Conditions, Expressions, and Role Statements”](#) on page 4-12.)

BEA recommends that you create expressions using the `Caller is a Member of the Group` condition where possible. When a group is used to create a security role, the security role can be granted to all members of the group (that is, multiple users).

Because the JMS subsystem performs its security check only once and the `Hours of Access are Between` condition requires a subsequent security check, you should *not* use the `Hours of Access are Between` condition if you are creating a scoped role for a JMS resource.

3. Click Add to display a customized window. (See [Figure 4-16](#).)

Figure 4-16 Customized Window for Caller is a Member of the Group Condition

4. If you selected the *Hours of Access are Between* condition, use the *Time Constraint* window to select start and end times, then click OK. The window closes and an expression appears in the *Role Statement* list box. (See [Figure 4-17](#) for an example.)

If you selected one of the other conditions, follow these steps:

- a. Use the *Users or Groups* window to enter the name of a user or group, then click Add. An expression appears in the list box.

You can repeat this step multiple times to add more than one user or group.

- b. If necessary, use the buttons located to the right of the list box to modify the expressions.

Move Up and Move Down change the ordering of the highlighted user or group name, and therefore the order in which they are evaluated. Change switches the highlighted *and* and *or* statements between expressions. Remove deletes the highlighted user or group name.

- c. Click OK to add the expression to the role statement. The window closes and the expression appears in the *Role Statement* list box. (See [Figure 4-17](#).)

Figure 4-17 Example Expression in Role Statement List Box

The screenshot shows two main sections: 'Role Condition' and 'Role Statement'. The 'Role Condition' section has a list box containing three items: 'User name of the caller', 'Caller is a member of the group' (which is highlighted), and 'Hours of access are between'. To the right of this list box is an 'Add' button. The 'Role Statement' section has a list box containing two items: 'Caller is a member of the group' and 'FullTimeBankEmployees'. To the right of this list box are five buttons: 'Move Up', 'Move Down', 'Change', 'Edit...', and 'Remove'.

5. If needed, repeat steps 2- 4 to add expressions based on different role conditions.
6. If necessary, use the buttons located to the right of the Role Statement list box to modify the expressions:
 - Move Up and Move Down change the ordering of the highlighted expression, and therefore the order in which they are evaluated.
 - Change switches the highlighted `and` and `or` statements between expressions.
 - Edit... reopens the customized window for the highlighted expression and allows you to modify the expression.
 - Remove deletes the highlighted expression.
7. When all the expressions in the Role Statement list box are correct, click Apply.

The General tab is displayed.

Note: You can also click Reset at the bottom of the Role Editor page to restore the page to its original state (that is, to undo any of your changes).

Modifying Scoped Roles

To modify a scoped role for a WebLogic resource:

1. Navigate to the Scoped Roles page for the WebLogic resource, as described in [“Step 1: Select the WebLogic Resource” on page 4-19](#).
2. From the table, select the scoped role that you want to modify.

A table that lists all the scoped roles for the WebLogic resource appears in the right pane.

3. Select the Conditions tab.
4. Make your changes, using the instructions in [“Step 3: Create the Role Conditions” on page 4-27](#) as a guide.
5. Click Apply to save your changes.

Deleting Scoped Roles

To delete a scoped role for a WebLogic resource:

1. Navigate to the Scoped Roles page for the WebLogic resource, as described in [“Step 1: Select the WebLogic Resource” on page 4-19](#).

A table that lists all the scoped roles for the WebLogic resource appears in the right pane.

2. Click the trash can icon that is located in the same row as the scoped role you want to delete.
3. Click Yes to confirm the deletion.
4. Click Continue.

The Scoped Roles page no longer shows the deleted scoped role in the table.

Security Policies

The following sections describe the features and functions of security policies:

- [“Overview of Security Policies” on page 5-1](#)
- [“Security Policy Granularity and Inheritance” on page 5-2](#)
- [“Security Policy Storage and Prerequisites for Use” on page 5-2](#)
- [“Default Security Policies” on page 5-3](#)
- [“Protected Public Interfaces” on page 5-4](#)
- [“Components of a Security Policy: Policy Conditions, Expressions, and Policy Statements” on page 5-5](#)
- [“Working with Security Policies” on page 5-7](#)

Overview of Security Policies

A **security policy** is an association between a WebLogic resource and one or more users, groups, or security roles and is designed to protect the WebLogic resource against unauthorized access.

Note: Security policies replace the access control lists (ACLs) and permissions that were used to protect WebLogic resources in previous releases of WebLogic Server.

Security Policy Granularity and Inheritance

Security policies are always scoped to a WebLogic resource, but because WebLogic resources are hierarchical, the level at which you define a security policy is up to you. For example, you can define security policies on an entire Enterprise Application (EAR), an EJB (Enterprise JavaBean) JAR containing multiple EJBs, a particular EJB within that JAR, or a single method within that EJB.

If you create a security policy for a *type* of WebLogic resource (for example, EJB resources), all new instances of that WebLogic resource inherit the security policy. (For more information about the types of WebLogic resources, see [Chapter 2, “Types of WebLogic Resources.”](#)) This inheritance of security policies means that you can secure multiple WebLogic resources efficiently. Out of the box, WebLogic Server secures each WebLogic resource type with a default security policy that is inherited by all instances of that WebLogic resource. For more information, see [“Default Security Policies” on page 5-3.](#)

A security policy created for a specific instance of a WebLogic resource will override any security policy assigned to the WebLogic resource type. So, if you create a security policy for a particular EJB, this security policy (and not the one you created for the EJB resource type) will be used.

Security Policy Storage and Prerequisites for Use

Security policies are stored in the security provider database of the Authorization provider that is configured in the default (active) security realm. By default, the WebLogic Authorization provider is configured, and security policies are stored in the embedded LDAP server.

When creating a security policy with a user or group, the user or group must be defined in the security provider database of the Authentication provider that is configured in the default security realm. When creating a security policy with a security role, the security role (whether global or scoped) must be defined in the security provider database of the Role Mapping provider that is configured in the default security realm. By default, the WebLogic Authentication and Role Mapping providers are configured, and the default groups and default global roles are stored in the databases for these security providers (also the embedded LDAP server).

Note: For more information about the WebLogic Authentication, Authorization, and Role Mapping providers, see [“The WebLogic Security Providers”](#) in *Introduction to WebLogic Security*.

Default Security Policies

By default, WebLogic Server defines the security policies shown in [Table 5-1](#). These security policies are defined for each type of WebLogic resource described in [Chapter 2, “Types of WebLogic Resources,”](#) and are based on the default global roles and default groups.

Table 5-1 Default Security Policies for WebLogic Resources

WebLogic Resource	Security Policy
Administrative resources	Default global role: Admin
Application resources	None
EIS resources	Default group: Everyone
EJB resources	Default group: Everyone
COM resources	None
JDBC resources	Default group: Everyone
JNDI resources	Default group: Everyone
JMS resources	Default group: Everyone
Server resources	Default global roles: <ul style="list-style-type: none"> Admin Operator
URL resources (previously Web resources, deprecated)	Default group: Everyone
Web Services resources	Default group: Everyone

Caution: Do not modify the default security policies for Administrative and Server resources to make them more restrictive. Eliminating some of the existing security roles might negatively impact the functioning of WebLogic Server. However, if you like, you can make the default security policies more inclusive (for example, by adding new security roles).

For more information about the WebLogic resources shown in [Table 5-1](#), see [Chapter 2, “Types of WebLogic Resources.”](#)

You can add to the default security policies by creating your own. See [“Working with Security Policies” on page 5-7](#) for more information.

Protected Public Interfaces

The WebLogic Server Administration Console, the `weblogic.Admin` command, and MBean APIs are secured using the default security policies, which are based on the default global roles and default groups described in [Table 4-1, “Default Global Roles and Their Privileges,” on page 4-6](#) and [Table 4-6, “Default Group Associations,” on page 4-12](#). Therefore, to use the Administration Console, a user must belong to one of these default groups or be granted one of these global roles. Additionally, administrative operations that require interaction with MBeans are secured using the MBean protections described in [“MBean Protections” on page 2-7](#). Therefore, interaction with the following protected public interfaces typically must satisfy both security schemes.

- *The WebLogic Server Administration Console*—The WebLogic Security Service verifies whether a particular user can access the Administration Console when the user attempts to log in. If a user attempts to invoke an operation for which they do not have access, they see an Access Denied error.

For information about using this public interface, see the [Administration Console Online Help](#).

- *The `weblogic.Admin` command*—The WebLogic Security Service verifies whether a particular user has permission to execute a command when the user attempts to invoke the command. If a user attempts to invoke an operation for which the user does not have access, WebLogic Server throws a `weblogic.management.NoAccessRuntimeException`, which developers can catch explicitly in their programs. The server sends this exception to its log file, but you can also configure the server to send exceptions to standard out.

For information about using this public interface, see [“Protected MBean Attributes and Operations” on page 4-6](#) and [“weblogic.Admin Command-Line Reference”](#) in the *WebLogic Server Command Line Reference*.

Note: The `weblogic.Admin` command is a convenience utility that abstracts the interaction with the MBean APIs (described below). Therefore, for any administrative task you can perform using the `weblogic.Admin` command, you can also perform using the MBean APIs.

- *MBean APIs*—The WebLogic Security Service verifies whether a particular user has permission to access the API when the user attempts to perform an operation on the

MBean. If a user attempts to invoke an operation for which the user does not have access, WebLogic Server throws a `weblogic.management.NoAccessRuntimeException`, which developers can catch explicitly in their programs. The server sends this exception to its log file, but you can also configure the server to send exceptions to standard out.

For information about using these APIs, see [“Protected MBean Attributes and Operations” on page 4-6](#) and *Programming WebLogic JMX Services*.

Components of a Security Policy: Policy Conditions, Expressions, and Policy Statements

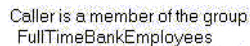
A **policy condition** is a condition under which a security policy will be created. The policy conditions that are available in this release of WebLogic Server are:

- **User Name of the Caller**—Creates a condition for a security policy based on a user name. For example, you might create a condition indicating that only the user `John` can access the `Deposit EJB`.
- **Caller is a Member of the Group**—Creates a condition for a security policy based on a group. When a group is used to create a security policy, the security policy is assigned to all members of the group. For example, you might create a condition indicating that only users in the group `FullTimeBankEmployees` can access the `Deposit EJB`.
- **Caller is Granted the Role**—Creates a condition for a security policy based on a security role. For example, you might create a condition indicating that only users and groups in the `BankTeller` security role can access the `Deposit EJB`.
- **Hours of Access are Between**—Creates a condition for a security policy based on a specified time period. For example, you might create a condition indicating that the `BankTeller` security role can only access the `Deposit EJB` when the bank is open.
- **Server is in Development Mode**—Creates a condition for a security policy based on whether the server is running in development mode.
- **Element requires signature by**—Creates a condition for a security policy based on who has digitally signed an element in the SOAP request message that invokes a Web Service operation. For example, you might create a condition that says the `getBalance` operation can only be invoked if the `AccountNumber` element in the incoming SOAP request has been digitally signed by the `BankTeller` security role.

Note: This policy condition is used only when securing Web Services and individual Web Service operations.

These policy conditions, along with the specific information you supply for the condition (such as an actual user name, group, or security role, or start/stop times), are called **expressions**. An example of an expression that you may see in the WebLogic Server Administration Console is shown in [Figure 5-1](#).

Figure 5-1 Expression Example



```
Caller is a member of the group  
FullTimeBankEmployees
```

In this expression example, the first line is the policy condition, the second line is the specific information you supply for the condition—in this case, a group called `FullTimeBankEmployees`.

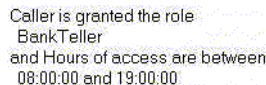
A **policy statement** is the collection of expressions that define who is granted access to a WebLogic resource, and is therefore the main part of any security policy you create. The ability to use multiple expressions means that you can create complex security policies that meet your organization's security requirements. The use of `and` and `or` between these expressions, as well as the ordering of the expressions, is also an important feature:

- `And` is used to specify that all the expressions must be true in order for the security policy to be applied.
- `Or` is used to specify that at least one of the expressions must be true in order for the security policy to be applied.

Notes: The entire policy statement must be true in order for security policy to be applied. More restrictive expressions should come later in a policy statement.

An example of a policy statement that you may see in the Administration Console is shown in [Figure 5-2](#).

Figure 5-2 Policy Statement Example



```
Caller is granted the role  
BankTeller  
and Hours of access are between  
08:00:00 and 19:00:00
```

In this policy statement example, there are two expressions: the first and second lines contain an expression based on the `Caller is Granted the Role` policy condition, and the third and fourth lines contain another expression based on the `Hours of Access are Between` policy condition.

Working with Security Policies

The following sections provide instructions for working with security policies for the various types of WebLogic resources:

- [“Creating Security Policies” on page 5-7](#)
- [“Modifying Security Policies” on page 5-21](#)
- [“Deleting Security Policies” on page 5-21](#)

Creating Security Policies

The instructions for working with security policies vary slightly from WebLogic resource to WebLogic resource. Be sure to follow any variations noted in this procedure that pertain to the type of WebLogic resource with which you are working. For more information, see [Chapter 2, “Types of WebLogic Resources.”](#)

Note: In this version of WebLogic Server, you must keep track of the security policies you create. With the exception of those created for URL (Web) resources, there is currently no listing mechanism for previously created security policies in the WebLogic Server Administration Console.

Step 1: Select the WebLogic Resource

Follow the instructions in the appropriate section to select the type of WebLogic resource:

- [“Administrative Resources” on page 5-8](#)
- [“Application Resources” on page 5-8](#)
- [“COM Resources” on page 5-9](#)
- [“EIS Resources” on page 5-10](#)
- [“EJB Resources” on page 5-11](#)
- [“JDBC Resources” on page 5-12](#)
- [“JMS Resources” on page 5-13](#)
- [“JNDI Resources” on page 5-14](#)
- [“Server Resources” on page 5-15](#)
- [“URL Resources” on page 5-16](#)

- [“Web Service Resources” on page 5-17](#)

Administrative Resources

In the left pane of the WebLogic Server Administration Console, right-click the name of the WebLogic Server domain (for example, `examples`), and choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).

Note: In this version of WebLogic Server, you can only secure the `unlockuser` method. For more information about user lockouts, see [“Protecting User Accounts” in *Managing WebLogic Security*](#).

Notice the `Caller is Granted the Role: Admin` policy statement that the Administrative resource you selected has inherited from the default security policy associated with the Administrative resource type. If you proceed to [“Step 2: Create the Policy Conditions” on page 5-18](#), you will be overriding this default security policy. For more information, see [“Default Security Policies” on page 5-3](#) and [“Security Policy Granularity and Inheritance” on page 5-2](#).

Application Resources

1. In the left pane of the WebLogic Server Administration Console, expand Deployments → Applications.

Optionally, expand the Enterprise Application (EAR) for which you are creating a scoped role to see the different types of WebLogic resources it contains.

2. Right-click the name of the Enterprise Application and choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3](#)).

Figure 5-3 Policy Editor Page

The screenshot shows the Policy Editor interface. At the top, the 'Methods' dropdown is set to 'ALL'. Below it, the 'Policy Condition' list contains five items: 'User name of the caller', 'Caller is a member of the group' (which is highlighted), 'Caller is granted the role', 'Hours of access are between', and 'Server is in Development Mode'. An 'Add' button is to the right of this list. The 'Policy Statement' section is a large empty box with five buttons to its right: 'Move Up', 'Move Down', 'Change', 'Edit...', and 'Remove'. At the bottom, the 'Inherited Policy Statement' section is also an empty box.

Notice that there are no default policy statements for Application resources. (For more information, see [“Default Security Policies” on page 5-3.](#))

COM Resources

If a package of EJB classes (such as `ejb20.basic.beanManaged.*`) will be accessed by a COM client:

1. In the left pane of the WebLogic Server Administration Console, expand Deployments → EJB. The EJB node expands to show the EJB JARs that are currently deployed.
2. Right-click the name of an EJB JAR containing the EJB that will be used to access the package, and choose Define Policies and Roles for Individual Beans... to display a list of EJBs.
3. Click the [Define JCOM Policies] link that is located in the same row as the EJB that will be used to access the package.

The General tab's COM Class field already shows the name of the package for which you want to create the security policy.

The value in the COM class field is a Java class or package name that is exposed to COM via the jCOM bridge.

4. Click the Define Policy... button to display the Policy Editor page (see [Figure 5-3, "Policy Editor Page," on page 5-9](#)).

Note: If you create a security policy for a package of EJB classes that will be accessed by a COM client and want to use scoped roles in the `Caller is Granted the Role` condition, be sure to use the scoped role you associated with the package of EJB classes (described in ["COM Resources" on page 4-20](#)).

If a package of Java classes (such as `java.util.*`) or individual classes (such as `java.util.Collection`) will be accessed by a COM client:

1. In the left pane of the WebLogic Server Administration Console, expand Services.
2. Right-click the JCOM node and choose Define Policy....
3. On the General tab, in the COM class field, enter the name of the Java class or package you want to protect, then click the Define Policy... button to display the Policy Editor page (see [Figure 5-3, "Policy Editor Page," on page 5-9](#)).

The value you enter in the COM class field is a Java class or package name that is exposed to COM via the jCOM bridge.

Notice that there are no default policy statements for COM resources. (For more information, see ["Default Security Policies" on page 5-3](#).)

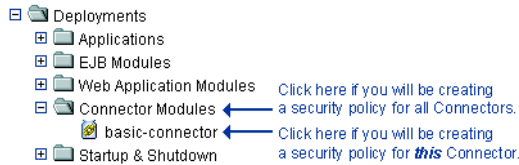
EIS Resources

1. In the left pane of the WebLogic Server Administration Console, expand Deployments.

The Deployments node expands to show the types of WebLogic resources that can be deployed.

2. Right-click at the level of the EIS resource at which you want to create the security policy, and choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3, "Policy Editor Page," on page 5-9](#)).

If you will be creating a security policy for *all* Connectors, right-click Connector Modules in the navigation tree. If you will be creating a security policy for a *particular* Connector, expand Connector Modules, then right-click the name of a Connector. [Figure 5-4](#) illustrates where you might click, using the `basic-connector` Connector as an example.

Figure 5-4 Deployments Portion of the Administration Console Navigation Tree

Notice the `Caller is Granted the Role: Everyone` policy statement that the EIS resource you selected has inherited from the default security policy associated with the EIS resource type. If you proceed to [“Step 2: Create the Policy Conditions” on page 5-18](#), you will be overriding this default security policy. For more information, see [“Default Security Policies” on page 5-3](#) and [“Security Policy Granularity and Inheritance” on page 5-2](#).

EJB Resources

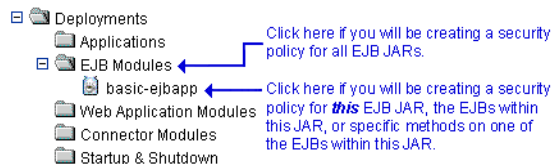
Note: These instructions also apply to Message-Driven Beans (MDBs).

1. In the left pane of the WebLogic Server Administration Console, expand Deployments.

The Deployments node expands to show the types of WebLogic resources that can be deployed.

2. Right-click at the level of the EJB resource at which you want to create the security policy.

If you will be creating a security policy for *all* the EJB JARs with a single security policy, right-click EJB Modules in the navigation tree. If you will be creating a security policy for a *particular* EJB JAR, an EJB within a JAR, or a method on one of the EJBs within a JAR, expand EJB Modules, then right-click the name of an EJB JAR. [Figure 5-5](#) illustrates where you might click, using the `basic-ejbapp` JAR as an example.

Figure 5-5 Deployments Portion of the Administration Console Navigation Tree

3. If you will be creating a security policy for all EJB JARs or for a particular EJB JAR, choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).

If you will be creating a security policy for a particular EJB within an EJB JAR, or a method on one of the EJBs within the JAR, follow these steps:

- a. Choose Define Policies and Roles for Individual Beans... to display a list of EJBs.
- b. Click the [Define Security Policies] link that corresponds to the particular EJB you want to secure (regardless of whether you want to secure the entire EJB or a particular method within the EJB) to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).

Notice the `Caller is Granted the Role: Everyone` policy statement that the EJB resource you selected has inherited from the default security policy associated with the EJB resource type. If you proceed to [“Step 2: Create the Policy Conditions” on page 5-18](#), you will be overriding this default security policy. For more information, see [“Default Security Policies” on page 5-3](#) and [“Security Policy Granularity and Inheritance” on page 5-2](#).

4. If will be creating a security policy for a particular EJB within an EJB JAR, specify which EJB method you want to protect, or select `ALL` to protect all methods.

JDBC Resources

1. In the left pane of the WebLogic Server Administration Console, expand Services →JDBC.

The JDBC node expands to show nodes for various JDBC components (connection pools, MultiPools, and data sources).

2. Right-click at the level of the JDBC resource at which you want to create the security policy, and choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).

If you will be creating a security policy for *all* the connection pools with a single security policy, right-click Connection Pools in the navigation tree. If you will be creating a security policy for a *particular* connection pool, expand Connection Pools, then right-click the name of a connection pool. If you will be creating a security policy for individual MultiPools, expand MultiPools, then right-click the name of a MultiPool.

Notes: You cannot secure all MultiPools with a single security policy.

If a security policy controls access to a connection pool that is in a MultiPool, access checks will be performed at both levels of the JDBC resource hierarchy (once at the MultiPool level, and again at the individual connection pool level). As with all types of WebLogic resources, this double checking ensures that the most restrictive security policy controls access.

[Figure 5-10](#) illustrates where you might click, using various connection pools and a MultiPool as an example.

Figure 5-6 Services Portion of the Administration Console Navigation Tree

Notice the Caller is Granted the Role: Everyone policy statement that the JDBC resource you selected has inherited from the default security policy associated with the JDBC resource type. If you proceed to [“Step 2: Create the Policy Conditions” on page 5-18](#), you will be overriding this default security policy. For more information, see [“Default Security Policies” on page 5-3](#) and [“Security Policy Granularity and Inheritance” on page 5-2](#).

3. If you will be creating a security policy for a particular connection pool, use the Methods drop-down menu to specify a method that you want to protect, or select ALL to protect all methods.

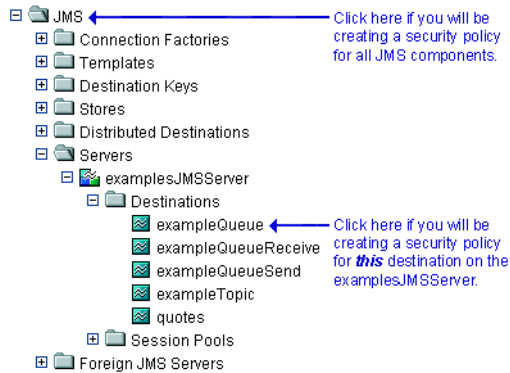
JMS Resources

1. In the left pane of the WebLogic Server Administration Console, expand Services → JMS.

The JMS node expands to show nodes for various JMS components (connection factories, templates, destination keys, and so on).

2. Right-click at the level of the JMS resource for which you want to create the security policy, and choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).

If you will be creating a security policy for *all* JMS components, right-click JMS in the navigation tree. If you will be creating a security policy for a *particular* destination (JMS queue or JMS topic) on a JMS server, expand Servers, then the JMS server and the Destinations node, then right-click the name of a destination. [Figure 5-7](#) illustrates where you might click, using various destinations on the `examplesJMS` as an example.

Figure 5-7 Services Portion of the Administration Console Navigation Tree

Notice the **Caller is Granted the Role: Everyone** policy statement that the JMS resource you selected has inherited from the default security policy associated with the JMS resource type. If you proceed to [“Step 2: Create the Policy Conditions” on page 5-18](#), you will be overriding this default security policy. For more information, see [“Default Security Policies” on page 5-3](#) and [“Security Policy Granularity and Inheritance” on page 5-2](#).

3. If you will be creating a security policy for a particular destination on a JMS server, use the Methods drop-down menu to specify a method that you want to protect, or select **ALL** to protect all methods.

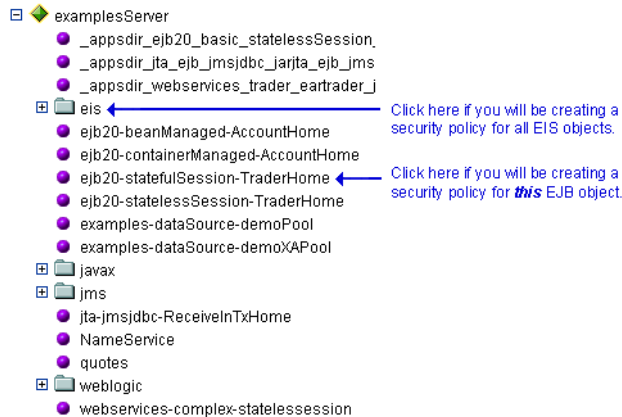
JNDI Resources

1. In the left pane of the WebLogic Server Administration Console, expand Servers.
The Servers node expands to show the servers available in the current WebLogic Server domain.
2. Right-click the name of the server that contains the JNDI resource for which you want to create the security policy. (For example, `myserver`.)
3. From the menu that appears, select the View JNDI Tree option.
The JNDI tree for the server appears in a new Administration Console window.
4. In the new Administration Console window, right-click at the level of the JNDI tree at which you want to create the security policy, and choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).

To create a security policy for *a group of* objects, right-click the node in the navigation tree that represents that object type. To create a security policy for a *particular* object, expand

the node that represents that object, then right-click the name of an object. [Figure 5-8](#) illustrates where you might click, using the `examplesServer` JNDI tree as an example.

Figure 5-8 New Administration Console Window for `examplesServer` JNDI Tree



Notice the Caller is Granted the Role: Everyone policy statement that the JNDI resource you selected has inherited from the default security policy associated with the JNDI resource type. If you proceed to “[Step 2: Create the Policy Conditions](#)” on page 5-18, you will be overriding this default security policy. For more information, see “[Default Security Policies](#)” on page 5-3 and “[Security Policy Granularity and Inheritance](#)” on page 5-2.

- Using the Methods drop-down menu, specify which JNDI method you want to protect, or select `ALL` to protect all methods.

Server Resources

- In the left pane of the WebLogic Server Administration Console, expand Servers. The Servers node expands to show the different server resources that can be secured.
- Right-click at the level of the Server resource at which you want to create a security policy, and choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3](#), “Policy Editor Page,” on page 5-9).

If you will be creating a security policy for *all* servers, right-click Servers in the navigation tree. If you will be creating a security policy for a *particular* server, expand Servers, then right-click the name of a server. [Figure 5-9](#) illustrates where you might click, using the `examplesServer` as an example.

Figure 5-9 Servers Portion of the Administration Console Navigation Tree



Notice the Caller is Granted the Role: Admin or Caller is Granted the Role: Operator policy statement that the server resource you selected has inherited from the default security policy associated with the server resource type. If you proceed to [“Step 2: Create the Policy Conditions” on page 5-18](#), you will be overriding this default security policy. For more information, see [“Default Security Policies” on page 5-3](#) and [“Security Policy Granularity and Inheritance” on page 5-2](#).

3. Using the Methods drop-down menu, specify a method that you want to protect, or select ALL to protect all methods.

URL Resources

1. In the left pane of the WebLogic Server Administration Console, expand Deployments.

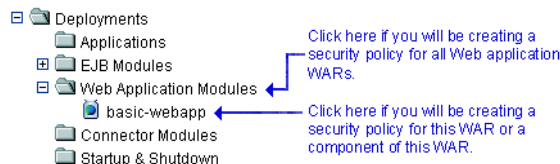
The Deployments node expands to show the types of WebLogic resources that can be deployed.

2. Right-click at the level of the Web Application resource at which you want to create the security policy.

If you will be creating a security policy for *all* Web Applications (WARs), right-click Web Application Modules in the navigation tree. If you will be creating a security policy for a *particular* WAR or a component of a WAR (for example, a specific servlet or JSP), expand Web Application Modules, then right-click the name of a Web Application (WAR).

[Figure 5-10](#) illustrates where you might click, using the `basic-webapp` WAR as an example.

Figure 5-10 Deployments Portion of the Administration Console Navigation Tree



3. If you will be creating a security policy for *all* Web Applications (WARs), choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).

If you will be creating the security policy for a particular WAR or component of the WAR, follow these steps:

- a. Choose Define Security Policy...

- b. On the General tab, enter a URL pattern in the text field.

A URL pattern is a path to a specific servlet within a Web Application. Or, you can use `/*` to protect all servlets within the Web Application.

- c. Click the Define Security Policy... button to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).


Notice the `Caller is Granted the Role: Everyone` policy statement that the URL resource you selected has inherited from the default security policy associated with the URL resource type. If you proceed to [“Step 2: Create the Policy Conditions” on page 5-18](#), you will be overriding this default security policy. For more information, see [“Default Security Policies” on page 5-3](#) and [“Security Policy Granularity and Inheritance” on page 5-2](#).

4. If you will be creating a security policy for a particular WAR or component of a particular WAR, specify which method you want to protect, or select `ALL` to protect all methods.

Web Service Resources

1. In the left pane of the WebLogic Server Administration Console, expand Deployments.
2. If the Web Service is packaged as an EAR file, expand Applications, then expand the name of the application that contains the Web Service. The application expands to show the components that make up the application, including the Web Service Web Application.

If the Web Service is packaged as a stand-alone Web Application, expand Web Application Modules.

Note: The  icon designates a Web Service.

3. Right-click the Web Service.

If you will be creating a security policy for an entire Web Service, choose Define Security Policy... to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).

If you will be creating the security policy for a particular operation of the Web Service, follow these steps:

- a. Choose Define Policies and Roles for Individual services... to display a list of Web Services.
- b. Click the [Define Security Policies] link that corresponds to the particular Web Service you want to secure to display the Policy Editor page (see [Figure 5-3, “Policy Editor Page,” on page 5-9](#)).

Notice the `Caller is Granted the Role: Everyone` policy statement that the Web Service resource you selected has inherited from the default security policy associated with the Web Service resource type. If you proceed to [“Step 2: Create the Policy Conditions” on page 5-18](#), you will be overriding this default security policy. For more information, see [“Default Security Policies” on page 5-3](#) and [“Security Policy Granularity and Inheritance” on page 5-2](#).

4. If you will be creating a security policy for a particular operation of the Web Service, specify which method you want to protect, or select `ALL` to protect all methods.

Step 2: Create the Policy Conditions

1. In the Policy Condition list box, select one of the conditions. (For more information about the different policy conditions, see [“Components of a Security Policy: Policy Conditions, Expressions, and Policy Statements” on page 5-5](#).)

BEA recommends that you create expressions using the `Caller is Granted the Role` condition where possible. Basing expressions on security roles allows you to create one security policy that takes into account multiple users or groups, and is a more efficient method of management.

2. Click Add to display a customized window. Click Add to display a customized window. (See [Figure 5-11](#).)

Figure 5-11 Customized Window for Caller is Granted the Role Condition

Note: If you selected the `Server is in Development Mode` condition in step 1, no customized window appears. Rather, the completed expression appears in the Policy Statement list box. Therefore, you can skip to step 5.

3. If you selected the `Hours of Access are Between` condition, use the Time Constraint window to select start and end times, then click OK. The window closes and an expression appears in the Policy Statement list box. (See [Figure 5-12](#) for an example.)

Because the JMS subsystem performs its security check only once and this condition requires a subsequent security check, you should *not* use the `Hours of Access are Between` condition if you are securing a JMS resource.

If you are securing a Web Services resource, and you selected the `Element Requires Signature By` condition, follow these steps to fill out the Signature Predicate window:

- a. Select whether a group or user is required to sign the SOAP element using the Of Type drop-down box.
- b. Enter the name of the SOAP message element that must be digitally signed in the Signed by Type text box. Use the following format:

LocalPart:Namespace

where *LocalPart* refers to the name of the element in the SOAP message that must be digitally signed and *Namespace* refers to its namespace. Use the WSDL of the Web Service to get these values.

Caution: You can specify only those elements that have already been configured to be digitally signed in the `web-services.xml` deployment descriptor of the Web Service. For details, see “[Configuring Security](#)” in *Programming WebLogic Web Services*.

- c. Enter the name of the group or user that must sign the element in the Valued At field.
- d. Click OK. An expression appears in the Policy Statement list box. (See [Figure 5-12](#) for an example.)

If you selected one of the other conditions, follow these steps:

- a. Use the Users, Groups, or Roles window to enter the name of a user, group, or security role, then click Add. An expression appears in the list box.

You can repeat this step multiple times to add more than one user, group, or security role.

- b. If necessary, use the buttons located to the right of the list box to modify the expressions:

Move Up and Move Down change the ordering of the highlighted user or group name, and therefore the order in which they are evaluated. Change switches the highlighted `and` and `or` statements between expressions. Remove deletes the highlighted user or group name.

- c. Click OK to add the expression to the policy statement. The window closes and an expression appears in the Policy Statement list box. (See [Figure 5-12](#).)

Figure 5-12 Example Expression in Policy Statement List Box

Policy Condition:

- User name of the caller
- Caller is a member of the group
- Caller is granted the role**
- Hours of access are between
- Server is in Development Mode

Policy Statement:

- Caller is granted the role BankTeller

Buttons: Add, Move Up, Move Down, Change, Edit..., Remove

4. If needed, repeat steps 1 - 3 to add expressions based on different policy conditions.

5. If necessary, use the buttons located to the right of the Policy Statement list box to modify the expressions:
 - Move Up and Move Down change the ordering of the highlighted expression, and therefore the order in which they are evaluated.
 - Change switches the highlighted `and` and `or` statements between expressions.
 - Edit... reopens the customized window for the highlighted expression and allows you to modify the expression.
 - Remove deletes the highlighted expression.
 6. When all the expressions in the Policy Statement list box are correct, scroll down the page and click Apply.
- Note:** You can also click Reset at the bottom of the Policy Editor page to restore the page to its original state (that is, to undo any of your changes).

Modifying Security Policies

To modify a security policy for a WebLogic resource:

1. Navigate to the Policy Editor page for the WebLogic resource, as described in [“Step 1: Select the WebLogic Resource” on page 5-7](#).
Pay special attention to the Inherited Policy Statement list box to ensure that you understand which security policies you may be overriding.
2. Make your changes, using [“Step 2: Create the Policy Conditions” on page 5-18](#) as a guide.
3. Click Apply to save your changes.

Deleting Security Policies

To delete a security policy for a WebLogic resource:

1. Navigate to the Policy Editor page for the WebLogic resource, as described in [“Step 1: Select the WebLogic Resource” on page 5-7](#).
2. Click Delete to delete the entire security policy.
3. Click Apply to save your changes.

Index

A

- Administration Console
 - Check Roles and Policies field
 - instructions for changing 2-17
 - interaction with Future Redeploys field 2-18
 - purpose 2-15
 - Future Redeploys field
 - instructions for changing 2-17
 - interaction with Check Roles and Policies field 2-18
 - purpose 2-17
 - ways to create security roles 4-3
- Administrative resources
 - description 2-2
- Application resources
 - description 2-2
- attributes, MBean
 - protected 4-6

C

- Check Roles and Policies field
 - instructions for changing 2-17
 - interaction with Future Redeploys field 2-18
 - purpose 2-15
- COM resources
 - description 2-3
- conditions
 - policy 5-5
 - role 4-12
- configurations, security
 - copying

- cautions 2-20
 - reinitializing 2-27
- customer support contact information ix

D

- deployment descriptors
 - securing URL (Web) and EJB resources 2-13
- document audience 1-3
- documentation, where to find it viii

E

- EIS resources
 - description 2-2
- EJB resources
 - description 2-12
 - reasons for combined technique 2-19
 - securing
 - Administration Console technique 2-13
 - deployment descriptor technique 2-13
 - prerequisite settings 2-14
 - specifying technique in Administration Console 2-17
- expressions
 - definition 4-12, 5-6

F

- Future Redeploys field
 - instructions for changing 2-17
 - interaction with Check Roles and Policies field 2-18

purpose 2-17

G

global roles

- creating in Administration Console 4-3
- default 4-5
- default group associations 4-11
- definition 4-3

groups

- adding users to 3-3
- creating 3-6
- default 3-5
- default global role associations 4-11
- definition 3-2
- deleting 3-8
- difference from security roles 4-1
- modifying 3-7
- nesting 3-7

I

improving performance of WebLogic Security Service 2-15

J

JDBC resources

- description 2-3

JMS resources

- description 2-4

JNDI resources

- description 2-5

L

layered security for Server resources 2-6

- example 2-8
- maintaining consistency 2-10
- verification of 2-8

M

main steps for securing WebLogic resources 1-3

mapping, role

- definition 4-2

MBean protections

- Server resource use of 2-7

MBeans

- protected attributes and operations 4-6

O

operations, MBean

- protected 4-6

P

permissions

- for starting and shutting down servers 2-11
 - Node Manager 2-12
 - weblogic.Server 2-11

policies, security

- creating 5-7
- default 5-3
- definition 5-1
- deleting 5-21
- granularity 5-2
- inheritance 5-2
- modifying 5-21
- overriding 5-2
- prerequisites for use 5-2
- Server resource use of 2-7
- storage 5-2

policy conditions

- definition 5-5

policy statements

- definition 5-6
- use of and and or 5-6

prerequisite security settings

- defaults 2-18
- instructions for changing 2-17
- understanding interaction 2-18

- printing product documentation viii
- process for securing WebLogic resources 1-1
- protections, MBean
 - Server resource use of 2-7

R

- reinitializing security configurations 2-27
- resources
 - Administrative 2-2
 - Application 2-2
 - COM 2-3
 - EIS 2-2
 - JDBC 2-3
 - JMS 2-4
 - JNDI 2-5
 - Server 2-5
 - layered security scheme 2-6
 - URL (Web) and EJB 2-12
 - prerequisite security settings 2-14
 - reasons for using combined technique 2-19
 - reinitializing security configurations 2-27
 - securing in Administration Console 2-13
 - securing with deployment descriptors 2-13
 - specifying technique for securing 2-17
 - techniques for securing 2-13
- Web Service 2-29
- WebLogic
 - hierarchical nature 5-2
 - main steps for securing 1-3
 - process for securing 1-1
 - role of security providers in securing 5-2
- roles
 - conditions
 - definition 4-12
 - global

- creating in Administration Console 4-3, 4-14
- default 4-5
- definition 4-3
- deleting 4-18
- group associations 4-11
- modifying 4-17
- mapping 4-2
- scoped
 - creating in Administration Console 4-4, 4-19
 - definition 4-3
 - deleting 4-31
 - modifying 4-30
- security
 - creating in Administration Console 4-3
 - definition 4-1
 - deleting 4-31
 - difference from groups 4-1
 - dynamically granting 4-2
 - types 4-3
- statements
 - definition 4-13
 - use of and and or 4-13

S

- scoped roles
 - creating in Administration Console 4-4
 - definition 4-3
 - deleting 4-31
 - modifying 4-30
- security configurations
 - copying
 - cautions 2-20
 - reinitializing 2-27
- security policies
 - creating in Administration Console 5-7
 - default 5-3
 - definition 5-1
 - deleting 5-21

- granularity 5-2
- inheritance 5-2
- modifying 5-21
- overriding 5-2
- prerequisites for use 5-2
- Server resource use of 2-7
- storage 5-2
- security providers
 - use in securing WebLogic resources 5-2
- security roles
 - creating in Administration Console 4-3
 - default global 4-5
 - group associations 4-11
 - definition 4-1
 - difference from groups 4-1
 - dynamically granting 4-2
 - global
 - creating in Administration Console 4-3, 4-14
 - default 4-5
 - definition 4-3
 - deleting 4-18
 - group associations 4-11
 - modifying 4-17
 - scoped
 - creating in Administration Console 4-4, 4-19
 - definition 4-3
 - deleting 4-31
 - modifying 4-30
 - types 4-3
- Server resources
 - description 2-5
 - layered security scheme 2-6
 - example 2-8
 - maintaining consistency 2-10
 - verification of 2-8
 - use of MBean protections 2-7
 - use of security policies 2-7
- servers

- permissions for starting and shutting down
 - 2-11
 - Node Manager 2-12
 - weblogic.Server 2-11
- statements
 - policy
 - definition 5-6
 - use of and and or 5-6
 - role
 - definition 4-13
 - use of and and or 4-13
- support
 - technical ix

U

- URL (Web) resources
 - description 2-12
 - reasons for using combined technique 2-19
 - securing
 - Administration Console technique 2-13
 - deployment descriptor technique 2-13
 - prerequisite security settings 2-14
 - specifying technique in Administration Console 2-17
- users
 - adding to groups 3-3
 - creating 3-2
 - definition 3-2
 - deleting 3-4
 - modifying 3-4

W

- Web Service resources
 - description 2-29
- WebLogic resources
 - Administrative 2-2
 - Application 2-2
 - COM 2-3
 - EIS 2-2
 - hierarchical nature 5-2

- JDBC 2-3
- JMS 2-4
- JNDI 2-5
- securing
 - main steps 1-3
 - process description 1-1
 - role of security providers 5-2
 - techniques for URLs and EJBs 2-13
- Server 2-5
 - layered security scheme 2-6, 2-8, 2-10
 - use of MBean protections 2-7
 - use of security policies 2-7
- URL (Web) and EJB 2-12
- Web Service 2-29
- WebLogic Security Service
 - improving performance 2-15
 - verification of layered security for Server resources 2-8