Oracle® Fusion Middleware

Oracle Authorization Policy Manager Administrator's Guide
(Oracle Fusion Applications Edition)

11*g* Release 6 (11.1.6)

E20839-04

January 2013

This guide explains the features, configuration, and use of Oracle Authorization Policy Manager, a graphical interface used to manage fine-grained policies and the security artifacts used to create them.

ORACLE®

Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition), 11*g* Release 6 (11.1.6)

E20839-04

Primary Author: Trish Fuzesy

Contributing Authors: Carlos Subi, Michael Teger

Contributor: Akila Natarajan

Contents

2 Understanding The Policy Model

3 Managing Policies and Policy Objects

4 Searching for Security Objects

5 Configuring Predefined Attribute Retrievers

6 Delegating With Administrator Roles

7 Upgrading Oracle Fusion Applications Policies

8 Customizing the User Interface

9 Managing Policy Distribution

10 Oracle Fusion Applications Data Role Templates

11 Managing Oracle Fusion Applications Data Security Policies

12 Managing System Configurations

13 Management Tasks

A Using an OpenLDAP Identity Store

B Troubleshooting Oracle Authorization Policy Manager

C Configuration Parameters

Index

List of Examples

List of Figures

List of Tables

Preface

The *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)* explains the features, configuration, and use of Oracle Authorization Policy Manager, a graphical interface used to manage fine-grained policies and the security artifacts used to create them. The information is relevant to the version of Oracle Entitlements Server, a fine-grained authorization and access control product, released with Oracle Fusion Applications Patch Set 1 only. This Preface addresses the following topics:

- Audience
- Documentation Accessibility
- Related Documentation
- Conventions

Audience

The intended audience of this guide is security administrators.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/support/contact.html` or visit `http://www.oracle.com/accessibility/support.html` if you are hearing impaired.

Related Documentation

Information about security administration is also found in the following documents:

- *Oracle Fusion Applications Administrator and Implementor Roadmap*
- *Oracle Fusion Applications Administrator's Guide*
- *Oracle Fusion Applications Enterprise Deployment Guide*

For a comprehensive list of Oracle documentation or to search for a particular topic within Oracle documentation libraries, see `http://www.oracle.com/technology/documentation/index.html`.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action.
italic	Italic type indicates book titles, emphasis, terms defined in text, or placeholder variables for which you supply particular values.
`monospace`	Monospace type within a paragraph indicates commands, URLs, Java class names and method names, file and directory names, text that appears on the screen, or text that you enter.

1 Getting Started With Oracle Authorization Policy Manager

Oracle Authorization Policy Manager is a graphical interface for managing policies and related security objects. This chapter describes a general overview of the product.

- Understanding Authorization Policy Manager
- Installing and Configuring Authorization Policy Manager
- Accessing the Authorization Policy Manager Administration Console
- Navigating the Authorization Policy Manager Administration Console

1.1 Understanding Authorization Policy Manager

Oracle Authorization Policy Manager is the graphical interface for Oracle Entitlements Server, a fine-grained authorization product that allows an organization to protect its resources by defining and managing policies and related security objects. From a high-level, Oracle Entitlements Server comprises centralized policy management with policy decision making. The following sections contain more information.

- Section 1.1.1, "What is Oracle Entitlements Server?"
- Section 1.1.2, "Using the Authorization Policy Manager Console"
- Section 1.1.3, "Putting It Together"

1.1.1 What is Oracle Entitlements Server?

Oracle Entitlements Server offers fine-grained authorization in which a context for the authorization request is provided and access is granted or denied based on it. Access privileges are defined in an *authorization policy* by specifying who can do what to which resource, when it can be done, and how. The authorization policy can enforce controls on all types of resources including software components (URLs, Java Server Pages, Enterprise JavaBeans, methods, servlets and the like used to construct an application) and business objects (representations of user accounts, personal profiles and contracts such as bank accounts in a banking application, patient records in a health care application, or anything used to define a business relationship). Additionally, Oracle Entitlements Server:

- Distributes policies from the Administration Server to the decision endpoints.
- Caches policies and authorization decisions for performance.
- Updates security policies at run time.

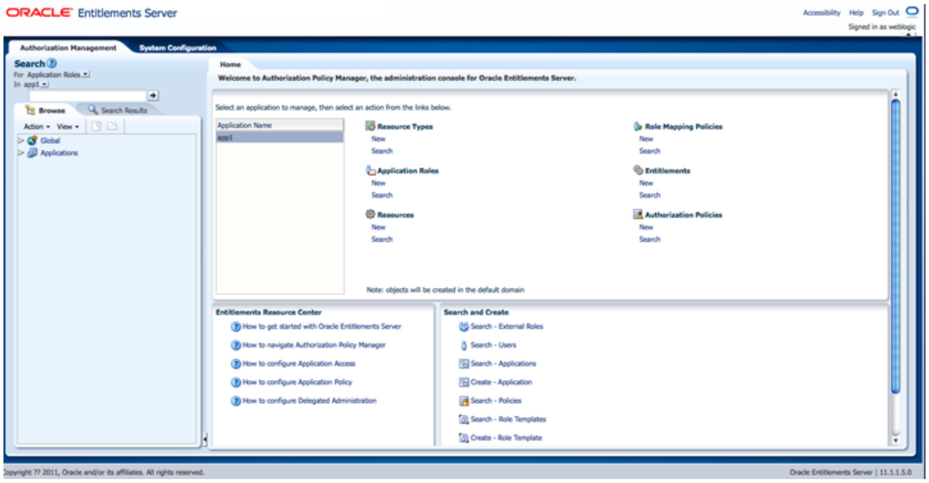
- Offers a flexible architecture that supports both embedded and remote decision points (for centralized or distributed policy decisions).
- Separates security decision making from application logic.
- Audits all access decisions and management operations.
- Supports the eXtensible Access Control Markup Language (XACML) request/response protocol for authorization inquiries.
- Integrates with existing security and identity systems by leveraging enterprise data in relational databases and LDAP directories.

Note: The Fusion Applications version of Oracle Entitlements Server is not meant to contain all of the functionality of Oracle Entitlements Server.

1.1.2 Using the Authorization Policy Manager Console

Oracle Authorization Policy Manager is the Administration Console for Oracle Entitlements Server. It is a browser-based, graphical interface for managing policies and related security objects. It supports the creation and management of Authorization Policies and Role Mapping Policies. An Authorization Policy defines the rules for accessing a software component or business object. A Role Mapping Policy defines which users are assigned which roles, and may be referenced by an Authorization Policy. Figure 1–1 is a screenshot of the Authorization Policy Manager Administration Console.

Figure 1–1 The Oracle Authorization Policy Manager Graphical Interface



Welcome to Authorization Policy Manager, the administration console for Oracle Entitlements Server.

Select an application to manage, then select an action from the links below.

Note: objects will be created in the default domain

Copyright ?? 2011, Oracle and/or its affiliates. All rights reserved.

Note: For purposes of this documentation, Authorization Policy Manager and variations of the Oracle Entitlements Server Administration Console (Administration Console, Console and the like) may be used interchangeably.

The following sections contain additional information.

- Section 1.1.2.1, "Assigning Administrators"

- Section 1.1.2.2, "Using the Identity Store"
- Section 1.1.2.3, "Using the Policy Store"

1.1.2.1 Assigning Administrators

Only users with sufficient privileges can log in to the Oracle Entitlements Server Administration Console or use administrative command-line tools such as the WebLogic Scripting Tool (WLST). An Oracle Entitlements Server system-level Administrator Role named `SystemAdmin` is created during installation and is mapped to the WebLogic Server administrator user, `weblogic`. The password is set during installation.

> **Note:** At first log in to the Oracle Entitlements Server Administration Console, `SystemAdmin` must use the credentials set during installation. The identifier and password can then be changed by using your identity store's management tool.

`SystemAdmin` has extensive privileges that includes the rights to create additional Administrative Roles for delegating administrative rights to others. You can than grant users these roles to give them different administrative rights for the Oracle Entitlements Server environment. For more information, see Section 6.6, "Managing System Administrators Using Administrator Roles."

1.1.2.2 Using the Identity Store

Oracle Entitlements Server administrator and user identities are stored in an identity store, typically an LDAP directory server. Users and External Roles (those defined in the identity store) are read-only. Oracle Entitlements Server reads and displays the data; it does not perform any management operations. Management of the identity data is accomplished using the identity store's tools or an identity management product such as Oracle Identity Manager. Supported identity stores are:

- Oracle Internet Directory
- Oracle Virtual Directory
- WebLogic EmbeddedLDAP
- Sun Java System Directory Service version 6.3
- Active Directory 2003, 2008
- Novell eDirectory 8.8
- OpenLDAP 2.2. For the special configuration required for this type, see Appendix A, "Using an OpenLDAP Identity Store."
- Tivoli Directory Server

For information about Oracle Fusion Middleware Certification and Supported Configurations, visit `http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html`.

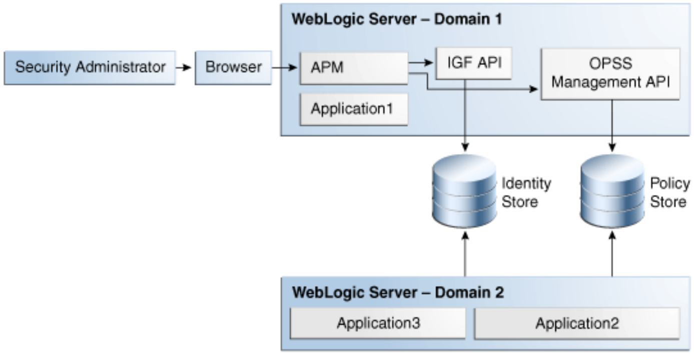
1.1.2.3 Using the Policy Store

For this release of Oracle Entitlements Server, the policy store used to maintain policy objects and defined policies can be a relational database (preferred) or an LDAP-based directory. (Oracle Internet Directory can be used as the policy store but has limited capabilities.) For links regarding hardware requirements, see Section 1.2.1, "Before You Begin." Instructions for creating and initializing the policy store can be found in *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*. Before using Authorization Policy Manager, make sure that the policy store has been reassociated to one of the supported repositories. For details on reassociating the domain policy store, see *Oracle Fusion Middleware Application Security Guide*.

1.1.3 Putting It Together

Figure 1–2 illustrates how a security administrator accesses Authorization Policy Manager, and how the tool communicates with the Oracle WebLogic server domain's policy and identity stores. Note that Authorization Policy Manager can access policies (and identities) shared by different domains. Authorization Policy Manager uses the Oracle Entitlements Server Management API to access the policy store and Oracle Identity Governance Framework API to access the identity store.

Figure 1–2 Authorization Policy Manager Deployed in a WebLogic Domain



1.2 Installing and Configuring Authorization Policy Manager

Before getting started using Oracle Entitlements Server, the following tasks must be done. They include installing the product and its components (for example, remote Security Modules), and configuring features like high availability and Secure Sockets Layer (SSL), if applicable. The following sections contain links to other documentation regarding the topics.

- Section 1.2.1, "Before You Begin"
- Section 1.2.2, "Installing Authorization Policy Manager"
- Section 1.2.3, "Changing From Basic to Advanced Policy Authorization"
- Section 1.2.4, "Reconfiguring the Default Identity Store"
- Section 1.2.5, "Configuring High Availability"
- Section 1.2.6, "Connecting with Secure Sockets Layer"
- Section 1.2.7, "Setting Loggers"
- Section 1.2.8, "Displaying Text in Foreign Languages"

1.2.1 Before You Begin

Two particular data sources must be set using WebLogic Server before beginning the installation process. They are `APMDBDS` and `mds-ApplicationMDSDB`. The first data source can be configured with the WebLogic Console by navigating to **JDBC > Data Sources**. Table 1–1 describes the characteristics of these data sources.

Table 1–1 Required Data Sources

Data Source Name	JNDI Name	Description
mds-ApplicationMDSDB	jdbc/mds/mds-ApplicationMDSDBDS	Stores MDS-related documents used by the application.
APMDBDS	jdbc/APMDBDS	Required to use the 3-way-diff patch method.

Additionally, applications whose policies are managed with Authorization Policy Manager are assumed to use Oracle Platform Security Services for authorization. For details about integrating an application with these services, see *Oracle Fusion Middleware Application Security Guide*.

1.2.2 Installing Authorization Policy Manager

Authorization Policy Manager is installed with Oracle Entitlements Server. Before getting started, install Oracle Entitlements Server and its components (for example, remote Security Modules), if applicable. For details about installation, see *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.

- For this release, the policy store managed by Oracle Entitlements Server can be a relational database (preferred) or an LDAP-based directory.
- The identity store associated with Oracle Entitlements Server must be an LDAP-based directory.

1.2.3 Changing From Basic to Advanced Policy Authorization

When Authorization Policy Manager is installed for Oracle Fusion Applications, it is configured to allow for basic authorization. Basic authorization is based on the permissions policy model in which permissions are granted to users, groups, and code sources. For users and groups, the permissions determine what a user or a group member is allowed to access. For code sources, they determine what actions the code is allowed to perform. Advanced authorization allows for the use of more fine-grained policy objects including Role Mapping Policies and hierarchical resources.

Use the following procedure to reconfigure Authorization Policy Manager for advanced authorization. It assumes WebLogic Server is installed and the Fusion Middleware home directory is available.

1. Change to the Fusion Middleware home directory at `$FMW_HOME/oracle_common/common/bin`.
2. Connect to the WebLogic Server using the `wlst.sh` command.

   ```
   connect ('weblogic','weblogic1','t3://localhost:7101')
   ```

   The command takes the following arguments: user name (weblogic), password associated with the user (weblogic1) and the T3 connection URL for the Administration Server. In this example, it is running locally on port 7101.
3. Export the Authorization Policy Manager configuration file using the `exportMetadata` command.

   ```
   exportMetadata(application='oracle.security.apm', server='AdminServer',
    toLocation='/tmp/repository/',
   ```

```
 docs='/oracle/security/apm/config/apm-config.xml')
```

This command will export the `apm-config.xml` configuration file to the `/tmp/repository` sub-directory on the machine that hosts the Administration Server. The command takes the following arguments:

- The application owner of the document being exported; in this case, the default value is `oracle.security.apm`.
- The name of the WebLogic Server Administration Server.
- The directory to which `apm-config.xml` will be exported. Be sure you have access rights to this directory.
- The document being exported; in this example, `oracle/security/apm/config/apm-config.xml` will be exported, so please make sure you have access right to that file path in the AdminServer machine once you downloaded the documents

4. Open the `apm-config.xml` configuration file in a text editor.

   The file is in the `/tmp/repository` directory as previously specified.

5. Change the value of the `oracle.security.apm.oes.mode` attribute in this file from *basic* to *advanced*.
6. Save the changes and close the file.
7. Upload the modified file back to the repository.

   ```
   importMetadata(application='oracle.security.apm', server='AdminServer',
    fromLocation='/tmp/repository/',
    docs='/oracle/security/apm/config/apm-config.xml')
   ```

   This command will import the `apm-config.xml` configuration file back to the machine that hosts the Administration Server. The command takes the following arguments:

   - The application owner of the document being imported; in this case, the default value is `oracle.security.apm`.
   - The name of the WebLogic Server Administration Server.
   - The directory from which `apm-config.xml` will be imported.
   - The document being imported; in this example, `apm-config.xml` will be imported to the `oracle/security/apm/config/` directory.

8. Issue the exit command.

   ```
   exit()
   ```

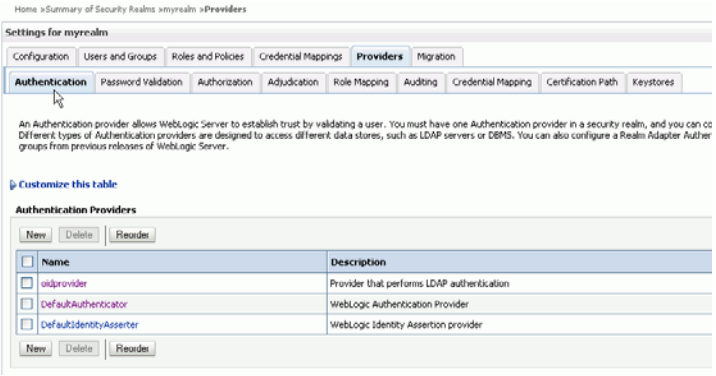
1.2.4 Reconfiguring the Default Identity Store

After installation, the Oracle Entitlements Server identity store is associated with the WebLogic Server embedded LDAP directory. While this embedded LDAP directory is fine for development purposes, a supported LDAP directory must be used in production. The following procedure reconfigures the default identity store settings. More specific information on configuring LDAP authentication providers can be found in the *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

1. Launch the WebLogic Server console.
2. Click Security Realms.

3. Click the settings for *myrealm*.
4. Click the Provider tab.
5. Click the Authentication tab as displayed in Figure 1–3.

Figure 1–3 The Authentication Provider Tab


Home >Summary of Security Realms >myrealm >**Providers**

Settings for myrealm

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can co
Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Auther
groups from previous releases of WebLogic Server.

Authentication Providers

□	Name	Description
□	oidprovider	Provider that performs LDAP authentication
□	DefaultAuthenticator	WebLogic Authentication Provider
□	DefaultIdentityAsserter	WebLogic Identity Assertion provider

6. Click the New button to create a new provider.
7. Enter a name and select the type of LDAP-based directory.

   For example, *OracleInternetDirectoryAuthenticator*.
8. Configure the provider-specific attributes of the LDAP-based directory.

   This might include the host name and port, credentials, group search base, user search base and the like.
9. Save the provider information.
10. Change the order of the providers so that the LDAP-based directory is first.

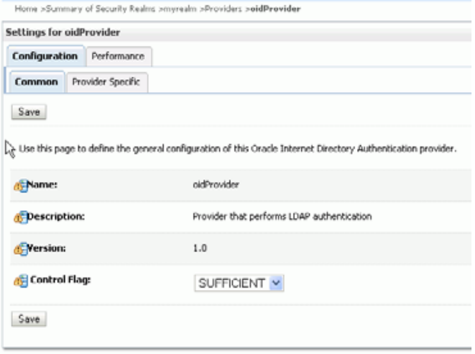
    DefaultAuthenticator and *DefaultIdentityAsserter* will follow.
11. Click the new provider name to configure it.

    a. Click the Configuration tab.

    b. Click the Common tab.

    c. Set the Control Flag to SUFFICIENT and click Save as displayed in Figure 1–4.

Figure 1–4 SUFFICIENT Control Flag


Home >Summary of Security Realms >myrealm >Providers >**oidProvider**

Settings for oidProvider

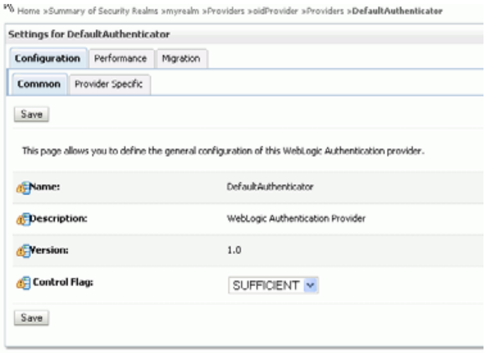
Use this page to define the general configuration of this Oracle Internet Directory Authentication provider.

Name:	oidProvider
Description:	Provider that performs LDAP authentication
Version:	1.0
Control Flag:	SUFFICIENT

   d. Click the Provider Specific tab.
   e. Enter the LDAP configuration information for your identity store and click Save.

12. Return to the Providers tab.
13. Click *DefaultAuthenticator* to change its configuration.
14. Set the Control Flag to SUFFICIENT and click Save as displayed in Figure 1–5.

Figure 1–5 DefaultAuthenticator Tab in WebLogic Server Console


Home »Summary of Security Realms »myrealm »Providers »oidProvider »Providers »DefaultAuthenticator
Settings for DefaultAuthenticator
Configuration
Performance
Migration
Common
Provider Specific
Save
Save

This page allows you to define the general configuration of this WebLogic Authentication provider.

Name:	DefaultAuthenticator
Description:	WebLogic Authentication Provider
Version:	1.0
Control Flag:	SUFFICIENT

15. Restart WebLogic Server.

1.2.5 Configuring High Availability

For details about high availability for Authorization Policy Manager, see *Oracle Fusion Middleware High Availability Guide*.

1.2.6 Connecting with Secure Sockets Layer

The connections that Authorization Policy Manager establishes with the policy store, the identity store, and the database can be secured through one-way Secure Sockets Layer (SSL). The access to Authorization Policy Manager via a browser can also be secured through one-way SSL. These settings are similar to those of any other application running in the Oracle WebLogic server.

For information about configuring one-way SSL for connections with the policy store, the identity store, and the database, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*. Access to Oracle Entitlements Server using a browser can also be secured through one-way SSL. These settings are similar to those of any other application running in the Oracle WebLogic Server.

- For details about configuring SSL in Oracle Fusion Middleware applications when OHS is not being used, see chapter 12 in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.
- For details about configuring SSL in Oracle Fusion Middleware applications when OHS is being used, see chapter 6 in *Oracle Fusion Middleware Administrator's Guide*.

1.2.7 Setting Loggers

Setting the loggers and a log level for Authorization Policy Manager is similar to setting them for any other application running in the Oracle WebLogic server. For details, see *Oracle Fusion Middleware Application Security Guide*.

1.2.8 Displaying Text in Foreign Languages

Oracle Authorization Policy Manager supports Globalization to display text, such as role and user names, in one of the following standard administrator languages:

- English
- Portuguese
- French
- German
- Italian
- Japanese
- Korean
- Spanish
- Chinese

Authorization Policy Manager determines the language in which text is displayed from the locale setting of the browser. When your browser locale is set to one of the above supported administrator languages, the Authorization Policy Manager text is displayed in that language.

1.3 Accessing the Authorization Policy Manager Administration Console

The following sections contain information on how to access the Authorization Policy Manager graphical interface (also referred to as the Administration Console).

- Section 1.3.1, "Signing In to the Administration Console"
- Section 1.3.2, "Signing Out of the Administration Console"

1.3.1 Signing In to the Administration Console

Follow this procedure to sign in to the Authorization Policy Manager Administration Console.

1. Enter the Authorization Policy Manager Administration Console URL in the address bar of your browser. For example:

   https://hostname:port/apm/

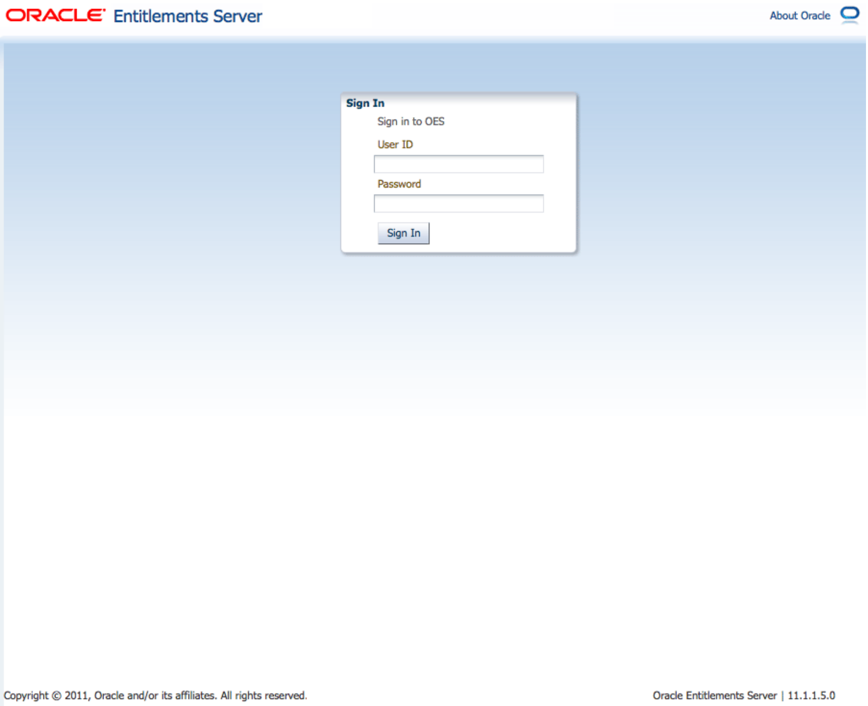
   where:

   - HTTPS represents the Hypertext Transfer Protocol (HTTP) with the Secure Socket Layer (SSL) enabled to encrypt and decrypt user page requests and the pages returned by the Web server.
   - *hostname* refers to the fully qualified domain name of the computer hosting the Oracle Authorization Policy Manager Administration Console.
   - *port* refers to the designated bind port for the Authorization Policy Manager Administration Console. (This is the same as the bind port for the WebLogic Server Administration Console.)
   - `/apm/` refers to the Authorization Policy Manager Log In page

2. Enter the System Administrator credentials.

The default system administrator identifier is `weblogic`. The password is the same one supplied during installation. Figure 1–6 is a screenshot of the Sign In page.

Figure 1–6 Administration Console Sign In Page


ORACLE Entitlements Server
About Oracle
Sign In
Sign in to OES
User ID
Password
Sign In
Copyright © 2011, Oracle and/or its affiliates. All rights reserved.
Oracle Entitlements Server | 11.1.1.5.0

3. Click **Sign In**.

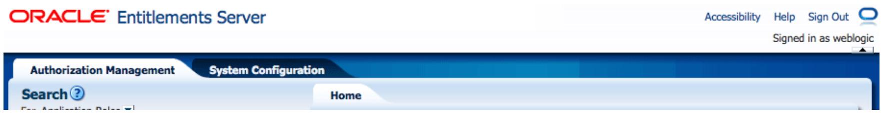
1.3.2 Signing Out of the Administration Console

Follow this procedure to sign out of the Authorization Policy Manager Administration Console.

1. Click the **Sign Out** link located in the upper right corner of the Administration Console.

   Figure 1–7 is a screenshot of the Sign Out link.

Figure 1–7 Administration Console Sign Out Link



2. Close the browser window.

1.4 Navigating the Authorization Policy Manager Administration Console

After a successful log in, the Authorization Policy Manager Administration Console is displayed with the Authorization Management Tab active. The Navigation Panel is on the left side and the Home area on the right side. Objects selected in the Navigation

Panel are opened in tabs and displayed in the Home area. Figure 1–1 is a screenshot of the Administration Console after an administrative user has successfully signed in. The following sections contain descriptions of the top-level items displayed.

- Section 1.4.1, "Understanding the Main Tabs"
- Section 1.4.2, "Using The Navigation Panel"
- Section 1.4.3, "Understanding the Home Area"
- Section 1.4.4, "Accessing Help"

1.4.1 Understanding the Main Tabs

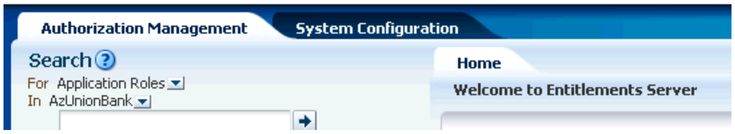
See the following sections for information on the organizational tabs used in the Administration Console. Each tabbed section is comprised of a Navigation Panel and Home area.

- Section 1.4.1.1, "Authorization Management Tab"
- Section 1.4.1.2, "System Configuration Tab"

1.4.1.1 Authorization Management Tab

The Authorization Management tab is used to search and manage policy objects. This tab is active upon successful log in to the Administration Console. Figure 1–8 is a screenshot of the Authorization Management tab.

Figure 1–8 Authorization Management Tab

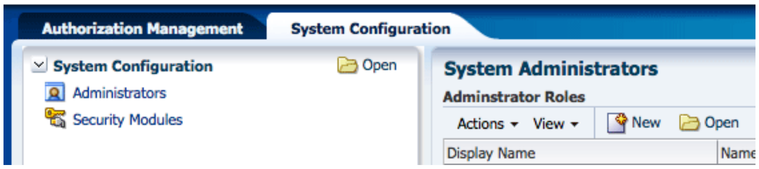


Under Authorization Management, the left side is the Navigation Panel and the right side is Home. The Home display changes based on what is selected from the Navigation Panel. For more information, see Section 1.4.2, "Using The Navigation Panel" and Section 1.4.3, "Understanding the Home Area."

1.4.1.2 System Configuration Tab

The System Configuration tab is used to manage administrative and system type objects for the Oracle Entitlements Server deployment. Figure 1–9 is a screenshot of an active System Configuration tab. The object selected in the Navigation Panel is displayed using tabs in the Home area.

Figure 1–9 System Configuration Tab



The following tasks are performed under System Configuration:

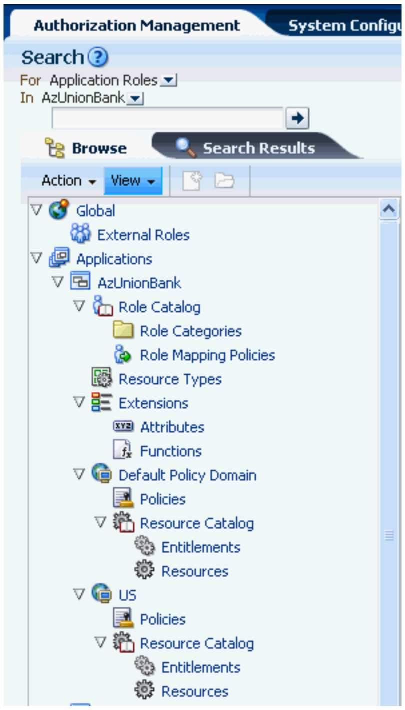
- Creating Security Modules
- Binding Security Modules to applications
- Managing system administrators (for example, creating additional system administrator roles, assigning users to system administrator roles, and assigning rights to system administrator roles)

For more information, see Chapter 12, "Managing System Configurations."

1.4.2 Using The Navigation Panel

The Navigation Panel is used to find security objects by browsing the Global or Applications information trees, or by conducting a simple search. It lists all Global and Application policy objects in a navigable tree. You can browse the tree or display objects as Search Results based on defined search criteria. Figure 1–1 is a screenshot that displays the Navigation Panel with its nodes collapsed. Figure 1–10 displays the Navigation Panel with its nodes expanded and many policy objects in view.

Figure 1–10 Navigation Panel Browse Tab with Nodes Expanded



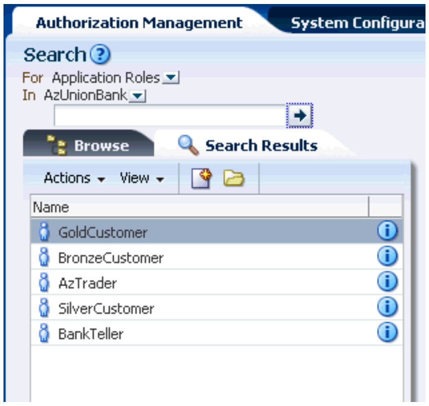
The Navigation Panel contains, from top to bottom, the following elements:

- A pull-down list to select the policy object for a simple search. For more information, see Section 4.2, "Finding Objects with a Simple Search."
- A pull-down list to select the scope of a simple search. For more information, see Section 4.2, "Finding Objects with a Simple Search."
- A text box to enter the simple search string. The string is compared against both the Name and Display Name of policy objects; those that match are displayed in the Search Results tab.
- The **Browse** tab displays the following expandable and collapsible nodes:

  - The **Global** node collects global objects such as external roles.
  - The **Applications** node contains one or more Applications being managed by the administrator that is logged in. (Only Applications which the logged in user is authorized to access are displayed.) From any of those displayed, the administrator can access application-specific policy objects such as resource types, entitlements, resources, policies, and roles. For more information, see Chapter 12, "Managing System Configurations."

- The **Search Results** tab displays the results of the last simple search as seen in Figure 1–11.
- Actions and View drop downs to select operations on the chosen policy object.

Figure 1–11 Navigation Panel Search Tab



For Application Roles

In AzUnionBank

Name
GoldCustomer
BronzeCustomer
AzTrader
SilverCustomer
BankTeller

From the Navigation Panel, there are two methods for displaying the **New** and **Open** options comprised in the Actions drop-down list.

- Locate the desired application, expand the node, and select the desired object. Click the Actions drop-down and select New.
- Locate the desired application, expand the node, and select the desired object. Right-click the object from the application node.

Select **New** to create a new object of the same type and select **Open** to display a search tab in the Home area. Double-clicking an object from the node also opens a Search tab in the Home area.

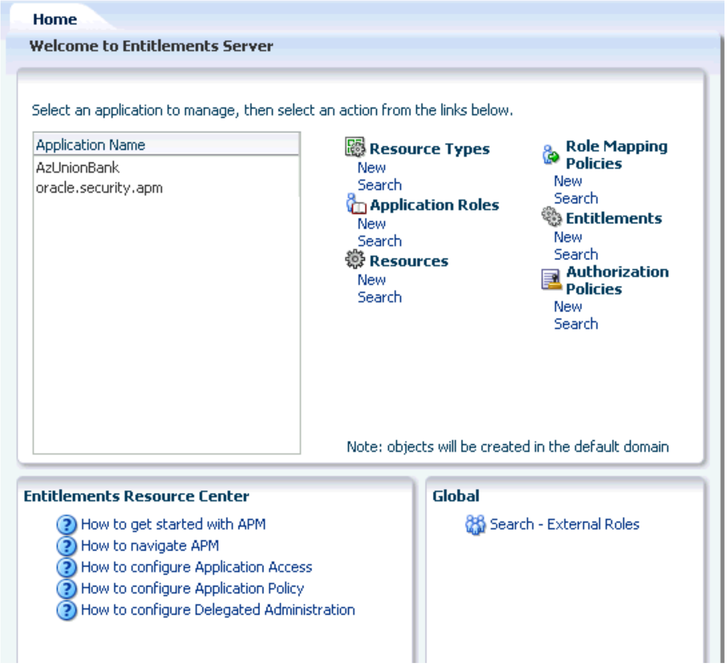
1.4.3 Understanding the Home Area

The Home area displays on the right side of the Navigation Panel and contains quick access links to New and Search screens for the most commonly used policy objects. As displayed in Figure 1–12, the Home area of the Administration Console is divided into the following sections.

- The **Application** area is the upper region of the Home area. The Application Name pane displays all applications available to the logged in user. To the right of this pane are links to screens for performing common operations such as creating new policy objects (entitlements, resources, resource types, application roles, and authorization policies) or searching defined policy objects.
- The **Global** section is the lower right region of the Home area. This section is for objects shared across all applications and includes external role search.

- The **Entitlements Resource Center** section is the lower left region of the Home area. It contains links to information regarding the most commonly used procedures.

Figure 1–12 The Home Area



Select an application to manage, then select an action from the links below.

Application Name
AzUnionBank
oracle.security.apm

Note: objects will be created in the default domain

Entitlements Resource Center
How to get started with APM
How to navigate APM
How to configure Application Access
How to configure Application Policy
How to configure Delegated Administration
Global
Search - External Roles

1.4.4 Accessing Help

To get more information while using the Administration Console, click the Help link located in the upper right corner (as seen in Figure 1–1). A separate window opens. From this window you can access both the online help and an embedded version of this book in HTML. After the window displays, select either Administration Console *Online Help* or *Authorization Policy Manager Administrator's Guide* from the drop-down Book list. The help topics link to the corresponding section of the embedded book as do the links in the Help's Table of Contents.

2 Understanding The Policy Model

A policy is a set of criteria that, when evaluated by Oracle Entitlements Server, specifies whether a user will be granted access to a particular protected resource or assignment to a particular role. This chapter contains an overview of the Oracle Entitlements Server policy model, the elements that comprise a policy and how the elements are organized in the policy store. It contains the following sections:

- Understanding Oracle Entitlements Server Policies
- How Oracle Entitlements Server Evaluates Policies
- The Policy Object Glossary
- Implementing a Policy Use Case

2.1 Understanding Oracle Entitlements Server Policies

Oracle Entitlements Server supports the creation of Authorization Policies and Role Mapping Policies. Section 2.1.1, "Granting and Denying Access Using Policies" explains how these two types of policies work together to grant or deny access to a protected resource. The referenced sections below contain detailed information regarding these policy types including how they are used.

- An *Authorization Policy* defines the criteria that control access to an organization's protected resources. See Section 2.1.2, "Understanding the Authorization Policy."

> **Note:** Resources may include software components or business objects. For more information, see Section 2.4, "Implementing a Policy Use Case."

- A *Role Mapping Policy* defines the criteria that control how external users, groups or roles are granted or denied membership to roles created using Oracle Entitlements Server. See Section 2.1.3, "Understanding Role Assignments and the Role Mapping Policy."

2.1.1 Granting and Denying Access Using Policies

By default, all access to a protected resource is denied until an Authorization Policy that explicitly grants access to it is written and distributed. If the Authorization Policy only grants access to a role, the requesting user must be statically assigned to it or a Role Mapping Policy that assigns the requestor (or not) to the defined role must be written and distributed. If an Authorization Policy denies a previously granted entitlement, it takes precedence over the previous grant. Explicit DENY Authorization Policies cannot be overruled. A practical use of a DENY policy is to explicitly deny an entitlement to ensure that a user or group can never gain access to a specific resource.

2.1.2 Understanding the Authorization Policy

An *Authorization Policy* is created to grant or deny access to a particular resource based on the profile of the requesting user. From a high level, the Authorization Policy defines an association between an effect (GRANT or DENY), a principal (requesting user), the target resource, the resource's allowed actions and an optional condition. An Authorization Policy is applicable to a request for access if the parameters in the request match those specified in the policy. Consider this Authorization Policy definition:

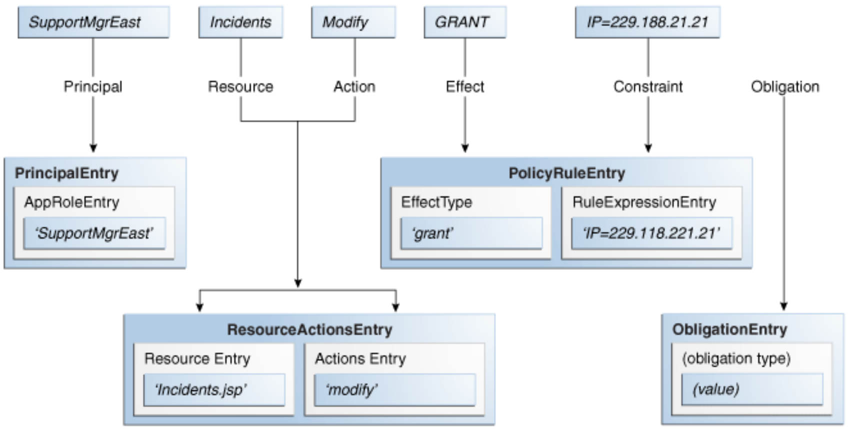
```
GRANT the SupportManagerEast role MODIFY access to the Incidents servlet
  if the request is made from an IP address of 229.188.21.21
```

This Authorization Policy will GRANT any user that is a member of the `SupportManagerEast` role access to the Incidents servlet for the purpose of modifying it. The policy is also constrained by a condition - the request must be made from IP address 229.188.21.21. Thus, if the parameters in the request match the parameters in the policy (a member of the `SupportManagerEast` role wants to modify the Incidents servlet), and the request is made from IP address 229.188.21.21, the request is granted. If the parameters in the request match the parameters in the policy (a member of the `SupportManagerEast` role wants to modify the Incidents servlet) but the request is NOT made from IP address 229.188.21.21, the policy is ignored. The following list of terms and values are extracted from this policy definition and comprise the components of the Authorization Policy.

- Effect: GRANT
- Action: MODIFY
- Resource: Incidents servlet
- Principal: member of SupportManagerEast role
- Condition/Constraint: IP address 229.188.21.21

Figure 2–1 illustrates how the components of this policy map to the Oracle Entitlements Server Authorization Policy objects.

Figure 2–1 Policy Components Mapped to Authorization Policy Objects



For information on how to create, update and delete Authorization Policies, see Section 3.5.5, "Managing Authorization Policies."

2.1.3 Understanding Role Assignments and the Role Mapping Policy

An *Application Role* is a collection of users, groups, or other Application Roles defined using Oracle Entitlements Server. It can be mapped to an enterprise user, group, or external role in an identity store, or another Application Role in the Oracle Entitlements Server policy store. (Assigning one Application Role to another Application Role allows you to build an Application Role hierarchy.) Application Roles can be assigned to a user in either of the following ways:

- By statically granting a specific user membership in the role.
- By referencing the Application Role in a *Role Mapping Policy* that will be used to dynamically assign role membership at runtime.

A Role Mapping Policy allows you to dynamically assign (GRANT) role membership to a user or dynamically revoke (DENY) role membership from a user. Consider the following Role Mapping Policy definition:

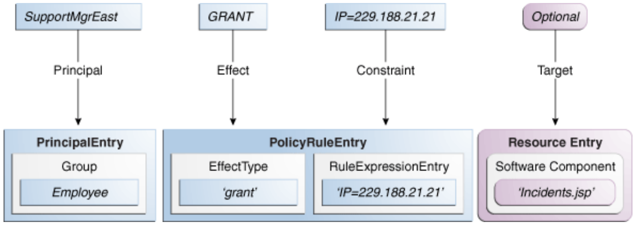
```
GRANT the Employee group application role SupportManageEast
   if the request is made from an IP address of 229.188.21.21
```

This policy grants the `SupportManagerEast` Application Role to any user that is a member of the group `Employee`. The policy is constrained by a condition though - the request must be made from IP address 229.188.21.21. Thus, if the parameters in the request match the parameters in the Role Mapping Policy (the requesting user is a member of the `Employee` group), and the request is made from IP address 229.188.21.21, the Application Role is granted. If the request is not made from the defined IP address, the Role Mapping Policy is ignored. The following terms and values are applicable to this Role Mapping Policy definition.

- Effect: GRANT
- Application Role: SupportManagerEast
- Principal: member of Employee group
- Condition/Constraint: IP address 229.188.21.21

Figure 2–2 illustrates how the components of this policy map to the Oracle Entitlements Server Role Mapping Policy objects.

Figure 2–2 Policy Components Mapped to Role Mapping Policy Objects



A Role Mapping Policy can also be used to prevent specific users from being assigned an Application Role. Consider the following Role Mapping Policy definition:

```
DENY the Customers group application role GoldCircle
   if the account balance is less then $10,000
```

This policy denies the `GoldCircle` Application Role to any members of the group Customers IF their account balance is less then $10,000. For information on how to create, update and delete Role Mapping Policies, see Section 3.5.7, "Managing Role Mapping Policies."

2.2 How Oracle Entitlements Server Evaluates Policies

During Oracle Entitlements Server runtime evaluation, the following occurs:

1. Based on the subject, a list of Application Roles is determined by:
   a. Retrieving the user’s static role membership.
   b. Evaluating all applicable Role Mapping Policies with a GRANT effect and adding them to the list of roles previously determined.
   c. Evaluating all applicable Role Mapping Policies with a DENY effect and removing them from the list of roles previously determined.
2. Based on the subject and list of retrieved Application Roles, a list of Authorization Policies is evaluated to find any that might be applicable based on the grantee, target matching and conditions. (The actions allowed on the resource are defined by the Authorization Policy.)
3. A final authorization decision is based on the “DENY overrides” combining algorithm and returned to the calling application.

2.3 The Policy Object Glossary

The policy objects defined in this section can be created, provisioned or managed using the Authorization Policy Manager Administration Console.

- **Policy Store**

  The policy store is where all Oracle Entitlements Server policy objects (including, but not limited to, Applications, Resources and various role types) are stored. A policy store can be a relational database (preferred) or an LDAP-based directory. For more information, see Section 1.1.2.3, "Using the Policy Store."

- **Principal**

  A Principal is the identity to which the access rights defined in a policy are granted. A principal can be a user, a group, an External Role, or an Application Role. Most frequently, it is an Application Role. For information on adding a principal to an Oracle Entitlements Server policy, see Section 3.5.5.1, "Creating an Authorization Policy" or Section 3.5.7.1, "Creating a Role Mapping Policy."

- **Application Role**

  An Application Role is a collection of users, groups, or other Application Roles defined using Oracle Entitlements Server. It can be assigned to an enterprise user, group, or external role in an identity store, or another Application Role in the Oracle Entitlements Server policy store. When creating an Application Role you might grant it all privileges necessary to access a given target Resource. Then, it can be assigned statically to a user by granting the user membership in the Application Role, or dynamically by referencing the Application Role as principal in a Role Mapping Policy. One target application may have several different roles, with each role assigned a different set of privileges for more fine-grained access.

  Many Application Roles can be mapped to one External Role. For example, the external group `employee` (stored in LDAP-based identity store) can be mapped to the Application Role `customersupport member` (defined in one application) and to the Application Role `IT member` (defined in another application). For more information, see Section 3.5.6, "Managing Application Roles in the Role Catalog."

> **Note:** Search for Application Roles in the Role Catalog node of the Oracle Entitlements Server Administration Console. See Chapter 4, "Searching for Security Objects" for more information.

- **External Role**

  An External Role is a collection of users and groups defined in an external identity store such as an LDAP server or a database. The term *external role* is often synonymous with the terms enterprise role or enterprise group; they are typically implemented as LDAP groups in the identity store. For information on adding an External Role to a policy, see Section 3.5.5.1, "Creating an Authorization Policy" or Section 3.5.7.1, "Creating a Role Mapping Policy."

  > **Note:** Within Oracle Entitlements Server, external roles and users are read-only. They are typically managed with a different tool, such as Oracle Identity Manager. For more information, see *Oracle Fusion Middleware System Administrator's Guide for Oracle Identity Manager*.

- **Authorization Policy**

  An Authorization Policy specifies a set of rights that an entity (the grantee, a principal or code source) is allowed on a protected resource. This might include viewing a web page or modifying a report. In short, it specifies who can do what to a resource protected by Oracle Entitlements Server. For more information, see Section 3.5.5, "Managing Authorization Policies."

  > **Note:** Search for Authorization Policies in the Default Policy Domain (or a custom Policy Domain, if applicable) node of a configured Application. See Chapter 4, "Searching for Security Objects" for more information.

- **Role Mapping Policy**

  A Role Mapping Policy is used to determine what external subjects (users, groups or External Roles) are assigned to the applicable Application Role. The Application Role, when referenced in an Authorization Policy, defines the principals affected by the Authorization Policy. Role Mapping Policies may also include conditions. For more information, see Section 3.5.7, "Managing Role Mapping Policies."

  > **Note:** Search for Role Mapping Policies under the Role Catalog node of the Oracle Entitlements Server Administration Console. See Chapter 4, "Searching for Security Objects" for more information.

- **Application**

  An Application is a high-level container for managing roles, policies, resource definitions, and other policy objects; in effect, all items needed to define secure access to a particular resource. An Application may correspond to a single deployed software application, a set of deployed software applications, or components of a software application (such as an Enterprise Java Bean). You can manage more than one Application with Oracle Entitlements Server. For more information, see Section 3.5.1, "Managing Applications."

- **Resource Type**

  A Resource Type represents the type of a secured object. Protected software application components that share common characteristics can be represented by particular Resource Type. For example, a set of pages can be represented by one Resource Type and bank accounts by another Resource Type. A Resource Type defines *resource attributes* and possible valid actions that are applicable to the protected component. It also defines how to match a resource passed by the software application to a Resource defined in an Authorization Policy. A Resource is instantiated from a Resource Type. For more information, see Section 3.5.2, "Managing Resource Types."

- **Resource**

  A Resource is a protected component or object to which access is granted or denied. A Resource represents the application component or business object that is secured by an Authorization Policy. At runtime, the application passes the Resource name to check for access definitions that will determine whether a Principal is authorized access. A Resource requires an associated Resource Type. For more information, see Section 3.5.3, "Managing Resources."

- **Policy Domain**

  A Policy Domain is a container under an Application object that can serve as a partition to facilitate management of Resources, Entitlements and Authorization Policies. The Policy Domain is an optional management construct that can restrict an administrator’s right to a particular subset of Resource, Entitlements, and Authorization Policies. The Policy Domain has no effect upon runtime policy evaluation. Multiple Policy Domains can be created and can be hierarchical. A *default policy domain* is added to each Application upon its creation. For more information, see Section 6.4, "Using Policy Domains to Delegate."

- **Entitlement**

  An Entitlement (also known as a *permission set*) represents a small set of Resources and the associated actions needed to perform a task. It groups related Resources, possibly of different types, needed to perform a business function. An Entitlement is a reusable collection of access rights that can be granted to multiple principals. For more information, see Section 3.5.4, "Managing Entitlements."

- **Attributes**

  An *Attribute* represents data that can be used in a policy *condition*, or returned with the policy determination as an *obligation*. It is defined by its name, the type of data it takes as a value, and whether the value is single or multiple. An attribute value can either be passed by the protected application as part of an authorization request, or retrieved by Oracle Entitlements Server using an attribute retriever. For more information, see Section 3.5.9, "Managing Attributes and Functions as Extensions."

- **Functions**

  A *Function* represents custom code that can be invoked as part of the evaluation of a policy condition; the returned value will affect the evaluation of the condition.

For more information, see Section 3.5.9, "Managing Attributes and Functions as Extensions."

- **Condition**

  A Condition is one or more constraints that must be evaluated to true in order for the policy to be included in the authorization decision. Adding a Condition to a policy is optional and when used, further restricts access to the protected resource. In general, conditions consist of boolean expressions that test the value of some user, resource, or system attribute. Individual conditions can be combined with the following logical operators: AND, OR, and NOT. Conditions can define constraints based on date, time, a time range, a day of week, and so forth. For more information, see Section 3.6, "Using the Condition Builder."

- **Obligation**

  An Obligation specifies optional information that is returned together with an authorization decision. When used, an Obligation may impose an additional requirement for the policy enforcing component, or simply contain useful information. For example, the reason a request for access has been denied might be returned as an Obligation. For information on adding an Obligation to a policy, see Section 3.5.5.1, "Creating an Authorization Policy."

- **Role Category**

  A Role Category is an optional tag that can be associated with an Application Role; it can be used for searching. Role Categories enable administrators to organize roles in arbitrary flat collections. They have no effect upon runtime policy evaluation. For more information, see Section 3.5.8, "Managing a Role Category."

2.4 Implementing a Policy Use Case

Oracle Entitlements Server provides the ability to externalize policy management and policy decision making logic from an organization’s resources. It secures access to the organization’s resources by implementing policies that specify the users, groups, and roles that can access them. Resources can be application software components (URLs, Enterprise JavaBeans, JavaServer Pages) or enterprise business objects (customer accounts, patient records). This use case considers how the policy model can be used to secure the financial services offered by Acme Investment Bank. It is based on the concept of *hierarchical resources* - resources are organized as a tree and inherit from their parent elements.

> **Note:** Oracle Entitlements Server also supports the concept of non-hierarchical (flat) resources. See Section 3.5.2, "Managing Resource Types" for more information.

In this use case, the following conditions apply:

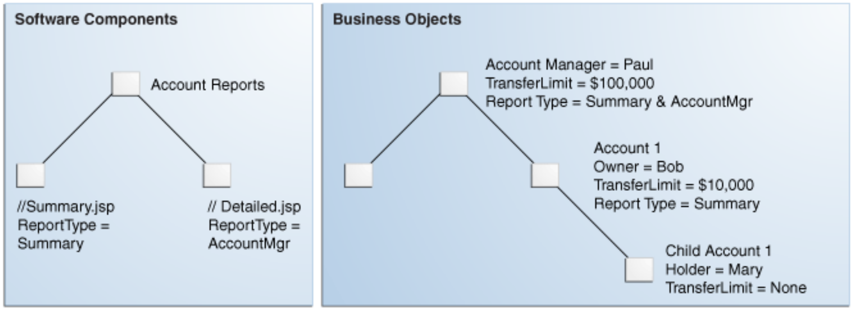
- A customer may open a family account and is considered an owner of the account.
- The customer may open a child account for family members and set transfer limits for each member. Transfer limits must be lower than the customer's own transfer limit.
- Each account has a bank employee that acts as an account manager and sets the transfer limits for the account.

- Both a Summary report and an Account Manager report are associated with each family account.

Figure 2–3 illustrates the financial services scenario by organizing the protected resources into business objects and software components. Paul is a bank employee who is an account manager and can set transfer limits up to $100,000 on the accounts that he manages. Paul manages account owner Bob's family account which has a transfer limit of $10,000. Bob manages his family account and a child account he created for Mary who may not transfer money. These accounts are considered business objects and are protected as such.

Account owner Bob has access to a Summary report generated for his family account. Account manager Paul has access to Bob's Summary report and his own Account Manager report. Child account holder Mary has access to no reports. These reports, generated as JavaServer Pages, are considered software components and are protected as such.

Figure 2–3 Use Case for Software Components and Business Objects



For any given user, a request for access to a financial services account (business object) or account report (software component) generates a decision based on the following questions:

1. Is this user an account holder, account owner or an account manager?
2. Can this user transfer funds from this account (subject to the role, transfer limit, and time of transaction)?
3. What reports can this user access?

The first two questions can be decided using policies created to protect business objects, while the last can be decided using policies created to protect software components. The following sections illustrate how to conceptualize the policies.

- Section 2.4.1, "Protecting Software Components"
- Section 2.4.2, "Protecting Business Objects"

2.4.1 Protecting Software Components

The Account Reports node in Figure 2–3, "Use Case for Software Components and Business Objects" represents the reporting application. `Summary.jsp` and `Detailed.jsp` are the software components. There are several options from which to choose when deciding how to model policies for securing these software components. One option is to set an Authorization Policy for the top node reporting application by using group membership. The following example illustrates how access is explicitly granted by naming the resource and the group of users that can access it.

```
GRANT the BankManagers group access to the AccountReports node
```

This top down Authorization Policy grants access to the Account Reports node for anyone in the BankManagers group. Because these resources are hierarchical, anyone in the allowed group has access to both the Summary and Detailed reports. But this access may be restricted using system-based or attribute-based conditions. For example, adding a condition based on time or based on the value of a specific user, group, or resource attribute would further limit access. The following example illustrates how a time-based condition restricts access of the reports to typical office hours.

```
GRANT the BankManagers group access to the AccountReports node
  IF the request is made between 09:00 and 17:00
```

Another option can set the top down Authorization Policy by defining the principal as a role rather than a group. A role is comprised of users or groups. (An LDAP role would be granted enterprise wide whereas an Application Role is specific to the Application for which it was configured.)

> **Note:** A user can be assigned to a role through direct membership or a Role Mapping Policy.

In the following example, the resource is explicitly named and access is implicitly granted to a user if the user is assigned the defined role.

```
GRANT access to AccountReports node if user has BankManagers role
```

You can also dynamically assign the BankManagers role to users accessing the reporting application if they are a member of the BankManagers group (as illustrated below).

```
GRANT BankManagers role to members of BankManagers group
  FOR access to AccountReports node
```

Another way to define the previous Authorization Policy is to assign the role based on a user attribute value rather than group membership. (In a large enterprise, it is typically more efficient to assign users based on attributes than on group membership.) The following example assigns the BankManagers role to the requesting user if the value of the UserType attribute in the user’s profile is BankManager.

```
GRANT BankManagers role to anyone defined by UserType 'BankManager'
  FOR access to the AccountReports node
```

The previous examples represent Authorization Policies that scope from the top node reporting application down to the reports but Authorization Policies can also be defined for the specific report nodes. The following example grants access to the `Summary.jsp` report to all assignees of the BankManagers and AccountOwners roles. The additional condition is that the principal requesting access must be listed on the account to which the report pertains.

```
GRANT Summary.jsp access to all members of BankManagers and AccountOwners role
  IF the requesting assignee is listed as OWNER or MANAGER on specified account
```

Another example illustrates how access to the `Detailed.jsp` report can be granted to anyone who is assigned the BankManager role.

```
GRANT Detailed.jsp access to all assignees of BankManagers role
```

The previous examples show how an Authorization Policy can be modeled for specific application software components. In a real enterprise scenario, each application may have tens or hundreds of resources so it might not be practical to write an Authorization Policy for each one. The concept of *resource attributes* has been implemented by Oracle Entitlements Server to address this proliferation of application software component resources and associated Authorization Policies.

By associating a resource with an attribute, you can grant access based on the value of the attribute. For example, *filetype* could be a resource attribute that is used to define an HTML page, an image, or a PDF. By defining a condition as `if filetype=pdf`, you can grant access to all PDF files that are associated with the resource. The following example uses a resource attribute; it allows users assigned the BankManagers and AccountOwners role access to all reports although access is granted only if the report type being requested matches the value of the UserReportType attribute in the specific user's profile.

```
GRANT users assigned BankManagers or AccountOwners roles
     access to AccountReports
   IF requested ReportType matches UserReportType attribute value
     in user profile
```

This policy grants BankManagers and AccountOwners access to all reports although access is constrained based on matching resource attribute values with user attribute values. An advantage of this approach is that the policy governing access need not change as resources are added to, or removed from, an application. As resources change, the ReportType resource attribute attached to the application continues to govern access.

2.4.2 Protecting Business Objects

There are several options from which to choose when deciding how to model Authorization Policies for securing business objects. In this banking scenario, business objects are bank accounts. Figure 2–3, "Use Case for Software Components and Business Objects" illustrates the Acme bank account structure.

Each bank account can have a manager, an owner, and a holder with each *scope* assigned a certain set of privileges (or *entitlements*). The policy evaluating what a user can do on the bank account is then based on the user's attributes rather than the resource. The following example allows anyone to transfer money but that privilege is only granted if the user is defined as owner of the account requested and the amount of money being transferred is less than or equal to the limit defined for the user.

```
GRANT anyone transfer privileges only
   IF the user is listed as OWNER on specified account
  AND transfer amount is equal to or less than the transfer limit
```

There is another option to acquire a user's entitlements. Rather than comparing a transfer request to a transfer limit, Oracle Entitlements Server can return the transfer limit amount as the output of evaluation. In this scenario, the user's ability to access the account is verified but the transfer amount is returned to the caller (in a Java object) as an *obligation*. This leaves verification that the transfer amount is within the transfer limit up to the application. The following example illustrates this model.

```
GRANT anyone transfer privileges only
   IF the user is listed as OWNER on specified account
  THEN RETURN transfer limit to calling application
```

A model where the bank account corresponds to an individual resource instance can also be used; however, this would yield a proliferation of policies (one for each account) and become unmanageable. For example, if Acme Investment Bank had 100,000 accounts, it would need at least 100,000 policies just to manage transfers. For more information on adding obligations, see Section 3.5.5, "Managing Authorization Policies."

3

Managing Policies and Policy Objects

The Oracle Entitlements Server Administration Console (Authorization Policy Manager) is used to manage Authorization Policies, Role Mapping Policies, and the security objects from which they are created. This chapter contains the following sections:

- Introducing Policy and Policy Object Management
- Defining an Authorization Policy And Its Components
- Adding Fine-Grained Elements to an Authorization Policy
- Implementing An Authorization Policy Step by Step
- Managing Policy Objects in An Application
- Using the Condition Builder

3.1 Introducing Policy and Policy Object Management

Oracle Entitlements Server allows administrators to perform create, read, update, and delete (CRUD) operations on all policies and global objects. Tasks performed in the Administration Console typically require that an administrator identify an object (by browsing or searching), select it, and choose one of the operations available for it. The following sections contain information relevant to managing policies and policy objects with Authorization Policy Manager.

- Section 3.1.1, "Organizing Policy Objects"
- Section 3.1.2, "Using Application Roles"
- Section 3.1.3, "Mapping Oracle Fusion Applications and Authorization Policy Manager Terms"

3.1.1 Organizing Policy Objects

Policy objects are organized into nodes that are displayed in the Authorization Policy Manager Navigation Panel. The nodes are Applications and Global.

- Application objects include the objects used to create Authorization Policies (Resources, Application Roles and the like). They apply to, and can only be used for policies within, the Application under which they are defined. The Applications node in the Navigation Panel is the branch under which all configured Applications (and their respective objects) are organized. This chapter contains information on managing Applications and their objects.
- Global objects include Users, External Roles, and system configurations for Attribute Retrievers, administrators and the like. These objects may apply to all

configured Applications throughout the system. The Global node in the Navigation Panel is the branch under which all systemwide objects are organized. These objects are discussed in Chapter 12, "Managing System Configurations."

> **Note:** Within Oracle Entitlements Server, External Roles (and Users) are read only; they are typically managed with a different tool, such as Oracle Identity Manager. For more information, see *Oracle Fusion Middleware System Administrator's Guide for Oracle Identity Manager*.

3.1.2 Using Application Roles

Oracle Entitlements Server supports the mapping of policies to individual users, External Roles, and Application Roles. However, mapping policies to Application Roles is recommended because:

- Managing authorization based on grants to individual Users and External Roles can quickly become unmanageable as the number increases.
- If the identity management source changes (for example, when a move between development, test and production environments results in a new LDAP server), no changes to policy definitions are needed. All that is required is a re-mapping of the Application Roles to the Users and External Roles already available in the target environment.

3.1.3 Mapping Oracle Fusion Applications and Authorization Policy Manager Terms

Authorization Policy Manager and Oracle Fusion Applications use different terms to signify role types. The terms Application Role and Enterprise Role are used in Authorization Policy Manager and the terms duty role, job role, data role, and abstract role are used in Oracle Fusion Applications. Table 3–1 lists the mappings for equivalent terms used in the implementations.

Table 3–1 Terminology Mapping Table

Oracle Fusion Applications Term	Authorization Policy Manager Term
Data Role	Enterprise role used only in data security policies. Typically, the name of a data role has the suffix _DATA.
Job	Enterprise role mapped to application role. Typically, the name of this role has the suffix _JOB.
Abstract Role	Enterprise role, which are persisted as LDAP groups and can be managed with Oracle Authorization Policy Manager and Oracle Identity Management.
Duty	Application Role (used only in Application object) The Oracle Fusion Applications security reference implementation defines a large number of duty roles that correspond to the responsibilities of individual job roles. Duty roles are specific to applications, stored in the policy store, and shared within an Oracle Fusion Applications instance.
Privilege	Entitlement (or Permission Set)
FND Grant (or Foundation Grant)	Data security policy, which ties a data role or job role to a specific set of data.

Each of the Oracle Fusion Applications roles is implemented in Authorization Policy Manager as follows.

- Internal roles are roles that are not assigned directly to end users. They are also called Application Roles because they are specific to an application.
- External roles are roles associated with a collection of end users and other groups. They are also called enterprise roles because they are shared across the enterprise. In Oracle Fusion Applications, enterprise roles include job roles, data roles, and abstract roles.

3.2 Defining an Authorization Policy And Its Components

Defining a policy requires that the objects be created in a particular order. For example, a Resource can only be created after defining a Resource Type. A policy can be composed by following the sequence described below.

1. Create an Application.

   In the Navigation Panel, an Application should be created as the overall container for policies and related information that secure the components of a particular resource. You may create as many Applications as needed although it is recommended that only one is created for each application to be secured. For more information, see Section 3.5.1, "Managing Applications."

2. Create a Resource Type.

   A Resource Type specifies one or more resource attributes, and definitions of all possible valid actions that can be performed on a particular kind of resource. The actions can be standard actions (GET and POST to a URL) or custom actions on a business object (transfer to or from a bank account). Consider the following Resource Types and their valid actions:

   - A text file may support Read, Write, Copy, Edit, and Delete.
   - A checking account application may support deposit, withdrawal, view account balance, view account history, transfer to savings, and transfer from savings.

   Resource instances are created from Resource Types. Actions defined by the Resource Type are granted or denied when accessing a protected Resource instance created from the Resource Type.

   > **Note:** A Resource instance is defined in a Policy Domain and references the Resource Type. For more information, see Section 3.5.3, "Managing Resources."

   For more information, see Section 3.5.2, "Managing Resource Types."

3. Instantiate a Resource from the Resource Type.

   A specific protected target (Resource) is instantiated from a Resource Type. A Resource represents a secured target (for example, an application) and is created under a Policy Domain in the Resource Catalog. If no Policy Domain is specified, it is created under the Default Policy Domain. For more information, see Section 3.5.3, "Managing Resources."

> **Note:** A Policy Domain is an optional object that is created for purposes of delegated administration and organization. See Chapter 6, "Delegating With Administrator Roles."

4. Build the Authorization Policy.

   This entails specifying the effect (GRANT or DENY), adding a user, group or role as the policy principal and the Resource and actions as the policy target. Optionally, you can add an Obligation or build a Condition. For more information, see Section 3.5.5, "Managing Authorization Policies."

3.3 Adding Fine-Grained Elements to an Authorization Policy

Section 3.2, "Defining an Authorization Policy And Its Components" documented the minimum components needed to create an authorization policy. The following fine-grained elements can be added to a simple policy.

- Entitlements

  An Entitlement associates an instantiated Resource with the applicable actions that can be performed on it. The set of actions for a Resource are a subset of the set of legal actions already defined in its corresponding Resource Type. For more information, see Section 3.5.4, "Managing Entitlements."

- Application Roles

  An Application Role can be assigned statically or dynamically to an enterprise user, group, or external role in an identity store, or another Application Role in the policy store. One target application may have several different Application Roles, with each role assigned a different set of privileges for more fine-grained access. For more information, see Section 3.5.6, "Managing Application Roles in the Role Catalog."

- Role Mapping Policy

  Membership to an Application Role can be granted dynamically with a Role Mapping Policy. An Application Role, referenced as a Principal in a Role Mapping Policy, could grant a user access to the defined resources but the Role Mapping Policy must be resolved before an authorization decision is reached. The resolution answers the question *Can the user requesting access be assigned this Application Role?* During runtime evaluation of a Role Mapping Policy, the following occurs:

  1. Based on the subject, a list of application roles is determined by retrieving static role membership and evaluating any applicable role mapping policies.
  2. Based on the subject and list of application roles, a list of Authorization Policies is evaluated to find any that might be applicable based on the grantee, target matching and constraints evaluation. The actions allowed on the Resource are defined by the Authorization Policy.
  3. Final authorization decision is based on the “DENY overrides” combining algorithm.

  For more information, see Section 3.5.7, "Managing Role Mapping Policies."

- A Condition can be added to a policy as a way of setting an additional contingency on the policy. It is applicable to either an Authorization Policy or a

Role Mapping Policy. A Condition is written in the form of an expression that resolves to true or false and has one of the following outcomes:

- If the expression resolves to true, the policy condition is satisfied and the effect defined in the PolicyRuleEntry is applicable.
- If the expression does not resolve to true, the policy is not applicable.

A Condition must be true for the policy to evaluate to true. Conditions can be complex combinations of boolean expressions that test the value of some user, resource, or system attribute or they can be custom Java evaluation functions that evaluate complex business logic. For more information, see Section 3.6, "Using the Condition Builder."

- An Obligation specifies optional information to be evaluated during the policy enforcement phase of authorization. The obligation is returned with the corresponding policy effect (GRANT or DENY). This information may or may not be taken into account during policy enforcement based on settings defined by the application. For example, the reason a request for access has been denied might be returned as an obligation. A different type of obligation might involve sending a message; for example, if a certain amount of money is withdrawn from a checking account, send a text message to the account holder's registered mobile phone. For more information, see Section 3.5.5, "Managing Authorization Policies."

3.4 Implementing An Authorization Policy Step by Step

In Section 2.4, "Implementing a Policy Use Case," several use cases for creating a policy are discussed. This section documents the step by step procedure to create an Authorization Policy (and the policy objects from which it is comprised) using the Administration Console. This procedure assumes you have installed Oracle Entitlements Server and a Java Security Module to protect an application.

1. Create an Application.

   The Application Name must match what is used in the application code. For example, create a `HelloOESworld` Application object to map to a `HelloOESworld` Application. See Section 3.5.1.1, "Creating an Application."

2. Create a Resource Type.

   The Resource Type Name must match what is used in the application code. For example, create a `Files` Resource Type object for use in collecting files that will be protected. Associate the *write* and *read* actions with the Resource Type. See Section 3.5.2.1, "Creating a Resource Type."

3. Create a Resource.

   A Resource Name must match what is used in the application code. Additionally, the Resource is created from the Resource Type. For example, create a `FinanceFile` Resource from the `Files` Resource Type. See Section 3.5.3.1, "Creating a Resource."

4. Create the Authorization Policy.

   In the `HelloOESworld` Application, create an Authorization Policy. Add one or more Principals (Roles or Users), one or more targets (Resources or Entitlements) and confirm the actions for the target. Optional conditions or obligations can also be added before saving. See Section 3.5.5.1, "Creating an Authorization Policy."

5. Create a Security Module definition and bind it to the Application.

This step defines the Security Module to which this Authorization Policy is distributed once binded. See Section 12.2, "Configuring Security Module Definitions."

6. Distribute the Authorization Policy to the Security Module.

   See Chapter 9, "Managing Policy Distribution."

3.5 Managing Policy Objects in An Application

The following sections describe how to manage policy objects specific to the Applications.

- Section 3.5.1, "Managing Applications"
- Section 3.5.2, "Managing Resource Types"
- Section 3.5.3, "Managing Resources"
- Section 3.5.4, "Managing Entitlements"
- Section 3.5.5, "Managing Authorization Policies"
- Section 3.5.6, "Managing Application Roles in the Role Catalog"
- Section 3.5.7, "Managing Role Mapping Policies"
- Section 3.5.8, "Managing a Role Category"
- Section 3.5.9, "Managing Attributes and Functions as Extensions"

3.5.1 Managing Applications

An Application is created as the overall container for policies and related artifacts that secure the components of a particular application. These artifacts include (but are not limited to) roles, resources, attributes and functions. You may create as many Application instances as needed although it is recommended that only one is created for each application to be secured. The following sections describe management operations on Application instances.

- Section 3.5.1.1, "Creating an Application"
- Section 3.5.1.2, "Modifying an Application"
- Section 3.5.1.3, "Deleting an Application"

3.5.1.1 Creating an Application

To create an Application, proceed as follows:

1. Right-click Applications in the Navigation Panel and select New from the menu.

   > **Note:** Alternately, click Create Application under Search and Create in the Home area.

   An Untitled page with several tabs displays in the Home area. The General tab is active. You can only configure the Delegated Administrators and Policy Distribution details after the Application has been created. See Section 3.5.1.2, "Modifying an Application" for information.

2. Provide the following information for the application being created under the General tab.

- **Display Name:** The Display Name is optional and case insensitive. Specifying a meaningful value, though, is recommended as it is displayed in the Administration Console and can be used as a search parameter.
- **Name**: The name is required and case insensitive. It must match what is used in the application code.
- **Description**: Although optional, it is recommended to provide useful information about the Application.

3. Select one of the following from the Save menu.
   - Save and Close saves the configuration, renames the tab with the value provided for the Application's Display Name and activates the Delegated Administrators and Policy Distribution tabs.
   - Save and Create Another saves the configuration to the information tree in the Navigation Panel but leaves the Untitled area open for you to create another Application.

3.5.1.2 Modifying an Application

To modify an Application (including the configuration of Delegated Administrators and Policy Distribution details), proceed as follows:

1. Expand the Applications node in the Navigation Panel.
2. Select the name of the Application to modify.
3. Right-click the Application name and select Open from the menu.

   Alternately, double-click the Application name. The Application page is displayed and the General tab, the Delegated Administrators tab and the Policy Distribution tab are all active.
4. Select the tab you want to modify or configure and see the appropriate section for parameter details.
   - **General**: Section 3.5.1.1, "Creating an Application"
   - **Delegated Administrators:** Chapter 6, "Delegating With Administrator Roles"
   - **Policy Distribution**: Chapter 9, "Managing Policy Distribution"
5. Apply or save as necessary.

3.5.1.3 Deleting an Application

To delete an Application instance, proceed as follows:

1. Find the Application to delete using an advanced search (as documented in Section 4.3.2, "Searching Applications").

   The Search Applications page is displayed.
2. Enter query parameters and click Search.

   The search results are displayed.
3. Select the Application name from the results and click Delete.

   A Delete Warning is displayed.
4. Click Delete.

   The Application is deleted.

> **Note:** Alternately, expand the Applications information tree in the Navigation Panel and double click the name of the Application to delete. The Application is displayed in the Home area. Click Delete in the upper right corner.

3.5.2 Managing Resource Types

Resource Types specify the full scope of traits for a particular kind of protected resource. It contains one or more resource attributes, and definitions of all possible valid actions that can be performed on the particular kind of resource. An *action* represents an activity or task in your business process that can be executed on a resource. Actions can be standard (GET and POST to a URL) or custom on a specific business object (transfer to or from a bank account). A Resource instance for a specific target is created from a Resource Type. The following sections describe management operations on Resource Types.

- Section 3.5.2.1, "Creating a Resource Type"
- Section 3.5.2.2, "Modifying a Resource Type"
- Section 3.5.2.3, "Deleting a Resource Type"

3.5.2.1 Creating a Resource Type

To create a Resource Type, proceed as follows:

1. Choose form the following methods to display the page for creating a Resource Type.
   - Expand the information tree in the Navigation Panel, right-click Resource Types under the particular Application in which the Resource Type will be created and select New from the menu.
   - In the Home area, select the Application Name under which the Resource Type will be created and click New under Resource Types.

   An Untitled page is displayed in the Home area.
2. Provide the following information for the Resource Type.
   - **Display Name**: The display name is optional and case insensitive. Specifying a meaningful value, though, is recommended as it is displayed in the Administration Console and can be used as a search parameter.
   - **Name:** The name is required and case insensitive.
   - **Resource Finder** : An (optional) class that implements the `oracle.security.jps.service.policystore.entitymanager.ResourceFinder` interface. It allows resources managed outside of the Policy Store to be consumed. *(Reserved for future use.)*
   - **Description** : Although optional, it is recommended to provide useful information. The description string is case insensitive.
3. Add actions allowed by the Resource Type in the Actions section.
   a. Click New to display the New Action dialog
   b. Enter the name of the action.

      The string entered must match the actions for which your application is asking for authorization. If a Permission class is added, the action must be meaningful to it.

c. Click Save.

The Action list is updated with the new action.

4. Choose one of the following methods to add attributes to the Resource Type.

- Drag and drop

  a. Use the Navigation Panel to list the Application's available attributes by performing a simple search on configured Resource instances. For more information, see Section 4.2, "Finding Objects with a Simple Search."

  b. Drag and drop attributes from the Search Results tab into the area labeled Attributes.

- Find Existing Attribute dialog

  a. In the Attributes section, click Add to display the Find Existing Attribute dialog.

  b. Select the attribute Type from the list.

  c. Enter an (optional) string to match in the Search text box.

  d. Click the arrow icon next to the Search text box to begin the search.

  e. Select the attributes to add and click Add.

  Use Ctrl + click to select multiple items from the list.

These attributes are used when instantiating a Resource. For more information, see Section 3.5.3.1, "Creating a Resource."

5. Configure the remaining fields.

The selection changes according to the Resource Type being created.

- Supports Resource Hiearchy - Select Yes or No to set the Resource Type as hierarchical. This means the following when the Resource Type is used to instantiate a Resource:

  – A policy applicable to a Resource created from a hierarchical Resource Type is also applicable to Resources that are its children.

  – Any attribute defined for a Resource created from a hierarchical Resource Type is inherited by Resources that are its children.

- Resource Name Delimiter - Only valid when Supports Resource Hierarchy is enabled. The default delimiter is `Slash (/).`

- Evaluation Logic - Evaluation logic for a Resource Type can be either a permission class or a default matching algorithm. Define the algorithm here or the permission class below.

- Permission Class - When the evaluation logic is a Permission class, a class name is required and is case sensitive.

- Action Name Delimiter - The specified character is used to separate actions in a list when the Resource Type represents a permission.

- All Action Keyword - If the policy's target contains the defined keyword as an action, the policy will match any action passed in with the authorization request. For example, assume that this parameter is set to `ANY` and you create the following policy:

```
GRANT user "Michael" action:"ANY" on resource:"Resource1
```

The decision for authorization requests like *Can Michael do 'write' on Resource1?* or *Can Michael do 'transfer' on Resource1?* will return ALLOW. The use of this parameter allows you to create a single Authorization Policy that would be applicable to any valid action for that Resource Type.

6. Select one of the following to save the Resource Type.
   - Save and Close saves the configuration and renames the tab with the value provided for the Display Name.
   - Save and Create Another saves the configuration to the information tree in the Navigation Panel but leaves the Untitled area open for you to create another Resource Type.

3.5.2.2 Modifying a Resource Type

To modify a Resource Type, proceed as follows:

1. Choose from the following methods to display the desired Resource Type.
   - Expand the information tree in the Navigation Panel to find the Resource Types node under the appropriate Application and double click it. A search dialog opens in the Home area.
   - In the Home area, select the Application Name under which the Resource Type was created and click Search under Resource Types. A search dialog opens in the Home area.

   For information about searching, see Section 4.3.3, "Searching Resource Types."
2. Enter query parameters and click Search.

   The search results are displayed.
3. Select the appropriate Resource Type name from the search results and click Open to display the details.

   Alternately, search for Resource Types using the Navigation Panel's search function and double-click the Resource Type name in the Search Results tab to display the details.
4. Modify as necessary.
5. Click Apply.

3.5.2.3 Deleting a Resource Type

To delete a Resource Type, proceed as follows:

1. Choose from the following methods to display the desired Resource Type.
   - Expand the information tree in the Navigation Panel to find the Resource Types node under the appropriate Application and double click it. A search dialog opens in the Home area.
   - Select the appropriate Application Name in the Home area and click Search under Resource Types. A search dialog opens in the Home area.

   For information about searching, see Section 4.3.3, "Searching Resource Types."
2. Enter query parameters and click Search.

   The search results are displayed.
3. Select the appropriate Resource Type from the search results to display the details.

Alternately, search for Resource Types using the Navigation Panel's search function and double-click the Resource Type name in the Search Results tab to display the details.

4. Click Delete.

   A Delete Warning is displayed.

5. Click Delete.

   The Resource Type is deleted.

3.5.3 Managing Resources

A Resource represents a specific, secured target in a protected application. Each Resource belongs to a defined Resource Type and can represent software components managed by a container (URLs, EJBs, JSPs) or business objects in an application (reports, transactions, revenue charts).

> **Note:** Resources can be hierarchical (in that the child resource inherits attributes from parent resources) or non-hierarchical. When organized in a hierarchy (root down), you can add new attributes to the parent resources or overwrite any existing attributes that are inherited.

The following sections describe management operations on Resources.

- Section 3.5.3.1, "Creating a Resource"
- Section 3.5.3.2, "Modifying a Resource"
- Section 3.5.3.3, "Deleting a Resource"

3.5.3.1 Creating a Resource

To create a Resource, proceed as follows

1. Choose from the following methods to display the page for creating a Resource.

   - Navigate to the Resource Catalog by expanding the applicable Policy Domain node in the appropriate Application node using the Navigation Panel. Right-click Resources from the Resource Catalog node and select New from the menu.
   - Select the Application under which you will create the Resource instance from the Home area and click New under Resources.

   > **Note:** This option creates the Resource in the Application's Default Policy Domain.

   An Untitled page is displayed in the Home area.

2. Provide the following information.

   - **Resource Type**: Select from the list. This defines what is displayed in the Instance Attributes and Overwrites table.
   - **Display Name** : The display name is optional and case insensitive. Specifying a meaningful display name is recommended since it is displayed in the

Administration Console, and provides extra information to help administrators identify objects.

- **Name :** The name is required and case insensitive. At runtime, this is the string the application passes to determine whether a user is authorized to access this Resource.
- **Description** : Although optional, it is recommended to provide useful information about the entitlement. The description string is case insensitive.

3. Add or remove the attributes for this Resource from those displayed in the Instance Attributes and Overwrites dialog.

   The Overwrites dialog is displayed only in the case of hierarchical Resources.

4. Select the attributes from the list (use Ctrl + Click to select multiple items from the list) and click Add.
5. Select one of the following from the Save menu.
   - Save and Close saves the configuration, renames the tab with the value provided for the Application's Display Name and activates the Delegated Administrators and Policy Distribution tabs.
   - Save and Create Another saves the configuration to the information tree in the Navigation Panel but leaves the Untitled area open for you to create another Application.

3.5.3.2 Modifying a Resource

To modify a resource, proceed as follows:

1. Choose from the following methods to display the desired Resource.
   - Navigate to the Resource Catalog by expanding the applicable Policy Domain node in the appropriate Application node using the Navigation Panel and double click it. A search dialog opens in the Home area.
   - In the Home area, select the Application Name under which the Resource Type was created and click Search under Resources. A search dialog opens in the Home area. (This search queries only in the Default Policy Domain.)

   For information about searching, see Section 4.3.6, "Searching Resources."

2. Enter query parameters and click Search.

   The search results are displayed.

3. Select the appropriate Resource name and click Open to display the details.

   Alternately, search for Resources using the Navigation Panel's search function and double-click the Resource name in the Search Results tab to display the details.

4. Modify the Resource as necessary.
5. Click Apply.

3.5.3.3 Deleting a Resource

To delete a Resource, proceed as follows:

1. Choose from the following methods to find the desired Resource.
   - Navigate to the Resource Catalog by expanding the applicable Policy Domain node in the appropriate Application node using the Navigation Panel and double click it. A search dialog opens in the Home area.

- In the Home area, select the Application Name under which the Resource was created and click **Search** under **Resources**. A search dialog opens in the Home area. (This search queries only in the Default Policy Domain.)

For information about searching, see Section 4.3.6, "Searching Resources."

2. Enter query parameters and click Search.

   The search results are displayed.

3. Select the appropriate Resource from the search results to display the details.

   Alternately, search for Resources using the Navigation Panel's search function and double-click the Resource name in the Search Results tab to display the details.

4. Click Delete.

   A Delete Warning is displayed.

5. Click Delete.

   The Resource is deleted.

3.5.4 Managing Entitlements

After instantiating a Resource, define the actions that can be performed on it in an Entitlement. The actions are defined using the set of legal actions defined in the Resource's parent Resource Type. The following sections describe management operations on Entitlements.

- Section 3.5.4.1, "Creating an Entitlement"
- Section 3.5.4.2, "Modifying an Entitlement"
- Section 3.5.4.3, "Deleting an Entitlement"

> **Note:** An Entitlement may be created if there are plans to use the same list of Resource and Action pairs in multiple policies. Otherwise, the Resource and Action pair itself can be directly specified as a target when you create an Authorization Policy. See Section 3.5.5, "Managing Authorization Policies" for more information.

3.5.4.1 Creating an Entitlement

To create an Entitlement, proceed as follows.

1. Display the page for creating an Entitlement by choosing from the following methods:

   - Navigate to the Resource Catalog by expanding the applicable Policy Domain node in the appropriate Application node using the Navigation Panel. Right-click Entitlements from the Resource Catalog node and select New from the menu.
   - In the Home area, select the Application Name under which the Entitlement will be created and click New from Entitlements.

   An Untitled page is displayed in the Home area.

2. Provide the following information.

   - **Display Name** : The display name is optional and case insensitive. Specifying a meaningful display name is recommended since it is displayed in the

Administration Console, and provides extra information to help administrators identify objects.

- **Entitlement Name :** The name is required and case insensitive. At runtime, this is the string the application passes to determine whether a user is authorized to access this Resource.
- **Description** : Although optional, it is recommended to provide useful information about the entitlement. The description string is case insensitive.

3. Choose one of the following methods to add Resources to the Entitlement.

   - Drag and drop

     a. Use the Navigation Panel to list the Application's available Resources by performing a search on Resource instances. The Resources must be searched from the same Policy Domain in which the Entitlement is being created. For more information, see Section 4.2, "Finding Objects with a Simple Search."

     b. Drag and drop Resources from the Search Results tab into the area labeled Resources.

   - Add Targets pop up search

     a. Click Add in the Targets section.

        The Add Targets dialog displays. This will search in the current Policy Domain.

     b. Search for available targets by entering a string.

        The resources matching the query are displayed in Search Results. If no search string was entered, a list of all objects of the specified type is returned.

     c. Select your choice(s) and click Add Selected.

        The Target(s) are added to the Selected Targets. Use Ctrl + click to select multiple items from the list.

> **Note:** Alternately, you can click the Resource Expression link under the Resources tab, select a Resource Type, enter a string expression and click Add to Targets. This will search for targets, using the defined criteria, dynamically at runtime. All Resources that belong to the selected Resource Type that contain the string expression are returned, within the context of the administrator privileges.

     d. Click Add Targets.

4. Add actions to the Resources as follows:

   a. Select an added resource from the Resources list to display the resource details in the Resource Details section.

   b. Expand the selected row to see the range of actions.

      Only the actions allowed for the type of the selected resource are available in this area.

   c. Check the desired actions for the Resource in the Actions section.

   d. Repeat this procedure for each Resource you have added to the Entitlement being created.

5. Select one of the following from the Save menu.

   - Save and Close saves the configuration, renames the tab with the value provided for the Application's Display Name and activates the Delegated Administrators and Policy Distribution tabs.
   - Save and Create Another saves the configuration to the information tree in the Navigation Panel but leaves the Untitled area open for you to create another Application.

3.5.4.2 Modifying an Entitlement

To modify an Entitlement, proceed as follows:

1. Choose from the following methods to display the desired Entitlement.

   - Navigate to the Resource Catalog by expanding the applicable Policy Domain node in the appropriate Application node using the Navigation Panel and double click it. A search dialog opens in the Home area.
   - In the Home area, select the Application Name under which the Entitlement was created and click Search under Entitlements. A search dialog opens in the Home area.

   For information about searching, see Section 4.3.7, "Searching Entitlements."

2. Enter query parameters and click Search.

   The search results are displayed.

3. Select the appropriate Entitlement name and click Open to display the details.

   Alternately, search for Entitlements using the Navigation Panel's search function and double-click the Resource name in the Search Results tab to display the details.

4. Modify the Entitlement details as necessary.

5. Click Apply.

3.5.4.3 Deleting an Entitlement

To delete an Entitlement, proceed as follows:

1. Choose from the following methods to delete the desired Entitlement.

   - Expand the information tree in the Navigation Panel to find the Entitlement node under the appropriate Application's Resource Catalog and double click it. A search dialog opens in the Home area. Enter criteria for the lookup and click Search. Select the appropriate Resource from the search results and click Delete.
   - Search for Entitlements using the Navigation Panel's search function and double-click the Entitlement name in the Search Results tab. The Entitlement details are displayed in the Home area. Click Delete.
   - In the Home area, select the Application Name under which the Entitlement was created and click **Search** under **Entitlements**. A search dialog opens in the Home area. Enter criteria for the lookup and click Search. Select the appropriate Resource from the search results and click Delete.

   A Delete Warning is displayed.

2. Click Delete.

   The Entitlement is deleted.

3.5.5 Managing Authorization Policies

The Authorization Policy is the mechanism that defines the access rights. A user, an Application Role or an External Role is *granted* or denied the rights of the policy. An Authorization Policy must have:

- At least one principal which can be a user, External Role or Application Role. Code sources are not allowed as a principal.
- At least one target that can be a Resource and Action association (created within the policy) or an Entitlement (created outside the policy and added to it) but not both.
- A defined effect of PERMIT or DENY.

> **Note:** Entitlement-based policies correspond closely with business functions. They are recommended in cases in which a business function considers securing a collection of resources; an entitlement can be used in one or more grants.

The following sections describe management operations on Authorization Policies.

- Section 3.5.5.1, "Creating an Authorization Policy"
- Section 3.5.5.2, "Modifying an Authorization Policy"
- Section 3.5.5.3, "Deleting an Authorization Policy"

3.5.5.1 Creating an Authorization Policy

To create a policy, proceed as follows:

1. Display the page for creating a policy by choosing one of the following methods:

   - Navigate to the Policy Domain under the appropriate Application node in the Navigation Panel and expand it. Right-click Authorization Policies from the Resource Catalog node and select New from the menu.
   - In the Home area, select the Application Name under which the Authorization Policy will be created and click New from Authorization Policies. (When using this option, the policy will be created in the Default Policy Domain.)

   An Untitled page is displayed in the Home area.

2. Provide the following information.

   - **Effect**: Select Permit if the policy will grant rights or Deny if the policy will deny rights.
   - **Display Name** : The display name is optional and case insensitive. Specifying a meaningful display name is recommended since it is displayed in the Administration Console, and provides extra information to help administrators identify objects.
   - **Name :** The name is required and case insensitive. At runtime, this is the string the application passes to determine whether a user is authorized to access this Resource.

- **Description** : Although optional, it is recommended to provide useful information about the entitlement. The description string is case insensitive.

3. Choose one of the following methods to add Principals to the Authorization Policy.

   - Drag and drop

     a. Use the Navigation Panel to list the Application's available Principals by performing a search on Users, External Roles or Application Roles. For more information, see Section 4.2, "Finding Objects with a Simple Search."

     b. Drag and drop Principals from the Search Results tab into the area labeled Principals.

     c. Select Any or All.

        If Any, the user must match at least one of the specified principals. For example, if the principals are roles, the user must be a member of at least one of the roles for the Authorization Policy to apply. If All, the user must match all of the specified principals. For example, if the principals are roles, the user must be a member of all of them for the Authorization Policy to apply.

   - Add Principals pop-up search

     For details on how to use the pop-up search box, see Section 4.1, "Searching with the Administration Console."

     a. Click Add in the Principals section.

        The Add Principals dialog displays.

     b. Select the appropriate tab to search for available Principals.

        Options are Application Roles, External Roles and Users. You can navigate between tabs and add as many selected Principal types as desired.

     c. Search for the available Principals by entering a string.

        The Principals matching the query are displayed in Search Results.

     d. Select your choice(s) and click Add Selected.

        The Principal(s) are added to the Selected Principals. Use Ctrl+click to select multiple items from the list.

     e. Click Add Principals.

     f. Select Any or All.

        If Any, the user must match at least one of the specified principals. For example, if the principals are roles, the user must be a member of at least one of the roles for the Authorization Policy to apply. If All, the user must match all of the specified principals. For example, if the principals are roles, the user must be a member of all of them for the Authorization Policy to apply.

4. Choose one of the following methods to add Targets to the Authorization Policy.

   This step adds either Resource and action associations or Entitlements or both to the Authorization Policy.

   - Drag and drop

     a. Use the Navigation Panel to list the Application's available Resources or Entitlements by performing a search. (Be sure to look for these objects in the same Policy Domain to which you are adding the Authorization Policy.) For more information, see Section 4.2, "Finding Objects with a Simple Search."

b. Drag and drop one or more Resources or Entitlements from the Search Results tab into the area labeled Targets. Expanding the added object in Targets allows you to associate an action with it.

- Add Targets pop up search

  For details on how to use the pop-up search box, see Section 4.1, "Searching with the Administration Console."

  a. Click Add in the Targets section.

     The Add Targets dialog displays.

  b. Select the appropriate tab to search for available Targets.

     Options are Entitlements and Resources. You can navigate between tabs and add as many selected Targets as desired.

  c. Search for available targets under the Entitlements tab by entering a string.

     The resources matching the query are displayed in Search Results. If no search string was entered, a list of all objects of the specified type is returned.

  d. Select your choice(s) and click Add Selected.

     The Target(s) are added to the Selected Targets. Use Ctrl+click to select multiple items from the list.

  e. Search for available targets under the Resources tab by entering a string.

     The resources matching the query are displayed in Search Results. If no search string is entered, a list of all objects of the specified type is returned.

     Alternately, you can click the Resource Expression link under the Resources tab, select a Resource Type, enter a string expression and click Add to Targets. This will search for targets, using the defined criteria, dynamically at runtime. All Resources that belong to the selected Resource Type that contain the string expression are returned, within the context of the administrator privileges.

  f. Click Add Targets.

5. Select the Conditions tab to add a condition.

   For more information, see Section 3.6, "Using the Condition Builder."

6. Select the Obligations tab.

   An Authorization Policy may have zero, one or more Obligations.

   a. Click New to display the New Obligation dialog.

   b. Provide a Name and an (optional) Display Name and Description for the New Obligation and click Add.

   c. Click New in the Attributes section to add an obligation attribute.

      An Obligation has a set of attributes. Each attribute is a name-value pair. The value can be either static or the value of a previously defined attribute. Each obligation should have at least one attribute. See Section 3.5.9, "Managing Attributes and Functions as Extensions" for information.

d. Provide a Name for the attribute in the New Obligation Attribute dialog.

   If the obligation attribute is static, select either String, Integer, Boolean, Date or Time for Data Type and provide a Value. If the obligation is an attribute, select Attribute for Data Type and choose from the list of predefined attributes.

e. Click Add.

7. Click Save to save the Authorization Policy.

3.5.5.2 Modifying an Authorization Policy

To modify a policy, proceed as follows:

1. Choose from the following methods to display the desired Authorization Policy.
   - Expand the information tree in the Navigation Panel to find the Authorization Policies node under the appropriate Application’s Policy Domain and double click it. A search dialog opens in the Home area.
   - In the Home area, select the Application Name under which the Authorization Policy was created and click Search under Authorization Policies. A search dialog opens in the Home area.
2. Enter query parameters and click Search.

   The search results are displayed.
3. Select the appropriate Authorization Policy and click Open to display the details.
4. Modify the policy as necessary.
   - Select the Principal to modify.

     For more information, see Section 3.5.5.1, "Creating an Authorization Policy."
   - Select (or expand) the Target to modify.

     For more information, see Section 3.5.5.1, "Creating an Authorization Policy."
   - Click the Conditions tab to edit conditions.

     For more information, see Section 3.6, "Using the Condition Builder."
   - Click the Obligations tab to modify the Obligation or its attributes.
     - To modify the obligation, click Edit from the Obligations table, make changes in the displayed dialog and click Update.
     - To modify an attribute, select the attribute from the Attributes table and click Edit. Make changes in the displayed dialog and click Update.
     - To delete the Obligation, select it in the Obligations table and click Remove.
5. Click Apply.

3.5.5.3 Deleting an Authorization Policy

To delete an Authorization Policy, proceed as follows:

1. Choose from the following methods to display the Authorization Policy search screen.

- Expand the information tree in the Navigation Panel to find the Authorization Policies node under the appropriate Application's Policy Domain, right-click it and select Open. A search dialog opens in the Home area.
- In the Home area, select the Application Name under which the Authorization Policy was created and click Search under Authorization Policies. A search dialog opens in the Home area.

For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."

2. Enter query parameters and click Search.

   The search results are displayed.

3. Select the appropriate Authorization Policy from the search results and click Delete.

3.5.6 Managing Application Roles in the Role Catalog

Application Roles are defined at the Application level (thus, its name) and created using Oracle Entitlements Server. An Application Role can be assigned to an External Role, user, or group in an identity store, or another Application Role in the policy store. One target application may have several different roles, with each assigned a different set of privileges for more fine-grained authorization. Membership can be granted statically to External Roles or individual users, or dynamically using a Role Mapping Policy that is processed at runtime.

Note: A Role Mapping Policy assigns the role to subjects and an Authorization Policy defines the role's access rights.

You can use Application Roles to control access by establishing relationships with the following procedure:

1. Define Application Roles to represent the functional roles users have in the application.
2. Map each Application Role to External Roles or individual Users.
3. Create Authorization Policies to provide the level of access rights (Permit/Deny) required to meet the goals of the Application Roles.
4. Add the Application Role as a Principal to one or more Authorization Policies.

Application Roles use role inheritance and hierarchy. The inheritance pattern is such that a subject assigned to a role (using a Role Mapping Policy or static role assignments) also inherits any child roles if it is not prohibited by Role Mapping Policies. When an Application Role is referenced as a Principal in a policy, access to the resource for all users assigned to the role is governed by the policy. The following sections describe management operations on Application Roles.

- Section 3.5.6.1, "Creating an Application Role"
- Section 3.5.6.2, "Modifying an Application Role"
- Section 3.5.6.3, "Mapping External Roles to an Application Role"
- Section 3.5.6.4, "Mapping External Users to an Application Role"
- Section 3.5.6.5, "Deleting an Application Role or Removing External Role Mappings"

3.5.6.1 Creating an Application Role

The following procedure describes the steps to create a new Application Role. You are not required to add members to the role at the same time and can return to the saved role later. To create an Application Role, proceed as follows:

1. Display the page for creating an Application Role by choosing one of the following methods:
   - Navigate to the Role Catalog under the appropriate Application node in the Navigation Panel. Right-click the Role Catalog node and select New from the menu.
   - In the Home area, select the Application Name under which the Application Role will be created and click New from Application Roles.

   An Untitled page with four tabs is displayed in the Home area: General (active), Application Role Hierarchy, External Role Mapping and External User Mapping.
2. Provide the following information under the General tab.
   - **Display Name** : The display name is optional and case insensitive. Specifying a meaningful display name is recommended since it is displayed in the Administration Console, and provides extra information to help administrators identify objects.
   - **Role Name :** The name is required and case insensitive. At runtime, this is the string the application passes to determine whether a user is authorized to access this Resource.
   - **Description** : Although optional, it is recommended to provide useful information about the entitlement. The description string is case insensitive.
   - **Role Category** : A Role Category is a tag you can assign to a role for ease of management. See Section 3.5.8, "Managing a Role Category."
3. Click Save.

   The page is renamed to match the entry provided for Role Name and the Application Role Hierarchy, External Role Mapping and External User Mapping tabs become active. At this point, you can create a policy with this Application Role as the Principal or find a policy with this Application Role as the Principal by clicking Create Policies or Find Policies, respectively. To define the Application Role Hierarchy continue to the next step.
4. Optionally, select the Application Role Hierarchy tab to define from which roles this Application Role will inherit permissions (Inherits) and for which roles this Application Role will define permissions (Is Inherited By). Hierarchy is not required but if you choose to define it, the following example sub procedure is specific to the former option.

   a. Click Inherits.

   b. Click Add.

   c. Select the radio button that corresponds to the role to which you are adding the hierarchy.

      When you add roles to the hierarchy, you can either add the roles to the role under which you are working or to a role that you can select in the Application Role Hierarchy table.

   d. Complete the criteria fields in the Add a Role dialog and click Search.

      The results display in the Search Results table. Empty strings fetch all roles.

e. Select the role from which this role will inherit permissions in the Search Results table.

   Use Ctrl+click to select multiple roles.

f. Click Add.

   The selected roles display in the Application Role Hierarchy tab, and the Application Role inherits permissions from them.

For information about external role mapping, see Section 3.5.6.3, "Mapping External Roles to an Application Role." For information about external user mapping, see Section 3.5.6.4, "Mapping External Users to an Application Role."

3.5.6.2 Modifying an Application Role

To modify or view an Application Role, proceed as follows:

1. Choose from the following methods to display the desired Application Role.

   - Expand the information tree in the Navigation Panel to find the Role Catalog node under the appropriate Application, right-click it and select Open. A search dialog opens in the Home area.
   - In the Home area, select the Application Name under which the Application Role was created and click Search under Application Roles. A search dialog opens in the Home area.

   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."

2. Enter query parameters and click Search.

   The search results are displayed.

3. Select the appropriate Application Role and click Open to display the details.

4. Select the tab that contains the parameters you want to modify and click Add.

   For information on the available tabs, see:

   - Application Role Hierarchy : Creating an Application Role
   - External Role Mapping : Mapping External Roles to an Application Role
   - External User Mapping : Mapping External Users to an Application Role

3.5.6.3 Mapping External Roles to an Application Role

To map external roles to an application role, proceed as follows:

1. Choose from the following methods to display the desired Application Role.

   - Expand the information tree in the Navigation Panel to find the Role Catalog node under the appropriate Application and double click it. A search dialog opens in the Home area.
   - In the Home area, select the Application Name under which the Application Role was created and click Search under Application Roles. A search dialog opens in the Home area.

   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."

2. Enter query parameters and click Search.

   The search results are displayed.

3. Select the appropriate Application Role and click Open to display the details.

   Alternately, search for Application Roles using the Navigation Panel's search function and double-click the Application Role name in the Search Results tab to display the details.

4. Select the External Role Mapping tab.

5. Click Add to display the Add a Role dialog.

6. Complete the query fields in the Add a Role dialog and click Search.

   Empty strings fetch all roles. The results display in the External Role Search table.

7. Select the external role to map to by clicking its name in the table.

   Use Ctrl+click to select multiple roles.

8. Click Map Roles.

   The selected roles display in the External Role Mapping tab.

3.5.6.4 Mapping External Users to an Application Role

To map an external user to an application role, proceed as follows:

1. Choose from the following methods to display the desired Application Role.

   - Expand the information tree in the Navigation Panel to find the Role Catalog node under the appropriate Application and double click it. A search dialog opens in the Home area.
   - In the Home area, select the Application Name under which the Application Role was created and click Search under Application Roles. A search dialog opens in the Home area.

   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."

2. Enter query parameters and click Search.

   The search results are displayed.

3. Select the appropriate Application Role and click Open to display the details.

   Alternately, search for Application Roles using the Navigation Panel's search function and double-click the Application Role name in the Search Results tab to display the details.

4. Select the External User Mapping tab.

5. Click Add to display the Add a User dialog.

6. Complete the query fields in the Add a User dialog and click Search.

   Empty strings fetch all roles. The results display in the External User Search table.

7. Select the user to map by selecting its name in the table.

   Use Ctrl+click to select multiple roles.

8. Click Map Users.

   The selected roles display in the External User Mapping tab.

3.5.6.5 Deleting an Application Role or Removing External Role Mappings

To delete an Application Role or remove External Role Mapping from an Application Role, proceed as follows:

1. Choose from the following methods to display the desired Application Role.
   - Expand the information tree in the Navigation Panel to find the Role Catalog node under the appropriate Application, right-click it and select Open. A search dialog opens in the Home area.
   - In the Home area, select the Application Name under which the Application Role was created and click Search under Application Roles. A search dialog opens in the Home area.

   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."
2. Enter query parameters and click Search.

   The search results are displayed.
3. Select the appropriate Application Role and click Open to display the details.

   Alternately, search for Application Roles using the Navigation Panel's search function and double-click the Application Role name in the Search Results tab to display the details.
4. Select the Application Role in the Search Results table and:
   - Click Delete to remove the role.
   - Select the appropriate mapping in the External Role Mapping table and click Remove.

3.5.7 Managing Role Mapping Policies

Membership to an Application Role can be granted statically or dynamically with a Role Mapping Policy. An Application Role, referenced in a Role Mapping Policy, could grant a user access to the defined resources. The following sections describe management operations on Role Mapping Policies.

- Section 3.5.7.1, "Creating a Role Mapping Policy"
- Section 3.5.7.2, "Modifying a Role Mapping Policy"
- Section 3.5.7.3, "Deleting a Role Mapping Policy"

3.5.7.1 Creating a Role Mapping Policy

To create a Role Mapping Policy, proceed as follows:

1. Display the page for creating a Role Mapping Policy by choosing one of the following methods:
   - Navigate to the appropriate Application node in the Navigation Panel and expand the Role Catalog branch. Right-click Role Mapping Policies and select New from the menu.
   - In the Home area, select the Application Name under which the Role Mapping Policy will be created and click New from Role Mapping Policies.

   An Untitled page is displayed in the Home area.
2. Provide the following information.
   - **Effect**: Select Permit if the policy will grant rights or Deny if the policy will deny rights.
   - **Display Name** : The display name is optional and case insensitive. Specifying a meaningful display name is recommended since it is displayed in the

Administration Console, and provides extra information to help administrators identify objects.

- **Name :** The name is required and case insensitive.
- **Description** : Although optional, it is recommended to provide useful information about the policy. The description string is case insensitive.

3. Choose one of the following methods to add Application Roles.

   - Drag and drop

     a. Use the Navigation Panel to list the Application's available Application Roles by performing a search. For more information, see Section 4.2, "Finding Objects with a Simple Search."

     b. Drag and drop Application Roles from the Search Results tab into the area labeled App Role.

   - Add Application Roles dialog

     a. Click Add in the App Role section.

        The Search Application Roles dialog displays.

     b. Search for the available Application Roles by entering a string.

        The resources matching the query are displayed in Search Results.

     c. Select the principals to add and click Add Application Roles.

        Use Ctrl+click to select multiple items from the list.

   Note: For this release, this dialog displays the Search Principals title and Add Principals button.

4. Choose one of the following methods to add Principals.

   - Drag and drop

     a. Use the Navigation Panel to list the Application's available Users and External Roles by performing a search. For more information, see Section 4.2, "Finding Objects with a Simple Search."

     b. Drag and drop Users and External Roles from the Search Results tab into the area labeled Principals.

   - Add Principals dialog

     a. Click Add in the Principals section.

        The Search Principals dialog displays.

     b. Search for the available Principals (in this case, Users or External Roles) by entering a string.

        The resources matching the query are displayed in Search Results.

     c. Select the principals to add and click Add Principals.

        Use Ctrl+click to select multiple items from the list.

5. Optionally, choose one of the following methods to add Resources (also referred to as Targets).

   - Drag and drop

  a. Use the Navigation Panel to list the Application's available Resources by performing a search. For more information, see Section 4.2, "Finding Objects with a Simple Search."

  b. Drag and drop one or more Resources from the Search Results tab into the area labeled Resources.

- Add Targets pop up search

  a. Click Add in the Resources section.

     The Add Targets dialog displays.

  b. Choose the Policy Domain that contains the Resource (if applicable).

  c. Enter a string and click Search.

     The resources matching the query are displayed in Search Results. If no search string was entered, a list of all objects of the specified type is returned.

  d. Select the appropriate Targets to add and click Add Selected.

     The Target(s) are added to the Selected Targets. Use Ctrl+click to select multiple items from the list.

  e. Click the Resource Expression link to add an expression as a Target.

     Select a Resource Type, enter a string expression and click Add to Targets. This will search for targets, using the defined criteria, dynamically at runtime. All Resources that belong to the selected Resource Type that contain the string expression are returned, within the context of the administrator privileges.

  f. Click Add Targets.

6. See Section 3.6, "Using the Condition Builder" for information on using the Condition Builder.

7. Click Save.

3.5.7.2 Modifying a Role Mapping Policy

To modify a Role Mapping Policy, proceed as follows:

1. Choose from the following methods to display the desired Role Mapping Policy.

   - Expand the information tree in the Navigation Panel to find Role Mapping Policies under the Role Catalog node of the appropriate Application and double click Role Mapping Policies. A search dialog opens in the Home area.

   - In the Home area, select the Application Name under which the Role Mapping Policy was created and click Search under Role Mapping Policies. A search dialog opens in the Home area.

   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."

2. Enter query parameters and click Search.

   The search results are displayed.

3. Select the appropriate Role Mapping Policy and click Open to display the details.

4. Modify the policy as necessary.

5. Click Apply.

3.5.7.3 Deleting a Role Mapping Policy

To delete a Role Mapping Policy, proceed as follows:

1. Choose from the following methods to display the desired Role Mapping Policy.
   - Expand the information tree in the Navigation Panel to find Role Mapping Policies under the Role Catalog node of the appropriate Application and double click Role Mapping Policies. A search dialog opens in the Home area.
   - In the Home area, select the Application Name under which the Application Role was created and click **Search** under **Role Mapping Policies**. A search dialog opens in the Home area.

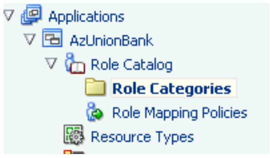
   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."
2. Enter query parameters and click Search.

   The search results are displayed.
3. Select the appropriate Role Mapping Policy and click Open to display the details.
4. Click Delete in the upper right corner of the Home area.

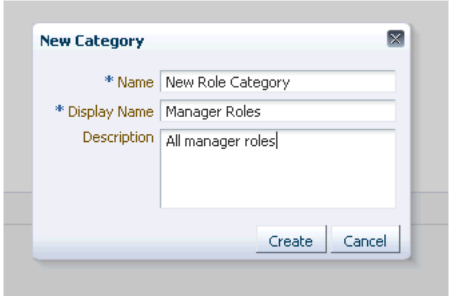
3.5.8 Managing a Role Category

A Role Category is a tag you can assign to a role for ease of management. You can create or delete a Role Category but you cannot modify them. To create a Role Category, proceed as follows. Instructions to delete a Role Category are detailed after the final step.

1. Expand the appropriate Application node in the Navigation Panel and double-click the Role Categories node.



   The Role Categories page opens in the Home area.
2. Click New to display the New Category dialog.
3. Provide the following information.
   - **Name**
   - **Display Name**
   - **Description**

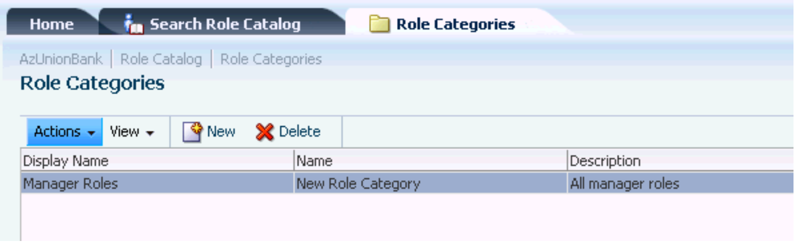


New Category

* Name	New Role Category
* Display Name	Manager Roles
Description	All manager roles

4. Click Create.

   The new category displays in the Role Categories list.



Display Name	Name	Description
Manager Roles	New Role Category	All manager roles

To delete a Role Category, expand the appropriate Application node in the Navigation Panel and double-click the Role Categories node. Select the Role Category to delete and click Delete.

3.5.9 Managing Attributes and Functions as Extensions

Attributes and Functions are definitions organized under the Extensions node of the Application for which they were created. Attribute and function definitions can be used in a Condition or an Obligation. In regards to a Condition, attribute and function definitions can be used to make an optional expression that can be added to a policy to further restrict access to the protected resource. In regards to an Obligation, this optional set of name-value pairs returns additional information, with a policy decision, to the calling application. There are two ways to define an Obligation:

- Statically where an attribute with an absolute value is returned.
- Dynamically where an attribute value, or a custom function, is evaluated at runtime and the output is returned.

An Attribute can be a value dynamically defined at runtime (for example, the locality of the user) or a value based on the type of protected resource (for example, creation date of a text file). During policy evaluation, attribute values can be passed in by the application or Oracle Entitlements Server can retrieve it using a custom attribute retriever. Attributes must have a defined type. Boolean, integer, date, time and string are Oracle Entitlements Server predefined types. An attribute may be singular or a multi-valued list. A Function is a definition of externally implemented logic. It can be added to a policy as a condition on the policy's outcome. The following sections describe management operations on Attributes and Functions.

- Section 3.5.9.1, "Creating an Attribute"
- Section 3.5.9.2, "Modifying an Attribute"
- Section 3.5.9.3, "Deleting an Attribute"
- Section 3.5.9.4, "Creating a Function"
- Section 3.5.9.5, "Modifying a Function"
- Section 3.5.9.6, "Deleting a Function"

3.5.9.1 Creating an Attribute

To create an attribute, proceed as follows:

1. Navigate to, and expand, Extensions under the appropriate Application node in the Navigation Panel.
2. Right-click the Attributes node and select New from the menu.

An Untitled page is displayed in the Home area.

3. Provide the following information for the attribute.

   - **Display Name** : The display name is optional and case insensitive. Specifying a meaningful display name is recommended since it is displayed in the Administration Console, and provides extra information to help administrators identify objects.
   - **Name :** The name is required and case insensitive. At runtime, this is the string the application passes to determine whether a user is authorized to access this Resource.
   - **Description** : Although optional, it is recommended to provide useful information about the entitlement. The description string is case insensitive.
   - **Category**: Select from Resource and Dynamic as a value for this required parameter.
   - **Type**: Select from String, Date, Integer, Boolean, Time.
   - **Input Values**: Select from Single and Multiple.

4. Select one of the following from the Save menu.

   - Save and Close saves the configuration and renames the page with the value provided for the Display Name.
   - Save and Create Another saves the configuration to the information tree in the Navigation Panel but leaves the Untitled area open for you to create another Attribute.

3.5.9.2 Modifying an Attribute

To modify an attribute, proceed as follows:

1. Navigate to, and expand, Extensions under the appropriate Application node in the Navigation Panel.
2. Double-click Attributes to open a search dialog in the Home area.

   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."

3. Enter query parameters and click Search.

   The search results are displayed.

4. Select the appropriate Attribute and click Open to display the details.

   Alternately, search for Attributes using the Navigation Panel’s search function and double-click the Attribute name in the Search Results tab to display the details.

5. Modify the attribute as necessary.
6. Click Apply.

3.5.9.3 Deleting an Attribute

To delete an attribute, proceed as follows:

1. Navigate to, and expand, Extensions under the appropriate Application node in the Navigation Panel.

   - Navigate to, and expand, Extensions under the appropriate Application node in the Navigation Panel. Right-click Attributes and select Open to display a search dialog in the Home area. Enter criteria for the lookup and click Search.

For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."

- Search for Attributes using the Navigation Panel's search function, right-click the Attribute name in the Search Results tab and select Open to display the Attribute in the Home area. For information about searching in the Navigation Panel, see Section 4.2, "Finding Objects with a Simple Search."

2. Double-click Attributes to open a search dialog in the Home area.

   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."

3. Enter query parameters and click Search.

   The search results are displayed.

4. Select the appropriate Attribute in the Search Results table.

   Alternately, search for Attributes using the Navigation Panel's search function and double-click the Attribute name in the Search Results tab to display the details.

5. Click Delete in the Search Results table or the Attribute details page.

   A Delete Warning is displayed.

6. Click Yes.

3.5.9.4 Creating a Function

To create a function, proceed as follows:

1. Navigate to, and expand, Extensions under the appropriate Application node in the Navigation Panel.

2. Right-click the Functions node and select New from the menu.

   An Untitled page is displayed in the Home area.

3. Provide the following information for the function.

   - **Display Name** : The display name is optional and case insensitive. Specifying a meaningful display name is recommended since it is displayed in the Administration Console, and provides extra information to help administrators identify objects.
   - **Name :** The name is required and case insensitive. At runtime, this is the string the application passes to determine whether a user is authorized to access this Resource.
   - **Description** : Although optional, it is recommended to provide useful information about the entitlement. The description string is case insensitive.
   - **Function Class Name**: The name of the class that provides the functionality.
   - **Input Parameter**: A list of the types of parameters passed to the function.
   - **Return Type**: Select the data type returned by the function.
   - **Syntax Preview** displays a preview of the function's syntax.

4. Select one of the following from the Save menu.

   - Save and Close saves the configuration and renames the page with the value provided for the Display Name.
   - Save and Create Another saves the configuration to the information tree in the Navigation Panel but leaves the Untitled area open for you to create another.

3.5.9.5 Modifying a Function

To modify a function, proceed as follows:

1. Navigate to, and expand, Extensions under the appropriate Application node in the Navigation Panel.
2. Double-click Functions to open a search dialog in the Home area.

   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."
3. Enter query parameters and click Search.

   The search results are displayed.
4. Select the appropriate Function in the Search Results table and click Open to display the details.

   Alternately, search for Functions using the Navigation Panel’s search function and double-click the Function name in the Search Results tab to display the details.
5. Modify the Function as necessary.
6. Click Apply.

3.5.9.6 Deleting a Function

To delete a Function, proceed as follows:

1. Navigate to, and expand, Extensions under the appropriate Application node in the Navigation Panel.
2. Double-click Functions to open a search dialog in the Home area.

   For information about searching in the Home area, see Section 4.3, "Finding Objects with an Advanced Search."
3. Enter query parameters and click Search.

   The search results are displayed.
4. Select the appropriate Function in the Search Results table.

   Alternately, search for Functions using the Navigation Panel’s search function and double-click the Function name in the Search Results tab to display the details.
5. Click Delete in the Search Results table or the Function’s detail page.

   A Delete Warning is displayed.
6. Click Yes.

3.6 Using the Condition Builder

An optional Condition in a policy rule can be used to further evaluate the applicability of an authorization decision returned in response to a request for access. For example, a Condition can be used to grant access to a resource only on the condition that the request was issued from a specific location or at a specific time.

Note: Conditions in Role Mapping Policies provide the same functionality, and take the same format, as conditions in Authorization Policies.

A Condition is written in the form of an expression that resolves to either true or false. If the expression resolves to true, the condition is satisfied and the policy is applicable. If the expression does not resolve to true, the policy is not applicable. The expression can operate on attributes, functions or literals. Oracle Entitlements Server contains predefined attributes and functions that can be inserted or you can create custom ones. The literals belong to the supported data types and are constants.

> **Note:** All Attributes and Functions (both custom and predefined) are created, collected and further managed under the Extensions node of the Application. For more information, see Section 3.5.9, "Managing Attributes and Functions as Extensions."

The Condition Builder allows an administrator to quickly create Condition expressions that can then be added to an Authorization Policy or a Role Mapping Policy. The following procedure illustrates how to use the Condition Builder to create a Condition for your policy. To create a Condition, you either create or modify an Authorization Policy or a Role Mapping Policy. Following one of these procedures will bring you to a step in which you can build a Condition.

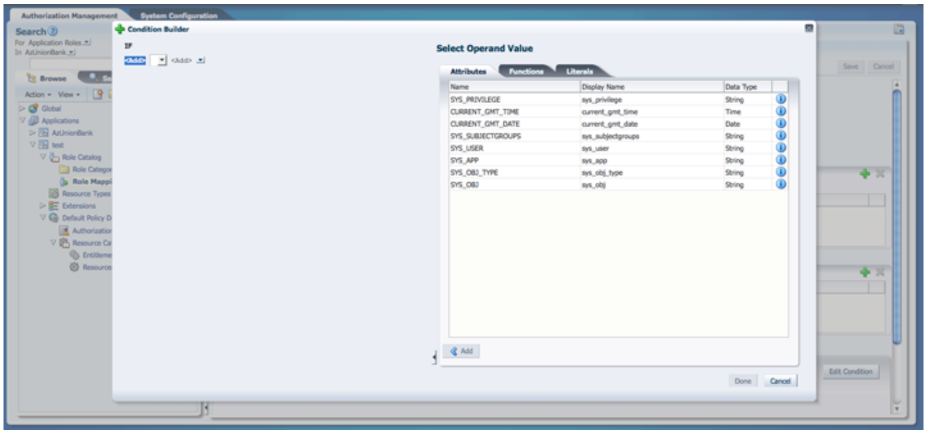
- Section 3.5.5.1, "Creating an Authorization Policy"
- Section 3.5.5.2, "Modifying an Authorization Policy"
- Section 3.5.7.1, "Creating a Role Mapping Policy"
- Section 3.5.7.2, "Modifying a Role Mapping Policy"

When you get to the appropriate screen, follow this procedure.

1. Click the Condition tab.
2. Click Edit Condition.

   The Condition Builder (as displayed in Figure 3–1) displays. Note the frame of the Condition expression on the left. The frame contains two Add replaceables and an operator drop down. (The drop down is empty until an operand has been added.) The tabs for expression components - Attributes, Functions and Literals - are on the right. You will add components from these tabs to the Expression frame to build your Condition.

Figure 3–1 The Condition Builder

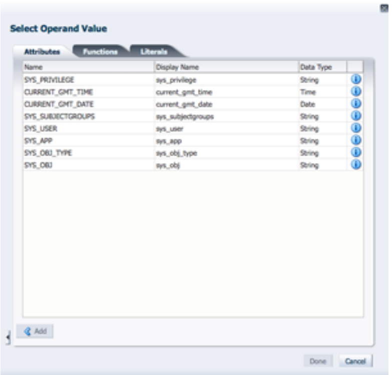


Name	Display Name	Data Type
SYS_PRIVILEGE	sys_privilege	String
CURRENT_GMT_TIME	current_gmt_time	Time
CURRENT_GMT_DATE	current_gmt_date	Date
SYS_SUBJECTGROUPS	sys_subjectgroups	String
SYS_USER	sys_user	String
SYS_APP	sys_app	String
SYS_OBJ_TYPE	sys_obj_type	String
SYS_OBJ	sys_obj	String

3. Click the tab that contains the component type you want to add to the Condition.

Figure 3–2 is a screen shot of the Operand Value tabs. The Attributes and Function listed in these tabs are filtered based on the Application in which the policy is being created. For example, a custom Function created within Application 1 will not be visible when the Condition Builder is activated to create a policy within Application 2.

Figure 3–2 Operand Value Tabs

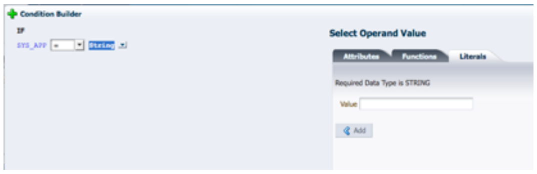


Name	Display Name	Data Type
SYS_PRIVILEGE	sys_privilege	String
CURRENT_GMT_TIME	current_gmt_time	Time
CURRENT_GMT_DATE	current_gmt_date	Date
SYS_SUBJECTGROUPS	sys_subjectgroups	String
SYS_USER	sys_user	String
SYS_APP	sys_app	String
SYS_OBJ_TYPE	sys_obj_type	String
SYS_OBJ	sys_obj	String

4. Select the line that contains the component you want to add to the Condition and click Add.

   Click the blue **i** to display a Details box with more information regarding the component. Figure 3–3 is a screenshot after having added a SYS_APP attribute which takes a string value.

Figure 3–3 Adding a Literal to the Condition



IF

SYS_APP = String

Required Data Type is STRING

Value

5. Populate the value on the right of the expression by selecting the appropriate Operand Value and click Add.

6. Specify the operator on the right of the Condition Builder by clicking the drop down and selecting your choice.

   The operator options are dependent on the Operand Value.

7. Add additional expressions by clicking the last arrow in the expression and selecting AND, OR or NOT from the crop down menu, if applicable.

   REMOVE will clear the expression of all components so you may begin again.

8. Select components for the additional expression from the appropriate Operand Value tabs, if applicable.

   You may add as many expressions (and components) as necessary by clicking the last arrow in the current expression and selecting from the Operand Value tabs.

9. Click Done to complete the Condition.

The following points should be taken into account as you navigate the Condition Builder to create your expression.

- The Condition Builder contains Tool Tips on most fields for additional details.
- Click the appropriate blue **i** for information on the Operand Value.
- At the minimum, an expression must contain two operands and an operator.
- You can compare an Attribute and an Attribute, an Attribute and a Function, an Attribute and a Literal, a Function and a Function, and a Function and a Literal.
- The input parameters for Functions can be Attributes, Literals or Functions.
- The choice of operators displayed is directly related to the first operand chosen. For example, you cannot do less than or equal to on a string.
- The choice of a second Operand Values displayed within an expression is also directly related to the first operand chosen.
- REMOVE clears the expression to which it is tied of all components so you may begin again. It does not clear the entire Condition.
- The completed Condition (expression) is evaluated by Oracle Entitlements Server at runtime. The interpretation is governed by the rules of precedence.
- The outcome of this Condition must be a boolean.

The following sections contain procedures for more complex conditions.

- Section 3.6.1, "Building a Complex Expression"
- Section 3.6.2, "Passing Parameters to Functions"

3.6.1 Building a Complex Expression

This procedure explains how you might build a complex expression using parenthesis.

1. Follow one of these procedures to bring you to the Condition Builder.
   - Section 3.5.5.1, "Creating an Authorization Policy"
   - Section 3.5.5.2, "Modifying an Authorization Policy"
   - Section 3.5.7.1, "Creating a Role Mapping Policy"
   - Section 3.5.7.2, "Modifying a Role Mapping Policy"
2. Click the Condition tab.
3. Click Edit Condition.

   The Condition Builder displays as in Figure 3–1.
4. Click the Attributes tab.
5. Select the `DateAttr` custom attribute and click Add.

   `DateAttr` is not a predefined Oracle Entitlements Server attribute so this step assumes a custom attribute has been defined as documented inSection 3.5.9, "Managing Attributes and Functions as Extensions." `DateAttr` is added to the left of the operator.
6. Select the equal sign (=) as the operator.
7. Select the `CURRENT_GMT_DATE` predefined attribute and click Add.

   `CURRENT_GMT_DATE` is a predefined Oracle Entitlements Server attribute and can be viewed under the Attributes tab. It is added to the right of the operator.

8. Add more complexity to the Condition by selecting the appropriate AND, OR or NOT operation at the end of the line of code.

   Parentheses must match; there must be an equal number of open and closing parentheses. If you select an operation at the end of a line of code, the operation will involve the code itself. If you select an operation at the end of the entire Condition, it will allow you to add on to the Condition as a whole.

9. Add additional conditions by choosing values from Attributes, Functions or Literals as necessary.

10. Click Done when finished.

3.6.2 Passing Parameters to Functions

This procedure describes how to pass parameters into a Function.

1. Follow one of these procedures to bring you to the Condition Builder.

   - Section 3.5.5.1, "Creating an Authorization Policy"
   - Section 3.5.5.2, "Modifying an Authorization Policy"
   - Section 3.5.7.1, "Creating a Role Mapping Policy"
   - Section 3.5.7.2, "Modifying a Role Mapping Policy"

2. Click the Condition tab.

3. Click Edit Condition.

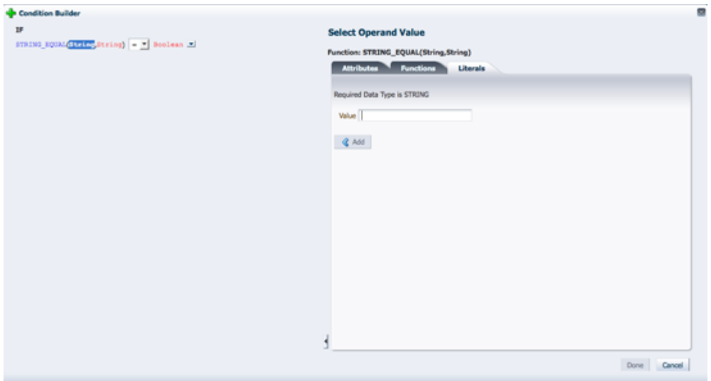
   The Condition Builder displays as in Figure 3–1.

4. Click the Functions tab.

5. Select STRING_EQUAL and click Add.

   Figure 3–4 illustrates an added Function and contains placeholders for the two parameters that must be passed to it. This Function will compare the two strings (one the value of a predefined attribute).

Figure 3–4 Adding a Function


IF
STRING_EQUAL(String,String) = Boolean

Function: STRING_EQUAL(String,String)

Required Data Type is STRING

Value

6. Select the first parameter if not already.

7. Click the Attributes tab.

8. Select SYS_USER and click Add.

   The second parameter is highlighted and the Literal tab is activated.

9. Enter a value for the second parameter and click Add.

   For this example, joe. The boolean to the right of the operator is highlighted and the Literal tab is activated.

10. Choose the appropriate operator.

11. Click the Boolean replaceable and select whether this function output should be true or false.

12. Add Additional operands as you see fit.

13. Click Done when finished.

4 Searching for Security Objects

Oracle Entitlements Server enables searching for policies and security objects from within the Authorization Policy Manager Administration Console. The following sections explain the types of searches and for what purposes they can be used.

- Searching with the Administration Console
- Finding Objects with a Simple Search
- Finding Objects with an Advanced Search

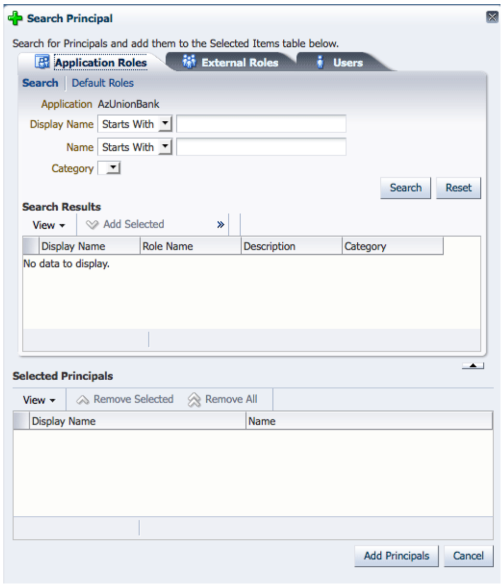
4.1 Searching with the Administration Console

Oracle Entitlements Server enables different kinds of search queries using the Authorization Policy Manager Administration Console.

- A *simple search* matches names and display names only. The search is generated from the top of the Navigation Panel and results are displayed in the Navigation Panel. For more information, see Section 4.2, "Finding Objects with a Simple Search."
- An *advanced search* uses operators that enable more sophisticated matching. The advanced search screen is launched by double-clicking an object in the Navigation Panel, or from the Home area. The search box opens in the Home area and results are also displayed there. For more information, see Section 4.3, "Finding Objects with an Advanced Search."
- A *pop-up search* opens from within the Authorization Policy or Role Mapping Policy screens, when the policy is being created or modified, by clicking the green Add button (plus sign). The pop-up search box uses a shopping cart paradigm. You add choices selected from the multiple, displayed tabs on the top of the search box to the Selected box on the bottom of the search box. All choices in the Selected box are added when you click Add.

  Figure 4–1 is a screen shot of the pop-up search box for adding a Principal. You can click between the three tabs (Application Roles, External Roles, and Users), selecting one or more policy subjects and adding them to the Selected Principals box. When you click Add Principals, all choices added from all tabs will then be added to the policy.

Figure 4–1 Pop-up Search Box



Application AzUnionBank

Display Name Starts With

Name Starts With

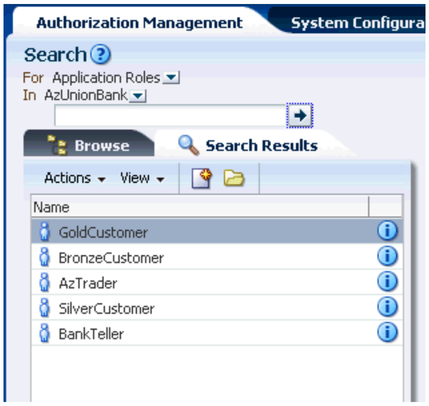
Category

No data to display.

4.2 Finding Objects with a Simple Search

A simple search matches names and display names only. The fields in the top portion of the Authorization Management tab in the Navigation Panel, as shown in Figure 4–2, are used to specify simple queries.

Figure 4–2 Simple Search Fields and Results Tab in Navigation Panel



For Application Roles

In AzUnionBank

GoldCustomer

BronzeCustomer

AzTrader

SilverCustomer

BankTeller

To specify a simple search, proceed as follows:

1. Select the policy object for which you are searching from the **For** list.

   The following object types are available:

   - Application Roles
   - External Roles
   - Users
   - Resources
   - Resource Types

- Entitlements
- Attributes

2. Select the search scope from the **In** list.

   The *search scope* defines the level at which the search will take place. When searching for Application Roles, Resources, Resource Types, Entitlements and Attributes, the search scope is an Application. For External Roles and Users, the search scope is Global. For Entitlements and Resources, the search scope is the Policy Domain within an Application.

   > **Note:** If performing a Resource search, you also select the Resource Type from the Type list.

3. Optionally, enter a string to match in the text box.

   Wildcard characters percent (%) and asterisk (*) are supported for a simple search.

4. Click the arrow icon next to the text box to begin the search.

   Names and display names matching the specified criteria are returned and displayed in the Search Results tab. If no search string was entered, a list of all objects of the specified type is returned.

5. Double-click the object to edit, right click the object and select New to create, or click the object's information icon for details.

   For more information on managing policy objects, see Chapter 3, "Managing Policies and Policy Objects."

4.3 Finding Objects with an Advanced Search

An advanced search is generally initiated by double-clicking the object name in the Navigation Panel, or from the Search link for the object in the Home area. An advanced search can use the following operators:

- Starts with
- Ends with
- Contains
- Equal to

There is no support for wildcard characters in an advanced search. In particular, the asterisk (*) or percent (%) characters are treated as plain text in any advanced search parameter. The following sections have information on searching for policy objects with an advanced search.

- Section 4.3.1, "Searching External Roles"
- Section 4.3.2, "Searching Applications"
- Section 4.3.3, "Searching Resource Types"
- Section 4.3.4, "Searching Application Roles"
- Section 4.3.5, "Searching Role Mapping Policies"
- Section 4.3.6, "Searching Resources"
- Section 4.3.7, "Searching Entitlements"

- Section 4.3.8, "Searching Authorization Policies"
- Section 4.3.9, "Searching Attributes"
- Section 4.3.10, "Searching Functions"
- Section 4.3.11, "Searching for Users Globally"

4.3.1 Searching External Roles

To search External Roles, proceed as follows:

1. Select from the following methods to display the Search External Roles page:
   - In the Navigation Panel, expand Global and double-click External Roles.

     Alternately, right-click External Roles and select Open.
   - In the Home area, click Search - External Roles from the Search and Create section.
2. Enter the following query parameters:
   - **Name**: Select an operator from the list and enter a string to match.
   - **Display Name**: Select an operator from the list and enter a string to match.

   Optionally, select from the Saved Search drop-down list of previously saved searches. Its query parameters automatically populate the search fields. Select Personalize... to set options for previously saved searches.
3. Optionally, click Save... to name the current query parameters.

   The named search is added to the Saved Search list.
4. Click Search.

   The results are displayed in Search Results.

4.3.2 Searching Applications

To search through Applications, proceed as follows:

1. Select from the following methods to display the Search Applications page:
   - In the Navigation Panel, double-click Applications to display the Search Applications page.

     Alternately, right-click Applications and select Open.
   - In the Home area, click Search - Applications from the Search and Create section.
2. Enter the following query parameters:
   - **Name**: Select an operator from the list and enter a string to match.
   - **Display Name**: Select an operator from the list and enter a string to match.

   Optionally, select from the Saved Search drop-down list of previously saved searches. Its query parameters automatically populate the search fields. Select Personalize... to set options for previously saved searches.
3. Optionally, click Save... to name the current query parameters.

   The named search is added to the Saved Search list.
4. Click Search.

The results are displayed in Search Results.

4.3.3 Searching Resource Types

To search Resource Types, proceed as follows:

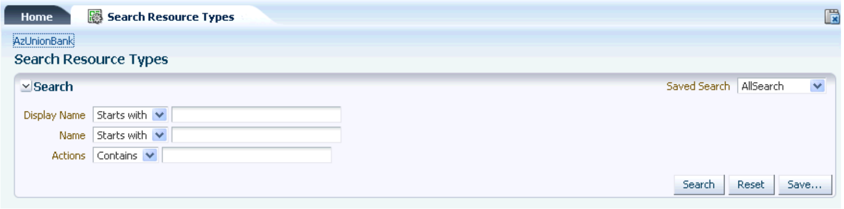
1. Select from the following methods to display the Search Resource Types page as in Figure 4–3.

   - In the Navigation Panel, expand the Application node and double-click Resource Types.

     Alternately, right-click Resource Types and select Open.

   - In the Home area, select the appropriate Application Name and click Search under Resource Types.

Figure 4–3 Searching for Resource Types



Saved Search: AllSearch

Display Name: Starts with

Name: Starts with

Actions: Contains

2. Enter the following query parameters:

   - **Display Name:** Select an operator from the list and enter a string to match.
   - **Name**: Select an operator from the list and enter a string to match.
   - **Actions**: Select an operator from the list and enter a string to match.

   Optionally, select from the Saved Search drop-down list of previously saved searches. Its query parameters automatically populate the search fields. Select Personalize... to set options for previously saved searches.

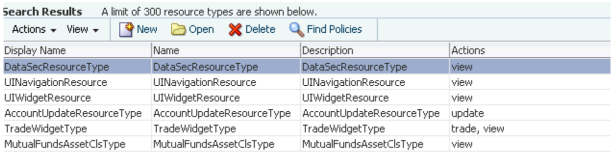
3. Optionally, click Save... to name the current query parameters.

   The named search is added to the Saved Search list.

4. Click Search.

   All results matching the query specifications are displayed in the Search Results table as illustrated in Figure 4–4.

Figure 4–4 Resource Type Search Results



A limit of 300 resource types are shown below.

Display Name	Name	Description	Actions
DataSecResourceType	DataSecResourceType	DataSecResourceType	view
UINavigationResource	UINavigationResource	UINavigationResource	view
UIWidgetResource	UIWidgetResource	UIWidgetResource	view
AccountUpdateResourceType	AccountUpdateResourceType	AccountUpdateResourceType	update
TradeWidgetType	TradeWidgetType	TradeWidgetType	trade, view
MutualFundsAssetClsType	MutualFundsAssetClsType	MutualFundsAssetClsType	view

4.3.4 Searching Application Roles

To search Application Roles, proceed as follows:

1. Select from the following methods to display the Search Role Catalog page.

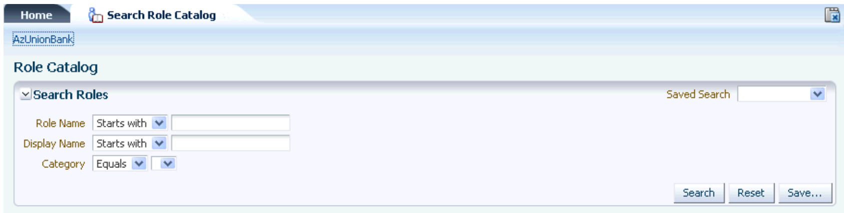
   - In the Navigation Panel, expand Applications and the named Application node applicable to the search, and double-click Role Catalog.

     Alternately, right-click Role Catalog and select Open.

   - In the Home area, select the Application Name and click Search from Application Roles.

   The Search Role Catalog tab is displayed as in Figure 4–5.

Figure 4–5 Searching for Application Roles in a Role Catalog



2. Enter the following query parameters:

   - **Role Name:** Select an operator from the list and enter a string to match.
   - **Display Name:** Select an operator from the list and enter a string to match.
   - **Category**: Select a Role Category from the list. (Oracle Entitlements Server only supports an *equals* search for Role Category.)

   Optionally, select from the Saved Search drop-down list of previously saved searches. Its query parameters automatically populate the search fields. Select Personalize... to set options for previously saved searches.

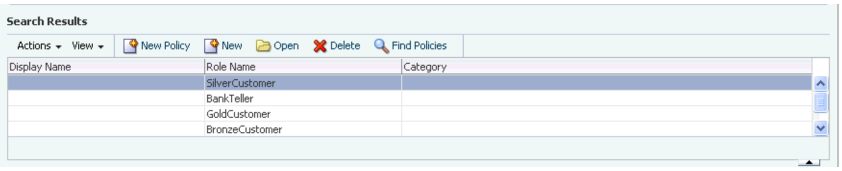
3. Optionally, click Save... to name the current query parameters.

   The named search is added to the Saved Search list.

4. Click Search.

   All results matching the query specifications are displayed in the Search Results table as in Figure 4–6.

Figure 4–6 Application Role Search Results



Display Name	Role Name	Category
	SilverCustomer	
	BankTeller	
	GoldCustomer	
	BronzeCustomer	

4.3.5 Searching Role Mapping Policies

1. Select from the following methods to display the Search Role Mapping Policies page:

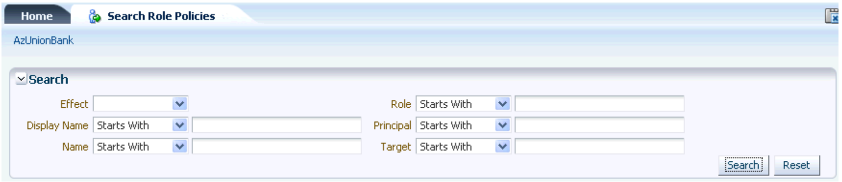
   - In the Navigation Panel, expand Applications and the named Application node applicable to the search, and double-click Role Mapping Policies.

     Alternately, right-click Role Mapping Policies and select Open.

- In the Home area, select the Application Name and click Search from Role Mapping Policies.

The Search Role Policies page is displayed as in Figure 4–7.

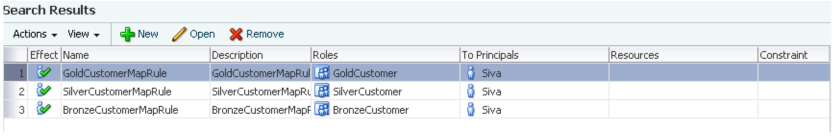
Figure 4–7 Searching for Role Mapping Policies



2. In the Search section, enter the query parameters as follows:
   - **Effect**: Select the policy effect (Grant/Deny) from the list.
   - **Display Name:** Select an operator from the list and enter a string to match.
   - **Name**: Select an operator from the list and enter a string to match.
   - **Role**: Select an operator from the list and enter a string to match.
   - **Principal**: Select an operator from the list and enter a string to match.
   - **Target**: Select an operator from the list and enter a string to match.
3. Click Search.

   All results matching the query specifications are displayed in the Search Results table as in Figure 4–8.

Figure 4–8 Role Mapping Policy Search Results



	Effect	Name	Description	Roles	To Principals	Resources	Constraint
1	✓	GoldCustomerMapRule	GoldCustomerMapRul	GoldCustomer	Siva		
2	✓	SilverCustomerMapRule	SilverCustomerMapRu	SilverCustomer	Siva		
3	✓	BronzeCustomerMapRule	BronzeCustomerMapF	BronzeCustomer	Siva		

4.3.6 Searching Resources

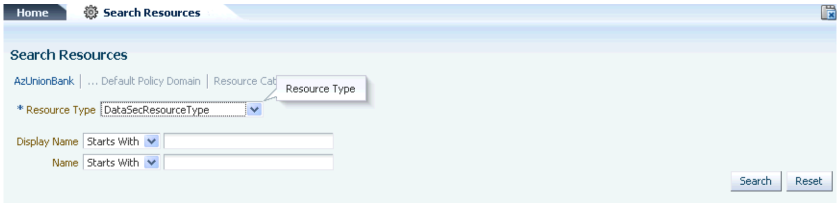
A Resource can be hierarchical (a scenario in which the sub resource inherits attributes from the parent resource) or non-hierarchical. If a Resource is hierarchical, its tiered-organization is shown in the Search results. To search Resources, proceed as follows:

1. Select from the following methods to display the Search Role Mapping Policies page:
   - In the Navigation Panel, expand Applications and the named Application node applicable to the search. Expand the appropriate Policy Domain and Resource Catalog and double-click Resources.

     Alternately, right-click Resources and select Open.
   - In the Home area, select the Application Name and click Search from Resources.

   The Search Resources page is displayed as in Figure 4–9.

Figure 4–9 Searching for Resources



* Resource Type DataSecResourceType

2. Enter the following query parameters:
   - **Resource Type**: Select a resource type from the list. This parameter is required.
   - **Display Name:** Select an operator from the list and enter a string to match.
   - **Name**: Select an operator from the list and enter a string to match.
3. Click Search.

   All results matching the query specifications are displayed in the Search Results table.

4.3.7 Searching Entitlements

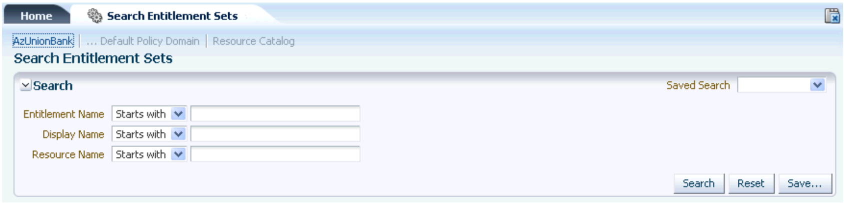
To search Entitlements, proceed as follows:

1. Select from the following methods to display the Search Entitlements page:
   - In the Navigation Panel, expand Applications and the named Application node applicable to the search. Expand the appropriate Policy Domain and Resource Catalog and double-click Entitlements.

     Alternately, right-click Entitlements and select Open.
   - In the Home area, select the Application Name and click Search from Entitlements. (In this case, the search is done only within the Default Policy Domain.)

   The Search Entitlements tab is displayed in the Home area as in Figure 4–10.

Figure 4–10 Searching for Entitlements



2. Enter the following query parameters:
   - **Entitlement Name:** Select an operator from the list and enter a string to match.
   - **Display Name:** Select an operator from the list and enter a string to match.
   - **Resource name**: Select an operator from the list and enter a string to match.

Optionally, select from the Saved Search drop-down list of previously saved searches. Its query parameters automatically populate the search fields. Select Personalize... to set options for previously saved searches.

3. Optionally, click Save... to name the current query parameters.

   The name search is added to the Saved Search list.

4. Click Search.

   All results matching the query specifications are displayed in the Search Results table.

4.3.8 Searching Authorization Policies

Authorization Policies can be searched by specifying a policy name, a principal, or a target. To search Authorization Policies, proceed as follows:

1. Select from the following methods to display the Search Policies page:

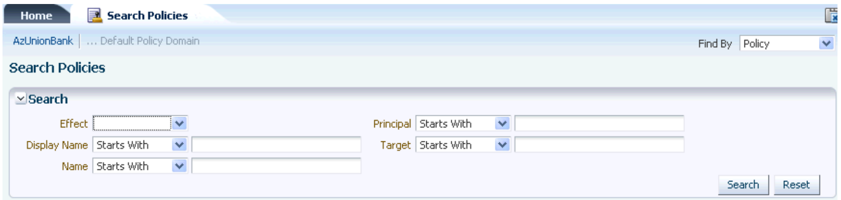
   - In the Navigation Panel, expand Applications and the named Application node applicable to the search. Expand the appropriate Policy Domain and Resource Catalog and double-click Authorization Policies.

     Alternately, right-click Authorization Policies and select Open.

   - In the Home area, select the Application Name, and click Search under Authorization Policies. (In this case, the search is done within the Default Policy Domain.)

   The Search Policies tab is displayed in Figure 4–11.

Figure 4–11 Searching for Policies



Find By: Policy

Effect:

Display Name: Starts With

Name: Starts With

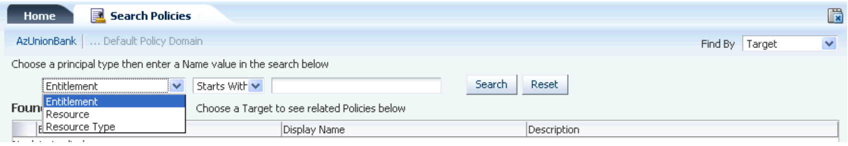
Principal: Starts With

Target: Starts With

2. Select the search type from the Find By list.

   The query parameters change according to the selection. Options include Policy, Principal or Target. Figure 4–11 is a screenshot in which Policy is selected. Figure 4–12 is a screenshot in which Target is selected.

Figure 4–12 Searching Polices by Target



Find By: Target

Choose a principal type then enter a Name value in the search below

Entitlement · Starts With

Choose a Target to see related Policies below

3. Search using the option based on your previous selection.

   - To Find By: Policy, enter the following query parameters.

     – **Effect:** Select the policy effect (Grant/Deny) from the list.

  - **Display Name:** Select an operator from the list and enter a string to match.
  - **Name:** Select an operator from the list and enter a string to match.
  - **Principal:** Select an operator from the list and enter a string to match.
  - **Target:** Select an operator from the list and enter a string to match.
- To Find By: Principal or Find By: Target, select an operator from the list, and enter a string to match.

  A Resource Type must be provided if the Resource or Resource Type operator is selected.

4. Click Search.

4.3.9 Searching Attributes

To search Attributes, proceed as follows:

1. In the Navigation Panel, expand Applications and the named Application node applicable to the search.
2. Expand Extensions and double-click Attributes to display the Search Attributes page.

   Alternately, right-click Attributes and select Open.
3. Enter the following query parameters.
   - **Display Name**: Select an operator from the list and enter a string to match.
   - **Name**: Select an operator from the list and enter a string to match.
   - **Type**: Select an operator from the list and enter a string to match.

   Optionally, select from the Saved Search drop-down list of previously saved searches. Its query parameters automatically populate the search fields. Select Personalize... to set options for previously saved searches.
4. Optionally, click Save... to name the current query parameters.

   The named search is added to the Saved Search list.
5. Click Search.

4.3.10 Searching Functions

To search application functions, proceed as follows

1. In the Navigation Panel, expand Applications and the named Application node applicable to the search.
2. Expand Extensions and double-click Functions to display the Search Functions page.

   Alternately, right-click Functions and select Open.
3. Enter the following query parameters.
   - **Name**: Select an operator from the list and enter a string to match.
   - **Display Name**: Select an operator from the list and enter a string to match.

   Optionally, select from the Saved Search drop-down list of previously saved searches. Its query parameters automatically populate the search fields. Select Personalize... to set options for previously saved searches.

4. Optionally, click Save... to name the current query parameters.

   The named search is added to the Saved Search list.

5. Click Search.

4.3.11 Searching for Users Globally

To search for Users, proceed as follows:

1. Select from the following methods to display the Search External Roles page:
   - In the Navigation Panel, expand Global and double-click Users.

     Alternately, right-click Users and select Open.
   - In the Home area, click Search - Users from the Search and Create section.
2. Enter the following query parameters:
   - **User Name**: Select an operator from the list and enter a string to match.
   - **Display Name**: Select an operator from the list and enter a string to match.

   Optionally, select from the Saved Search drop-down list of previously saved searches. Its query parameters automatically populate the search fields. Select Personalize... to set options for previously saved searches.
3. Optionally, click Save... to name the current query parameters.

   The named search is added to the Saved Search list.
4. Click Search.

   The results are displayed in Search Results.

5

Configuring Predefined Attribute Retrievers

The Policy Information Point (PIP) is an information store that acts as a source for attribute values. Oracle Entitlements Server relies on an Attribute Retriever plug-in to get attribute values from one or more of these PIP information stores. Predefined Attribute Retrievers are shipped with Oracle Entitlements Server. This chapter documents these predefined Attribute Retrievers and related configuration requirements. It contains the following sections.

- Understanding Predefined Attribute Retrievers
- Configuring the Predefined Attribute Retrievers
- Modifying jps-config.xml
- Setting Up PIP Connection Credentials

5.1 Understanding Predefined Attribute Retrievers

Oracle Entitlements Server contains predefined Attribute Retrievers that are used to connect to, and retrieve attribute values from, Lightweight Directory Access Protocol (LDAP) data stores and relational database management systems (RDBMS). These plug-ins can handle one or more attributes defined in the system without additional programming. They also contain a caching feature and failover.

- An in-memory cache mechanism is used to improve performance by reducing communications between Oracle Entitlements Server and the external repository. The cache holds up to 1000 entries and can be enabled for each individual attribute. The cache size is not configurable. If the limit is reached, cache items are removed randomly. Example 5–2 illustrates the definition of an individual attribute with the `cached` and `ttl` properties.
- Repository failover can also be configured. When a call for an attribute is received, Oracle Entitlements Server checks whether the primary repository is active. If it is active, the value is retrieved. If the primary repository is not active, it has failed previously and the backup repository is active. In the latter case, Oracle Entitlements Server checks to see if it is time to switch back to the active repository (based on configuration). If it is time to switch back, the switch is made and the value is retrieved from the primary repository. If the configured time has not yet passed, the value is retrieved from the active backup repository.

Note: If errors occur when retrieving values from the primary repository, Oracle Entitlements Server searches the backup repositories, trying them one by one until an active one is found.

See Section 5.2.3, "Configuring Individual Attributes for Predefined Attribute Retrievers" for configuration information.

5.2 Configuring the Predefined Attribute Retrievers

Configuration information for these Attribute Retrievers is defined in the jps-config.xml configuration file. You must configure two types of information: attribute query information and repository connection information

- Repository connection information is used to connect to the data store and may include its location, JDBC driver and URL or LDAP URL (whichever is applicable) and the user/credential information. This connection information is related to a particular retriever instance. Repository connection information is defined in the <serviceInstances> section of jps-config.xml as illustrated in Example 5–1.

Example 5–1 Repository Connection Information Defined for Attribute Retriever

```
<serviceInstance name="policystore.rdbms" provider="policy.rdbms">
 <property name="jdbc.url"
  value="jdbc:oracle:thin:@hostname.example.com:1521:orcl"/>
 <property name="jdbc.driver" value="oracle.jdbc.driver.OracleDriver"/>
 <property name="bootstrap.security.principal.key" value="keyname"/>
 <property name="bootstrap.security.principal.map" value="mapname"/>
 <property name="oracle.security.jps.ldap.root.name" value="cn=jpsTestNode"/>
 <property name="oracle.security.jps.farm.name"
  value="cn=wcai_view_jing.atzsrg"/>
</serviceInstance>
```

  Section 5.2.1, "Configuring the LDAP Repository Attribute Retriever Parameters," Section 5.2.2, "Configuring the Database Repository Attribute Retriever Parameters," and Section 5.3, "Modifying jps-config.xml" contain information regarding a repository connection configuration.

  > **Note:** The instance must also be defined in the default <jpsContexts> section. See Example 5–8, "Declaring the Predefined Attribute Retriever in jpsContext".

- Attribute query information is related to a particular attribute and includes its name, the name of the predefined Attribute Retriever used, the search query for retrieval (for example, a SQL query if the store is a relational database or an LDAP query if it's a directory), and any attribute caching information. Attribute query information is defined in the <propertySets> section of jps-config.xml as illustrated in Example 5–2.

Example 5–2 Attribute Query Information Defined for Attribute Retriever

```
<propertySet name="ootb.pip.attribute.age.based.on.myattr.ldap">
    <property name="ootb.pip.attr.type" value="OOTB_PIP_ATTRIBUTE"/>
    <property name="ootb.pip.ref" value="pip.service.ootb.ldap"/>
    <property name="name" value="oespipage_myattr"/>
    <property name="query" value="(cn=%MyAttr%)"/>
    <property name="cached" value="true"/>
    <property name="ttl" value="60"/>
</propertySet>
```

Section 5.2.3, "Configuring Individual Attributes for Predefined Attribute Retrievers" and Section 5.3, "Modifying jps-config.xml" contain information regarding an attribute query configuration.

> **Note:** These predefined Attribute Retrievers can be configured with Oracle Database 11gR1, Oracle Internet Directory 11gR1, and Oracle Virtual Directory 11gR1.

The following sections contain information on the configuration parameters for each type of Attribute Retriever. As previously mentioned, these parameters are in the `jps-config.xml`, the configuration file (used by Java EE containers) located in the *$DOMAIN_HOME*`/config/fmwconfig` directory.

- Section 5.2.1, "Configuring the LDAP Repository Attribute Retriever Parameters"
- Section 5.2.2, "Configuring the Database Repository Attribute Retriever Parameters"
- Section 5.2.3, "Configuring Individual Attributes for Predefined Attribute Retrievers"

5.2.1 Configuring the LDAP Repository Attribute Retriever Parameters

Table 5–1 documents the parameters that must be defined when using the LDAP Attribute Retriever. See Example 5–5, "Using the Predefined LDAP Attribute Retriever" and Example 5–10, "Configuring LDAP Failover" for sample configuration code.

Table 5–1 LDAP Attribute Retriever Parameters

Name	Usage
name	**Description:** The predefined Attribute Retriever’s name **Mandatory** **Accepted Value:** String defining the Attribute Retriever service instance.
description	**Description:** A description of the predefined Attribute Retriever **Optional** **Accepted Value:** string
type	**Description:** The predefined Attribute Retriever’s type **Mandatory** **Accepted Value:** LDAP_PIP
failed.server.retry.interval	**Description:** After communication with a primary repository has failed, this attribute defines the interval of time during which the backup repository is used before switching back to the primary repository. **Optional** **Accepted Value:** Takes a value equal to the number of seconds. Default value is 15.

Table 5–1 (Cont.) LDAP Attribute Retriever Parameters

Name	Usage
bootstrap.security.principal.key	**Description:** Defines the key for the password credentials to access the LDAP policy store, stored in the CSF store. Valid in JEE and JSE applications. Applies to LDAP and database stores. See Section 5.4, "Setting Up PIP Connection Credentials." **Optional:** For production mode only. **Accepted Value:** key name of the credential; for example, `oes_sm_key`.
bootstrap.security.principal.map	**Description:** Defines the map for the password credentials to access the LDAP policy store, stored in the CSF store. Valid in JEE and JSE applications. Applies to LDAP and database stores. See Section 5.4, "Setting Up PIP Connection Credentials." **Optional:** For production mode only. **Accepted Value:** map name of the credential; for example, `oes_sm_map`.
ldap.url	**Description:** Defines the URL of the LDAP policy store. Valid in JEE and JSE applications and only applies to LDAP stores. **Mandatory** **Accepted Value:** URI of the LDAP policy store in the format `ldap://host:port`.

5.2.2 Configuring the Database Repository Attribute Retriever Parameters

Table 5–2 documents the parameters that must be defined when using the RDBMS Attribute Retriever. See Example 5–6, "Using the Predefined RDBMS Attribute Retriever with JDBC" and Example 5–7, "Using the Predefined RDBMS Attribute Retriever with SQL" for sample configuration code.

Table 5–2 RDBMS Attribute Retriever Parameters

Name	Usage
name	**Description:** The predefined Attribute Retriever's name **Mandatory** **Accepted Value:** String defining the Attribute Retriever service instance.
description	**Description:** A description of the predefined Attribute Retriever **Optional** **Accepted Value:** string
type	**Description:** The predefined Attribute Retriever's type **Mandatory** **Accepted Value:** RDBMS_PIP

Table 5–2 (Cont.) RDBMS Attribute Retriever Parameters

Name	Usage
failed.server.retry.interval	**Description:** After the primary repository has failed, this attribute identifies the interval of time during which the backup repository is used before switching back to the primary repository. **Optional** **Accepted Value:** Takes a value equal to the number of seconds. Default value is 15.
bootstrap.security.principal.key	**Description:** Defines the key for the password credentials to access the database, stored in the CSF store. Valid in JEE and JSE applications. See Section 5.4, "Setting Up PIP Connection Credentials." **Optional:** For production mode only. **Accepted Value:** key name of the credential; for example, `oes_sm_key`.
bootstrap.security.principal.map	**Description:** Defines the map for the password credentials to access the database, stored in the CSF store. Valid in JEE and JSE applications. See Section 5.4, "Setting Up PIP Connection Credentials." **Optional:** For production mode only. **Accepted Value:** map name of the credential; for example, `oes_sm_map`.
jdbc.driver	**Description:** Location of the driver when using Java Database Connectivity (JDBC) API to connect to a database. **Mandatory:** When using JDBC API to connect to database. **Accepted Value:** `oracle.jdbc.driver.OracleDriver`, for example
jdbc.url	**Description:** Takes a URL that points to the database. **Mandatory:** When using JDBC API to connect to database. **Accepted Value:** A list of comma-delimited URLs. The first is treated as primary and so on. For example, `jdbc:oracle:thin:@hostname.example.com:1521:orcl`
datasource.jndi.name	**Description:** Data source JNDI name if you want the PIP instance working through data source rather than directly through JDBC. The data source scenario is supported on WebLogic Server and WebSphere Application Server only. **Mandatory:** If you want the PIP instance working through data source rather than directly through JDBC. **Accepted Value:** JNDI name of pre-defined data source object

5.2.3 Configuring Individual Attributes for Predefined Attribute Retrievers

Table 5–3 documents the parameters to be defined for each attribute retrieved by the configured Attribute Retriever. See Example 5–9, "Enabling an Attribute’s Cache" for a sample configuration.

Table 5–3 Configure Attributes to be Retrieved

Name	Usage
name	**Description:** The name of the attribute as defined in the policy store. When using the LDAP predefined Attribute Retriever, the attribute name defined for Oracle Entitlements Server must be the same as the attribute name defined in the LDAP store. Currently, there is no name mapping functionality. **Mandatory** **Accepted Value:** Attribute name
query	**Description:** The SQL command or LDAP filter used for the query. Users can use a built-in and custom attributes in the query string. For example, the built-in attribute `sys_user` can be used to define a query such as *select age from customers where name=%sys_user%;*. The token is automatically replaced by its value before sending the query to the data store. Bi-directional dependency (where, for example, AttributeA's query string contains AttributeB and AttributeB's query string contains AttributeA) can also be detected and, in such cases, an exception is thrown. **Mandatory** **Accepted Value:** SQL command or LDAP filter.
search.base	**Description:** The LDAP search base. **Mandatory:** For LDAP only. **Accepted Value:** The DN of the search base object.
ttl	**Description:** The time-to-live in seconds of any cached attribute values when cached is enabled. **Optional** **Accepted Value:** Any integer; default value is 60 seconds if cache is enabled.
cached	**Description:** Enables the caching of attribute values. **Optional** **Accepted Value:** Default value is false.
ootb.pip.attr.type	**Description:** Should be set to OOTB_PIP_ATTRIBUTE. **Mandatory** **Accepted Value:** OOTB_PIP_ATTRIBUTE.
ootb.pip.ref	**Description:** Should be set to an OOTB PIP instance. **Mandatory** **Accepted Value:** The PIP service instance name defined in the `<serviceInstance>` section of `jps-config.xml`

5.3 Modifying jps-config.xml

To configure the predefined Attribute Retriever in `jps-config.xml`, modify the elements as described in each example in this section. Example 5–3 is a sample `jps-config.xml` file. The examples following it illustrate the modifications that can be made.

Example 5–3 Sample jps-config.xml File

```
<?xml version="1.0"?>

<jpsConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="
```

```
http://xmlns.oracle.com/oracleas/schema/jps-config-11_0.xsd">

   <property name="oracle.security.jps.jaas.mode" value="off"/>
   <property name="oracle.security.jps.enterprise.user.class"
    value="weblogic.security.principal.WLSUserImpl"/>
   <property name="oracle.security.jps.enterprise.role.class"
    value="weblogic.security.principal.WLSGroupImpl"/>

<propertySets>
<!-- These are the global authenticated role properties -->
  <propertySet name="authenticated.role.properties">
     <property name="authenticated.role.name" value="authenticated-role"/>
     <property name="authenticated.role.uniquename" value="authenticated-role"/>
     <property name="authenticated.role.description"
        value="This is the authenticated role used by identity store
        service instance."/>
  </propertySet>

<!-- attribute defined for ldap retriever -->
  <propertySet name="ootb.pip.attribute.age.ldap">
     <property name="ootb.pip.attr.type" value="OOTB_PIP_ATTRIBUTE"/>
     <property name="ootb.pip.ref" value="pip.service.ootb.ldap"/>
     <property name="name" value="oespipage"/>
     <property name="query" value="(cn=%SYS_USER%)"/>
     <property name="cached" value="true"/>
     <property name="ttl" value="60"/>
  </propertySet>

  <propertySet name="ootb.pip.attribute.age.based.on.myattr.ldap">
     <property name="ootb.pip.attr.type" value="OOTB_PIP_ATTRIBUTE"/>
     <property name="ootb.pip.ref" value="pip.service.ootb.ldap"/>
     <property name="name" value="oespipage_myattr"/>
     <property name="query" value="(cn=%MyAttr%)"/>
     <property name="cached" value="true"/>
     <property name="ttl" value="60"/>
  </propertySet>

  <propertySet name="ootb.pip.attribute.gender.ldap">
     <property name="ootb.pip.attr.type" value="OOTB_PIP_ATTRIBUTE"/>
     <property name="ootb.pip.ref" value="pip.service.ootb.ldap"/>
     <property name="name" value="oespipgender"/>
     <property name="query" value="(oespipage=%oespipage%)"/>
     <property name="cached" value="true"/>
     <property name="ttl" value="60"/>
  </propertySet>

<!-- attribute defined for rdbms retriever -->
  <propertySet name="ootb.pip.attribute.age.rdbms">
     <property name="ootb.pip.attr.type" value="OOTB_PIP_ATTRIBUTE"/>
     <property name="ootb.pip.ref" value="pip.service.ootb.db"/>
     <property name="name" value="oespipage"/>
     <property name="query" value="select oespipage
         from pip_info_store where username=%SYS_USER%"/>
     <property name="cached" value="true"/>
     <property name="ttl" value="60"/>
  </propertySet>

  <propertySet name="ootb.pip.attribute.age.based.on.myattr.rdbms">
     <property name="ootb.pip.attr.type" value="OOTB_PIP_ATTRIBUTE"/>
     <property name="ootb.pip.ref" value="pip.service.ootb.db"/>
```

```
      <property name="name" value="oespipage_myattr"/>
      <property name="query" value="select oespipage
          as oespipage_myattr from pip_info_store where username=%MyAttr%"/>
      <property name="cached" value="true"/>
      <property name="ttl" value="60"/>
   </propertySet>

   <propertySet name="ootb.pip.attribute.gender.rdbms">
      <property name="ootb.pip.attr.type" value="OOTB_PIP_ATTRIBUTE"/>
      <property name="ootb.pip.ref" value="pip.service.ootb.db"/>
      <property name="name" value="oespipgender"/>
      <property name="query" value="select oespipgender
           from pip_info_store where oespipage=%oespipage%"/>
       <property name="cached" value="true"/>
      <property name="ttl" value="60"/>
   </propertySet>
 </propertySets>

 <serviceProviders>

   <serviceProvider type="CREDENTIAL_STORE" name="credstoressp"
        class="oracle.security.jps.internal.credstore.ssp.
               SspCredentialStoreProvider">
      <description>SecretStore-based CSF Provider</description>
   </serviceProvider>

   <serviceProvider class="oracle.security.jps.az.
         internal.runtime.provider.PIPServiceProvider"
         name="pip.service.provider" type="PIP"/>

   <serviceProvider type="POLICY_STORE" name="policy.rdbms"
         class="oracle.security.jps.internal.policystore.
         OPSSPolicyStoreProvider">
      <property name="policystore.type" value="DB_ORACLE"/>
      <description>DBMS based PolicyStore</description>
   </serviceProvider>

   <serviceProvider name="pdp.service.provider" type="PDP"
          class="oracle.security.jps.az.internal.
          runtime.provider.PDPServiceProvider">
       <description>OPSS Runtime PDP Service Provider</description>
   </serviceProvider>

   <serviceProvider name="idstore.xml.provider" type="IDENTITY_STORE"
          class="oracle.security.jps.internal.idstore.
          xml.XmlIdentityStoreProvider">
        <description>XML-based IdStore Provider</description>
   </serviceProvider>

   <serviceProvider name="jaas.login.provider" type="LOGIN"
          class="oracle.security.jps.internal.
          login.jaas.JaasLoginServiceProvider">
        <description>This is Jaas Login Service Provider and is used
          to configure login module service instances</description>
   </serviceProvider>

   <serviceProvider name="policy.xml" type="POLICY_STORE"
         class="oracle.security.jps.internal.
         policystore.xml.XmlPolicyStoreProvider">
      <description>XML-based PolicyStore</description>
```

```
</serviceProvider>

<serviceProvider type="POLICY_STORE" name="policy.oid"
         class="oracle.security.jps.internal.
         policystore.ldap.LdapPolicyStoreProvider">
     <description>LDAP-based PolicyStore</description>
     <property name="policystore.type" value="OID"/>
     <property name="connection.pool.maxsize" value="30"/>
     <property name="connection.pool.provider.type" value="idmpool"/>
  </serviceProvider>

  <serviceProvider type="AUDIT" name="audit.provider"
        class="oracle.security.jps.internal.audit.AuditProvider">
       <description>Audit Service</description>
  </serviceProvider>
</serviceProviders>

<serviceInstances>

  <serviceInstance name="credstore" provider="credstoressp" location="./">
          <description>File Based Credential Store Service Instance</description>
  </serviceInstance>

  <serviceInstance name="idstore.xml" provider="idstore.xml.provider">
<!-- Subscriber name must be defined for XML Identity Store -->
     <property name="subscriber.name" value="jazn.com"/>
<!-- This is the location of XML Identity Store -->
     <property name="location" value="./user-data.xml"/>
<!-- This property set defines the authenticated role -->
     <propertySetRef ref="authenticated.role.properties"/>
  </serviceInstance>
  <serviceInstance name="idstore.loginmodule"
          provider="jaas.login.provider">
          <description>Identity Store Login Module</description>
     <property name="loginModuleClassName" value="oracle.security.jps.internal.
            jaas.module.idstore.IdStoreLoginModule"/>
     <property name="jaas.login.controlFlag" value="REQUIRED"/>
     <property name="debug" value="true"/>
     <property name="addAllRoles" value="true"/>
  </serviceInstance>

  <serviceInstance name="policystore.rdbms" provider="policy.rdbms">
        <property name="jdbc.url"
           value="jdbc:oracle:thin:@hostname.example.com:1521:orcl"/>
        <property name="jdbc.driver" value="oracle.jdbc.driver.OracleDriver"/>
        <property name="bootstrap.security.principal.key" value="keyname"/>
        <property name="bootstrap.security.principal.map" value="mapname"/>
        <property name="oracle.security.jps.ldap.root.name"
           value="cn=jpsTestNode"/>
        <property name="oracle.security.jps.farm.name"
           value="cn=wcai_view_jing.atzsrg"/>
  </serviceInstance>

  <serviceInstance name="policystore.rdbms.ds" provider="policy.rdbms">
     <property name="oracle.security.jps.ldap.root.name"
         value="cn=jpsTestNode"/>
     <property name="oracle.security.jps.farm.name"
         value="cn=wcai_view_jing.atzsrg"/>
     <property value="atzsrgds" name="datasource.jndi.name"/>
  </serviceInstance>
  <serviceInstance name="pdp.service" provider="pdp.service.provider">
     <property name="oracle.security.jps.runtime.pd.client.sm_name"
           value="${atzsrg.pdp.configuration_id}"/>
     <property name="oracle.security.jps.pdp.
           AuthorizationDecisionCacheEnabled" value="true"/>
     <property name="oracle.security.jps.pdp.
           AuthorizationDecisionCacheEvictionCapacity" value="500"/>
     <property name="oracle.security.jps.pdp.
           AuthorizationDecisionCacheEvictionPercentage" value="10"/>
     <property name="oracle.security.jps.pdp.
           AuthorizationDecisionCacheTTL" value="60"/>
     <property name="oracle.security.jps.ldap.
           policystore.refresh.interval" value="30000"/>
     <property name="oracle.security.jps.policystore.
           refresh.purge.timeout" value="600000"/> <!-- 10 minutes -->
     <property name="loading_attribute_backward_compatible" value="false"/>
<!-- Properties for controlled mode PD -->
     <property name="oracle.security.jps.runtime.
           pd.client.policyDistributionMode" value="non-controlled"/>
     <property name="oracle.security.jps.runtime.
           instance.name" value="${atzsrg.pdp.instance_name}"/>
  </serviceInstance>

  <serviceInstance name="policystore.oid" provider="policy.oid">
    <property name="max.search.filter.length" value="4096"/>
    <property name="bootstrap.security.principal.key" value="keyname"/>
    <property name="bootstrap.security.principal.map" value="mapname"/>
    <property name="ldap.url" value="ldap://hostname.example.com:3060"/>
    <property name="oracle.security.jps.ldap.root.name"
       value="cn=jpsTestNode"/>
    <property name="oracle.security.jps.farm.name"
       value="cn=wcai_view_jing.atzsrg"/>
    <property name="oracle.security.jps.policystore.resourcetypeenforcementmode"
       value="Lenient"/>
  </serviceInstance>

  <serviceInstance name="policystore.xml" provider="policy.xml"
            location="./system-jazn-data.xml"/>

  <serviceInstance name="user.authentication.loginmodule"
            provider="jaas.login.provider">
         <description>User Authentication Login Module</description>
         <property name="loginModuleClassName"
              value="oracle.security.jps.internal.
              jaas.module.authentication.JpsUserAuthenticationLoginModule"/>
         <property name="jaas.login.controlFlag" value="REQUIRED"/>
  </serviceInstance>

  <serviceInstance name="user.assertion.loginmodule"
            provider="jaas.login.provider">
         <description>User Assertion Login Module</description>
         <property name="loginModuleClassName"
              value="oracle.security.jps.internal.
              jaas.module.assertion.JpsUserAssertionLoginModule"/>
         <property name="jaas.login.controlFlag" value="REQUIRED"/>
  </serviceInstance>

  <serviceInstance name="pip.service.ootb.ldap" provider="pip.service.provider">
         <property name="type" value="LDAP_PIP"/>
```

```
        <property name="ldap.url" value="ldap://hostname.example.com:3060"/>
        <property name="bootstrap.security.principal.key" value="keyname"/>
        <property name="bootstrap.security.principal.map" value="mapname"/>
        <property name="search.base" value="cn=pip_info_store,
             cn=wcai_view_jing.atzsrg,cn=JPSContext,cn=jpsTestNode"/>
        <property name="failed.server.retry.interval" value="10"/>
 </serviceInstance>
<!-- JPS Audit Service Instance-->
 <serviceInstance name="audit" provider="audit.provider">
        <property name="audit.filterPreset" value="None"/>
        <property name="audit.maxDirSize" value ="0"/>
        <property name="audit.maxFileSize" value ="104857600"/>
        <property name="audit.loader.jndi" value="jdbc/AuditDB"/>
        <property name="audit.loader.interval" value="15" />
        <property name="audit.loader.repositoryType" value="File" />
 </serviceInstance>

 <serviceInstance name="pip.service.ootb.db" provider="pip.service.provider">
        <property name="type" value="RDBMS_PIP"/>
        <property name="jdbc.url"
             value="jdbc:oracle:thin:@hostname.example.com:1521:orcl"/>
        <property name="jdbc.driver" value="oracle.jdbc.driver.OracleDriver"/>
        <property name="bootstrap.security.principal.key" value="keyname"/>
        <property name="bootstrap.security.principal.map" value="mapname"/>
        <property name="failed.server.retry.interval" value="10"/>
 </serviceInstance>

 <serviceInstance name="pip.service.ootb.db.ds" provider="pip.service.provider">
        <property name="type" value="RDBMS_PIP"/>
        <property value="atzsrgds" name="datasource.jndi.name"/>
        <property name="failed.server.retry.interval" value="10"/>
 </serviceInstance>

</serviceInstances>

    <jpsContexts default="default">
        <jpsContext name="default">
            <serviceInstanceRef ref="policystore.oid"/>
            <serviceInstanceRef ref="pdp.service"/>
            <serviceInstanceRef ref="audit"/>
            <serviceInstanceRef ref="idstore.xml"/>
            <serviceInstanceRef ref="idstore.loginmodule"/>
            <serviceInstanceRef ref="pip.service.ootb.ldap"/>
            <serviceInstanceRef ref="pip.service.ootb.db"/>
        </jpsContext>
        <jpsContext name="smsec">
            <serviceInstanceRef ref="credstore"/>
        </jpsContext>

    </jpsContexts>

</jpsConfig>
```

Example 5–4 illustrates how the `serviceProvider` element defines the use of a predefined Attribute Retriever by defining the internal Oracle Entitlements Server class.

Example 5–4 Declaring the Predefined Attribute Retriever

```
<serviceProvider
```

```
  class="oracle.security.jps.az.internal.runtime.provider.PIPServiceProvider"
  name="pip.service.provider" type="PIP"/>
```

The following examples illustrate how to modify the `serviceInstance` element for the predefined Attribute Retriever being used.

- Example 5–5, "Using the Predefined LDAP Attribute Retriever"
- Example 5–6, "Using the Predefined RDBMS Attribute Retriever with JDBC"
- Example 5–7, "Using the Predefined RDBMS Attribute Retriever with SQL"
- Example 5–8, "Declaring the Predefined Attribute Retriever in jpsContext"
- Example 5–9, "Enabling an Attribute's Cache"
- Example 5–10, "Configuring LDAP Failover"

Example 5–5 illustrates how to modify the `serviceInstance` element when using the predefined LDAP Attribute Retriever.

Example 5–5 Using the Predefined LDAP Attribute Retriever

```
<serviceInstance name="pip.service.ootb.ldap" provider="pip.service.provider">
   <property name="type" value="RDBMS_PIP"/>
   <property name="ldap.url" value="ldap://hostname.example.com:3080"/>
   <property name="bootstrap.security.principal.key" value="keyname"/>
   <property name="bootstrap.security.principal.map" value="mapname"/>
   <property name="failed.server.retry.interval" value="10"/>
</serviceInstance>
```

The following two examples illustrate how to modify the `serviceInstance` element when using the predefined RDBMS Attribute Retriever. Example 5–6 is when using Java Database Connectivity (JDBC) API.

Example 5–6 Using the Predefined RDBMS Attribute Retriever with JDBC

```
<serviceInstance name="pip.service.ootb.db" provider="pip.service.provider">
    <property name="type" value="RDBMS_PIP"/>
    <property name="jdbc.url"
     value="jdbc:oracle:thin:@hostname.example.com:1521:orcl"/>
    <property name="jdbc.driver" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="bootstrap.security.principal.map" value="mapname"/>
    <property name="bootstrap.security.principal.key" value="keyname"/>
    <property name="failed.server.retry.interval" value="10"/>
</serviceInstance>
```

Example 5–7 is when using a SQL database.

Example 5–7 Using the Predefined RDBMS Attribute Retriever with SQL

```
<serviceInstance name="pip.service.ootb.db" provider="pip.service.provider">
    <property name="type" value="RDBMS_PIP"/>
    <property name="datasource.jndi.name" value="DB_RAC"/>
    <property name="failed.server.retry.interval" value="10"/>
</serviceInstance>
```

Example 5–8 illustrates how to declare the predefined Attribute Retriever reference in the `jpsContext` element. This sample defines a predefined RDBMS Attribute Retriever.

Example 5–8 Declaring the Predefined Attribute Retriever in jpsContext

```
  <jpsContext name="default">
```

```
        <serviceInstanceRef ref="policystore.db"/>
        <serviceInstanceRef ref="pdp.service"/>
        <serviceInstanceRef ref="audit"/>
        <serviceInstanceRef ref="idstore.xml"/>
        <serviceInstanceRef ref="idstore.loginmodule"/>
        <serviceInstanceRef ref="pip.service.ootb.db"/>
</jpsContext>
```

Example 5–9 illustrates how to configure the caching of a specific attribute value. Caching is enabled per attribute. In this example, the cache record is deleted after 60 seconds.

Example 5–9 Enabling an Attribute's Cache

```
<propertySet name="ootb.pip.attribute.gender.ldap">
        <property name="ootb.pip.attr.type" value="OOTB_PIP_ATTRIBUTE"/>
        <property name="ootb.pip.ref" value="pip.service.ootb.ldap"/>
        <property name="name" value="oespipgender"/>
        <property name="query" value="(oespipage=%oespipage%)"/>
        <property name="cached" value="true"/>
        <property name="ttl" value="60"/>
</propertySet>
```

Example 5–10 illustrates how to configure the failover behavior. In this example, the primary connection is `ldap://dadvmg0065:3080` and the backup connection is `ldap://scl58123:3060`. The failed server retry interval is 10 seconds.

Example 5–10 Configuring LDAP Failover

```
<serviceInstance name="pip.service.ootb.ldap" provider="pip.service.provider">
      <property name="type" value="LDAP_PIP"/>
      <property name="ldap.url"
         value="ldap://dadvmg0065:3080,ldap://scl58123:3060"/>
      <property name="bootstrap.security.principal.key" value="keyname"/>
      <property name="bootstrap.security.principal.map" value="mapname"/>
      <property name="failed.server.retry.interval" value="10"/>
</serviceInstance>
```

5.4 Setting Up PIP Connection Credentials

As documented in Table 5–1, " LDAP Attribute Retriever Parameters" and Table 5–2, " RDBMS Attribute Retriever Parameters", the `bootstrap.security.principal.key` and `bootstrap.security.principal.map` parameters define the key and the map (respectively) to access the data store. Oracle Entitlements Server ships with `oesPassword.sh` which sets these LDAP and database connection credentials in the bootstrap credential store. The tool is located in the *$OES_SM_INSTANCE_DIRECTORY*`/bin/` directory. Use the following command to run it.

```
./oesPassword.sh -setpass
```

It prompts for the security principal key name, the security principal map name, the username and associated password.

6 Delegating With Administrator Roles

System administrative rights and policy management permissions can be delegated from one administrator to another by creating Administrator Roles with restricted rights, or by granting an existing Administrator Role to a user. This chapter documents information on how to delegate policy and system administrative tasks. It contains the following sections:

6.1 About Delegated Administrators

Administration is when one or more authorized rights are granted to someone to do a certain job. Delegation is the ability for that someone to transfer the authorized right that has been granted them to another. In combination, we can define delegating administration as the transference of authorized rights from one to another. In Oracle Entitlements Server, administrators who are authorized to perform a task on policy objects and entities may transfer this right to others using Administration Roles. Administration Roles consist of a subject (the person to whom the role is granted), the resources (the objects to which the role pertains) and actions (view, manage/modify).

Oracle Entitlements Server allows you to define delegating Administrator Roles by assigning Administration Privileges, and mapping external roles and users, to it. When a user is logged in as an Administrator, the Navigation Panel displays only the set of Applications the logged in user is authorized to administer. In point of fact, all objects that a delegating Administrator cannot administer are hidden. Any nondefault delegating Administrator Role can perform management operations if it is granted the Admin Role with VIEW and MANAGE privileges.

> **Note:** A nondefault Administrator Role is any Administrator Role created manually. This would not include Administrator Roles automatically created when you create an Application or a Policy Domain.

The following restrictions also apply to Administrator Roles.

- Non-system level (delegating) Administration Roles can only manage other Administration Roles within its scope. For example, an Administration Role created for Application1 can manage Administration Roles in Application 1 Policy Domains but cannot manage peer Administration Roles in Application1, or any roles in Application2 and its Policy Domains. Scope and granularity are discussed further in Section 6.2, "Delegating Using Scope and Granularity."
- System level Administration Roles (as discussed in Chapter 12, "Managing System Configurations") can manage delegating Administration Roles in any Application or Policy Domain.
- Nondefault Administration Roles (again, created manually) cannot manage default Administration Roles in any Application or Policy Domain.

6.2 Delegating Using Scope and Granularity

Delegated administration is all about transferring management of resources and policy objects from one person to another. The scope of the delegation (or range of objects covered by the delegation) is defined in levels. The granularity of administration defines the type of objects managed at each scope. A default Administration Role is automatically created when each scope is created; additional Administration Roles can be created later.

Note: The following is applicable to all default Administration Roles.

- Default Administrator Roles cannot be deleted individually.
- If a Policy Domain is deleted, all Administration Roles (including the default) are deleted.
- If the Application is deleted, all Administration Roles are deleted.
- Privileges assigned to default Administrator Roles cannot be modified.

From highest to lowest, the scopes and applicable granularity are as follows:

- The top-level `SystemAdmin` has privileges to manage system-level resources as well as all policy-related objects. System resources include Administrator Roles, system configurations and Security Module bindings. Policy objects include the Application objects.

  Note: System Administrators have rights to all policy objects, including all Application objects and child Policy Domains but they are primarily intended to manage configurations, Application objects, and the bindings between the two.

  Information on managing system level Administrator Roles is in Chapter 12, "Managing System Configurations."

- Application administrators have privileges to manage all objects in the Application to which they are assigned. One `ApplicationPolicyAdmin` is generated for each Application that is created. They are primarily intended to delegate the management of policy objects within the Application (including the Policy Domains and its children, such as Functions, Attributes, Application Roles and Resource Types). For more information, see Section 6.3, "Delegating Application Administration."

- Policy Domain administrators have privileges to manage all child objects in the Policy Domain to which they are assigned. One `PolicyDomainAdmin` is generated for each Policy Domain that is created. They are primarily created to delegate the management of policies, permissions and resources within a Policy Domain. For an overview of this concept, see Section 6.4, "Using Policy Domains to Delegate." For additional information, see Section 6.5, "Delegating Policy Domain Administration."

6.3 Delegating Application Administration

The following sections explain how to manage administrators for an Application.

- Section 6.3.1, "Adding a Delegated Administrator for An Application"
- Section 6.3.2, "Modifying or Deleting an Application's Delegated Administrator"

6.3.1 Adding a Delegated Administrator for An Application

This procedure documents how to create a new Administrator Role and assign it to the applicable roles or users. To add a delegated administrator to an Application, proceed as follows.

1. Expand the Applications node in the Navigation Panel.
2. Select the Application to modify.
3. Right-click the Application name and select Open from the menu.

   The General tab, the Delegated Administrators tab and the Policy Distribution tab are all active.
4. Click the Delegated Administrators tab.

   The Application name is listed in the displayed table. Click the arrow next to the Application name to see the default `ApplicationPolicyAdmin` created when the Application object was created. Click the Administrator Role name to display its details, in tabs, below the Delegated Administrators table.

   - Role Details
   - External Role Mapping
   - External User Mapping
5. Click New to create a new Administrator Role.

   Be sure to select the name of the Application to activate New. Alternately, select the Application and select New from the Actions menu. A New Administrator Role dialog is displayed.
6. Provide the following values for the new Administrator Role and click OK.

   - **Name**: The entry must be a unique.
   - **Display Name**
   - **Description**
7. Select the new Administrator Role to activate its configuration tabs.

   The Role Details tab is active.

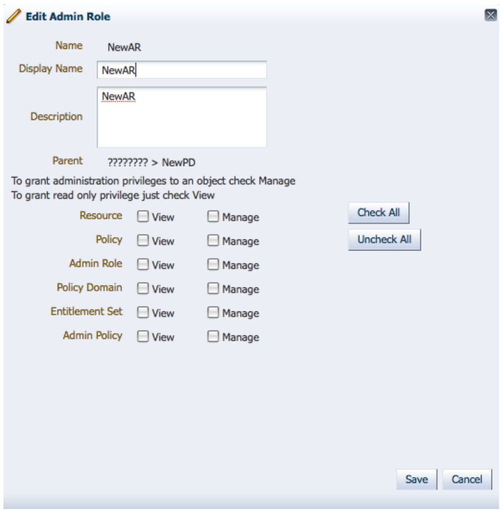
8. Click Edit to define the role details.

   An Edit Administrator Role dialog is displayed.

9. Grant View or Manage privileges for the appropriate policy objects and click Save.

   Figure 6–1 is the Edit Admin Role privileges pop up screen. Select View or Manage for the listed policy objects. For example, Admin Policy allows the administrator to assign new permissions to an Admin Role. Admin Role, however, allows the administrator to assign members to an Admin Role. See Section 2.3, "The Policy Object Glossary" for details on the other listed objects.

Figure 6–1 Edit Admin Role Pop Up Screen



Name NewAR

Display Name NewAR

Description NewAR

Parent ???????? > NewPD

To grant administration privileges to an object check Manage
To grant read only privilege just check View

Resource	□ View	□ Manage
Policy	□ View	□ Manage
Admin Role	□ View	□ Manage
Policy Domain	□ View	□ Manage
Entitlement Set	□ View	□ Manage
Admin Policy	□ View	□ Manage

10. Click the External Role Mapping tab to grant the Administrator Role to members of External Roles.

11. Click Add to display the Search Principals dialog.

12. Complete the query fields in the External Roles search box and click Search.

    Empty strings fetch all roles. The results display in the Search Results table.

13. Select the external role to map to by clicking its name in the table.

    Use Ctrl+click to select multiple roles.

14. Click Add Principals.

    The selected roles display in the External Role Mapping tab.

15. Click the External User Mapping tab to grant the Administrator Role to External Users.

16. Click Add to display the Search Principals dialog.

17. Complete the query fields in the Users search box and click Search.

    Empty strings fetch all roles. The results display in the Search Results table.

18. Select the user to map by selecting its name in the table.

    Use Ctrl+click to select multiple roles.

19. Click Add Principals.

    The selected roles display in the External User Mapping tab.

6.3.2 Modifying or Deleting an Application's Delegated Administrator

To modify or delete an Application's configured Administrator Role, proceed as follows.

1. Expand the Applications node in the Navigation Panel.
2. Select the Application to modify.
3. Right-click the Application name and select Open from the menu.

   The General tab, the Delegated Administrators tab and the Policy Distribution tab are all active.
4. Click the Delegated Administrators tab.
5. Navigate to the Administrator Role you want to modify and select it.

   The Role Details, External Role Mapping and External User Mapping tabs are displayed.
6. Select the tab which contains the configuration to modify or delete.

   - To modify the configuration, see Section 6.3.1, "Adding a Delegated Administrator for An Application" for details.
   - To remove a mapping from an Administrator Role, select the applicable Administrator Role and the appropriate Mapping tab. Select the mapping and click Remove.
   - To delete an Administrator Role, select the Administrator Role and click Delete.

6.4 Using Policy Domains to Delegate

Administration of the policies securing one protected application may be delegated using one or more (optional) Policy Domains. A Policy Domain contains the components of completed policy definitions. It is the amalgamation of a target Resource (an instance of the Resource Type), an Entitlement (the actions that can be performed on the Resource), and a Policy (a rule that assembles the controls and the principals they affect).

The use of multiple Policy Domains allows policies to be partitioned according to some defined logic, such as the architecture of the protected application or how administration of the policies are delegated. For example, one Policy Domain can be used to maintain all policies securing a Resource or multiple Policy Domains can be used to reflect a particular characteristic of the Resource. Different administrators can then be placed in charge of different Policy Domains.

> **Note:** Because the creation of a Policy Domain is optional, if there is no need to delegate policy administration, there is no need to create any Policy Domains. In this case, a default Policy Domain is created with each Application that will contain all the Application's policy objects.

The following sections contain the management procedures for Policy Domains.

- Section 6.4.1, "Creating a Policy Domain"
- Section 6.4.2, "Modifying a Policy Domain"
- Section 6.4.3, "Deleting a Policy Domain"

6.4.1 Creating a Policy Domain

To create a Policy Domain, proceed as follows.

1. Right-click the name of the Application in the Navigation Panel under which the Policy Domain will be created and select **New** from the menu.

   An Untitled page displays in the Home area.

2. Provide the following information for the Policy Domain.

   - **Display Name**
   - **Name**
   - **Description**: Although optional, it is recommended to provide useful information about the entitlement.

3. Select one of the following from the Save menu.

   - Save and Close saves the configuration and renames the tab with the value provided for the Policy Domain's Display Name.
   - Save and Create Another saves the configuration to the information tree in the Navigation Panel but leaves the Untitled area open for you to create another Application.

6.4.2 Modifying a Policy Domain

To modify a Policy Domain, proceed as follows.

1. Navigate to the Application under which the Policy Domain you want to delete was created and expand the information tree.
2. Double click the name of the Policy Domain you want to modify.

   The Policy Domain configuration displays in the Home area.

3. Modify as necessary and click Apply.

6.4.3 Deleting a Policy Domain

To delete a Policy Domain, proceed as follows.

1. Navigate to the Application under which the Policy Domain you want to delete was created and expand the information tree.
2. Double click the name of the Policy Domain you want to delete.

   The Policy Domain configuration displays in the Home area.

3. Click Delete.

   A confirmation dialog is displayed.

4. Click OK to delete.

6.5 Delegating Policy Domain Administration

The following sections describe how to manage administrators for Policy Domains.

- Section 6.5.1, "Adding a Delegated Administrator to a Policy Domain"
- Section 6.5.2, "Modifying or Deleting a Policy Domain's Delegated Administrator"

6.5.1 Adding a Delegated Administrator to a Policy Domain

This procedure documents how to create a new Administrator Role and assign it to the applicable roles or users. To add a delegated administrator to a Policy Domain, proceed as follows.

1. Expand the Applications node in the Navigation Panel.
2. Select the Application to modify.
3. Right-click the Application name and select Open from the menu.

   The General tab, the Delegated Administrators tab and the Policy Distribution tab are all active.
4. Click the Delegated Administrators tab.

   The Policy Domain names are listed in the displayed table. Clicking the arrow next to the Policy Domain expands the hierarchy and displays any Administrator Roles already configured; for example, the default `PolicyDomainAdmin`.
5. Select the Policy Domain under which you will create the Administrator Role.
6. Click New to create a new Administrator Role.

   Be sure to select the name of the Policy Domain to activate New. Alternately, select the Policy Domain and select New from the Actions menu. A New Administrator Role dialog is displayed.
7. Provide the following values for the new Administrator Role and click OK.
   - **Name**: The entry must be a unique.
   - **Display Name**
   - **Description**
8. Select the new Administrator Role to activate its configuration tabs.

   The Role Details tab is active.
9. Click Edit to define the role details.

   An Edit Administrator Role dialog is displayed.
10. Grant View or Manage privileges for the appropriate Policy Domain objects and click Save.
11. Click the External Role Mapping tab.
    a. Click Add to display the Search Principals dialog.
    b. Complete the query fields in the External Roles search box and click Search.

       Empty strings fetch all roles. The results display in the Search Results table.
    c. Select the external role to map to by clicking its name in the table.

       Use Ctrl+click to select multiple roles.
    d. Click Add Principals.

       The selected roles display in the External Role Mapping tab.
12. Click the External User Mapping tab.
    a. Click Add to display the Search Principals dialog.
    b. Complete the query fields in the **Users** search box and click Search.

       Empty strings fetch all roles. The results display in the Search Results table.

c. Select the user to map by selecting its name in the table.

   Use Ctrl+click to select multiple roles.

d. Click Add Principals.

   The selected roles display in the External User Mapping tab.

6.5.2 Modifying or Deleting a Policy Domain’s Delegated Administrator

To modify or delete a Policy Domain’s configured Administrator Role, proceed as follows.

1. Expand the Applications node in the Navigation Panel.
2. Select the Application to modify.
3. Right-click the Application name and select Open from the menu.

   The General tab, the Delegated Administrators tab and the Policy Distribution tab are all active.
4. Click the Delegated Administrators tab.
5. Navigate to the Administrator Role you want to modify and select it.

   The Role Details, External Role Mapping and External User Mapping tabs are displayed.
6. Select the tab which contains the configuration to modify or delete.

   - To modify the configuration, see Section 6.5.1, "Adding a Delegated Administrator to a Policy Domain" for details.
   - To remove a mapping from an Administrator Role, select the applicable Administrator Role and the appropriate Mapping tab. Select the mapping and click Remove.
   - To delete an Administrator Role, select the Administrator Role and click Delete.

6.6 Managing System Administrators Using Administrator Roles

You can delegate system administration privileges to users by creating and configuring System Administrator Roles. By default, `SystemAdmin` is created during installation and is displayed in the System Administrators table when you navigate to System Administrators under the main System Configuration tab. `SystemAdmin` manages system-level resources (including other Administrator Roles, and system configurations and bindings) and maps to the WebLogic Server `weblogic` user.

The following sections document the management operations for all Oracle Entitlements Server System Administrator Roles.

- Section 6.6.1, "Creating a New Administrator Role"
- Section 6.6.2, "Assigning Privileges to an Administrator Role"
- Section 6.6.3, "Modifying Administrator Role Membership"
- Section 6.6.4, "Deleting an Administrator Role"

6.6.1 Creating a New Administrator Role

To create a new Administrator Role, proceed as follows.

1. Select the System Configuration tab from the Home area.

   The System Administrators tab is displayed in the Home area.

2. Click New under Administrator Roles to create a new Administrator Role.

   A dialog is displayed.

3. Provide the following values for the new Administrator Role.

   - **Name**: The entry must be a unique.
   - **Display Name**
   - **Description**

4. Click Create.

6.6.2 Assigning Privileges to an Administrator Role

To assign privileges to an Administrator Role, map external roles, external users or both to the role as documented in this procedure.

1. Select the System Configuration tab from the Home area.

   The System Administrators tab and configured Administrator Roles are displayed in the Home area. Alternately, right-click Administrators and select Open.

2. Select the name of the Administrator Role from the table.

3. Select the Modify or View option to define the Administrator Control.

   Modify defines the administrator as having management (and by proxy viewing) privileges on all system administrator resources. View defines the administrator as having only viewing privileges.

4. Click the External Role Mapping tab.

   a. Click Add or select Add from the Actions menu.

      The Add Roles search dialog is displayed.

   b. Enter a search string in the text box and click the arrow to search for External Roles.

      Alternately, click Search with no search string to return all available External Roles.

   c. Select one or more roles from the results and click Add Selected.

      Alternately, click Add All to add all returned results.

   d. Click Add Principals.

5. Click the External User Mapping tab.

   a. Click Add or select Add from the Actions menu.

      The Add Users search dialog is displayed.

   b. Enter a search string in the text box and click the arrow to search for External Users.

      Alternately, click Search with no search string to return all available External Users.

   c. Select one or more users from the results and click Add Selected.

      Alternately, click Add All to add all returned results.

   d. Click Add Principals.

6.6.3 Modifying Administrator Role Membership

To modify Administrator Role membership, delete the mappings as documented in this procedure.

1. Select the System Configuration tab from the Home area.
2. Double-click System Administrators in the Navigation Panel.

   Alternately, right-click System Administrators and select Open. The System Administrators page is displayed.
3. Select the name of the Administrator Role from the table.
4. Modify the Modify or View Administrator Control as necessary.
5. Click the External Role Mapping tab.
   a. Select the External Role to delete.
   b. Click Remove.

      Alternately, select Remove from the Actions menu.
6. Click the External User Mapping tab.
   a. Select the External User to delete.
   b. Click Remove.

      Alternately, select Remove from the Actions menu.

6.6.4 Deleting an Administrator Role

To delete an Administrator Role, proceed as follows.

1. Select the System Configuration tab from the Home area.
2. Double-click System Administrators in the Navigation Panel.

   Alternately, right-click System Administrators and select Open. The System Administrators page is displayed.
3. Select the name of the Administrator Role from the table.
4. Click Delete.

   A confirmation dialog is displayed.
5. Click Remove.

7 Upgrading Oracle Fusion Applications Policies

The information in this chapter is specific to Oracle Fusion Applications only.

This chapter describes how to use Oracle Authorization Policy Manager to upgrade application policies in an LDAP-based domain policy store with the changes introduced by a new release of the application. Details are in the following sections:

- Overview
- Prerequisites to Patching Policies
- The Policy Upgrade Management Tab
- Analyzing Patch Differences
- Resolving Patch Differences
- Applying a Patch

7.1 Overview

First we introduce some terms used throughout this chapter, and then an overview of the process of upgrading the policy store.

- Section 7.1.1, "Terminology"
- Section 7.1.2, "Upgrading Process Overview"

7.1.1 Terminology

The following terms refer to the three policy stores involved in an application policy upgrading. They are also used in the Authorization Policy Manager user-interface.

Baseline - The original policy store, represented by the XML file `jazn-data.xml` and available with the application out-of-the-box. Presumably, this policy store was migrated to the domain policy store when the application was first deployed.

Production - The domain policy store, where the current state of application policies reside. This store is assumed LDAP-based. Presumably, policies in the application stripe in this store has undergone modifications since the application was first deployed.

Patch - The policy store of the new version of the application, represented by the XML file `jazn-data.xml` and available with the new application out-of-the-box.

7.1.2 Upgrading Process Overview

Application policy upgrading allows security administrators to solve the following problem, with which they are faced every time a new version of an application is released.

Out-of-the-box, an application typically includes the file `jazn-data.xml` (baseline policy store) that describes the application policies for that particular version of the application. Typically, at application deployment the baseline policy store is migrated to the domain policy store (production policy store) for the first time.

Thereafter, application policies in the production store may undergo modifications to accommodate evolving requirements; these changes include adding, deleting, or modifying any application-specific security artifact such as roles, grants, resource types, resources, and entitlements.

When a new version of the application is available and before that new version is deployed, a security administrator needs to:

- Identify the customizations that have been introduced since the migration of the old application version, that is, the delta between the baseline and the production stores.
- Identify the differences between the customized application policies and the policies in the new application version, that is, the delta between the production and patch stores.
- Decide, for each difference, which artifact to use.

Authorization Policy Manager facilitates the resolution of each of the above tasks by providing a security administrator with a user interface that allows him to:

- Analyze a new patch, that is, generate all differences.
- Inspect and decide, for each difference reported by the analysis, which specification to use.
- Apply the patch.

> **Important:** Before patching application policies, make sure that you backup the policy store as explained in Prerequisites to Patching Policies.

7.2 Prerequisites to Patching Policies

The analysis must be performed first. The resolution of changes and conflicts is performed next. These tasks do not have any particular requirements and can be accomplished at different times during one Authorization Policy Manager session or even across different sessions.

Before applying a patch, however, proceed as follows:

1. Take off line any WebLogic domain that uses the policy store where the application policies to be patched reside.
2. Backup the policy store by using either of the following tools:
    1. Oracle Internet Directory `ldifwrite` to obtain an LDIF file for the policy store. For an example of use of this command, see *Oracle Fusion Middleware Application Security Guide*.

2. Oracle Platform Security Services `migrateSecurityStore` to export the policy store into a replica of it. For details about this command, see *Oracle Fusion Middleware Application Security Guide*.

Now you can apply the patch.

If for any reason the policy store needs to be restored, proceed as follows:

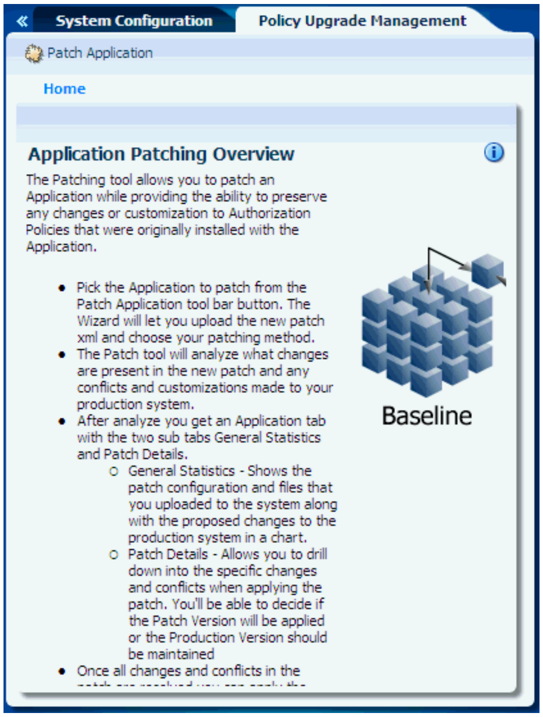
1. If you have saved the policy store in an LDIF file, use `bulkload` to restore it. For details about this command, see *Oracle Fusion Middleware Application Security Guide*.

2. If you have exported the policy store, use Oracle Platform Security Services `migrateSecurityStore` to restore it. For details about this command, see *Oracle Fusion Middleware Application Security Guide*.

7.3 The Policy Upgrade Management Tab

The **Policy Upgrade Management** tab, partially illustrated in Figure 7–1, contains the tab **Home**, where the upgrading process begins and which succinctly describes the steps you follow to upgrade application policies. The first step is to select the application whose policies to upgrade.

Figure 7–1 The Policy Upgrade Management Tab



Application Patching Overview

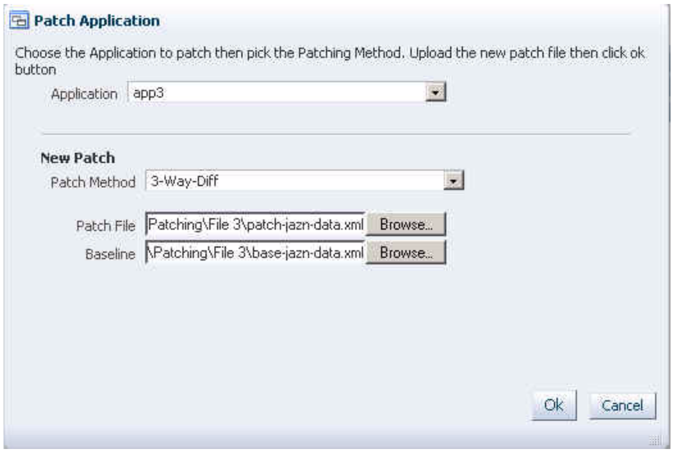
The Patching tool allows you to patch an Application while providing the ability to preserve any changes or customization to Authorization Policies that were originally installed with the Application.

- Pick the Application to patch from the Patch Application tool bar button. The Wizard will let you upload the new patch xml and choose your patching method.
- The Patch tool will analyze what changes are present in the new patch and any conflicts and customizations made to your production system.
- After analyze you get an Application tab with the two sub tabs General Statistics and Patch Details.
  - General Statistics - Shows the patch configuration and files that you uploaded to the system along with the proposed changes to the production system in a chart.
  - Patch Details - Allows you to drill down into the specific changes and conflicts when applying the patch. You'll be able to decide if the Patch Version will be applied or the Production Version should be maintained
- Once all changes and conflicts in the

To select application policies to patch, proceed as follows:

1. In the Home tab of the Policy Upgrade Management page, click the button **Patch Application** at the top left corner of the page to bring up the **Patch Application** dialog illustrated in Figure 7–2.

Figure 7–2 Patch Application Dialog



Patch Application

Choose the Application to patch then pick the Patching Method. Upload the new patch file then click ok button

Application: app3

New Patch

Patch Method: 3-Way-Diff

Patch File: Patching\File 3\patch-jazn-data.xml

Baseline: \Patching\File 3\base-jazn-data.xml

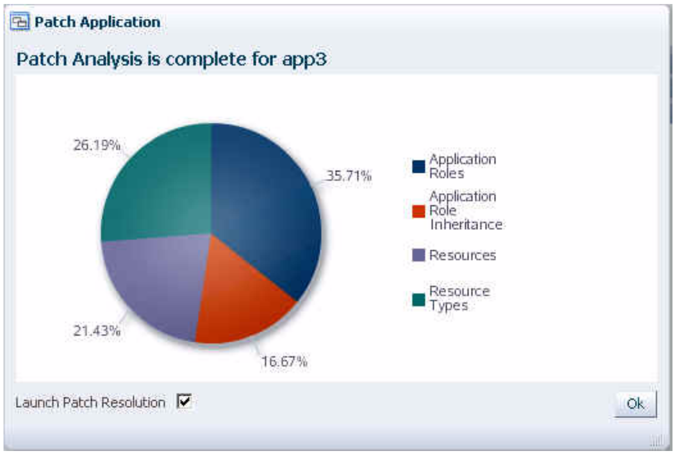
2. Select the application to patch from the pull-down **Application** list.

   This list displays only applications that are currently deployed in the domain. After selecting the application, the dialog takes a different form according to whether or not the application selected has a patching in progress.

   - If the application has a patching in progress, you can continue with it or abort it.
   - If the application does not have a patching in progress, select the **Baseline** file (specifies the location of the baseline policy store) and the **Patch file** (specifies the location of the patch policy store) and click **OK**. The only **Patch Method** available in this release is a 3-way DIFF which considers differences between the baseline, the production, and the patch stores

   The rest of this procedure assumes a new patching process. Authorization Policy Manager displays an indicator showing the progress of the analysis phase in the Patch Application dialog. Once this phase is completed, the Patch Application dialog displays the statistics of the analysis as illustrated in Figure 7–3.

Figure 7–3 Statistics of a Patch Analysis



Patch Application

Patch Analysis is complete for app3

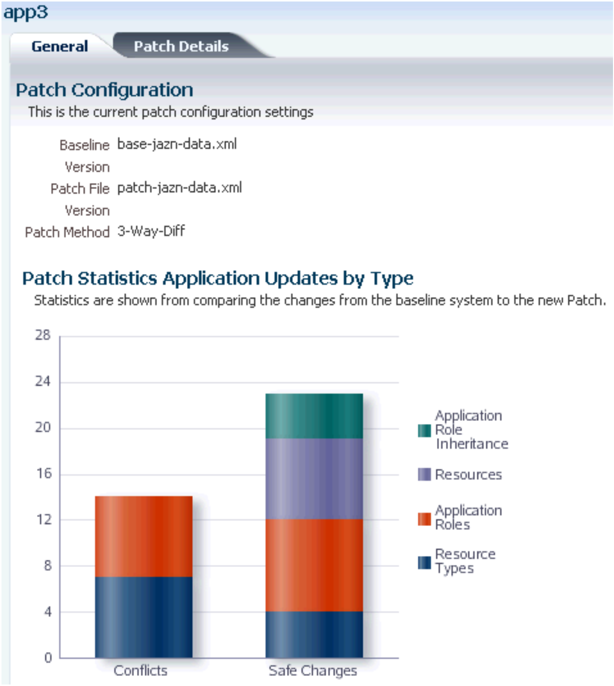
Launch Patch Resolution ☑

3. Check the box **Launch Patch Resolution** (checked by default) and click **OK** to launch the patch resolution phase.

Authorization Policy Manager creates a new tab (named after the application display name) that contains the details of the results, that is, the conflicts and differences encountered, in two sub-tabs:

- **General** - This tab displays the files you have specified at the start of the patching and a chart showing the number changes and conflicts found, per artifacts, between the baseline and the patch stores. For details about these terms, see Section 7.5.1, "Changes and Conflicts." Figure 7–4 illustrates the General tab.

Figure 7–4 The General Tab



Patch Configuration

This is the current patch configuration settings

Baseline base-jazn-data.xml
Version
Patch File patch-jazn-data.xml
Version
Patch Method 3-Way-Diff

Patch Statistics Application Updates by Type

Statistics are shown from comparing the changes from the baseline system to the new Patch.

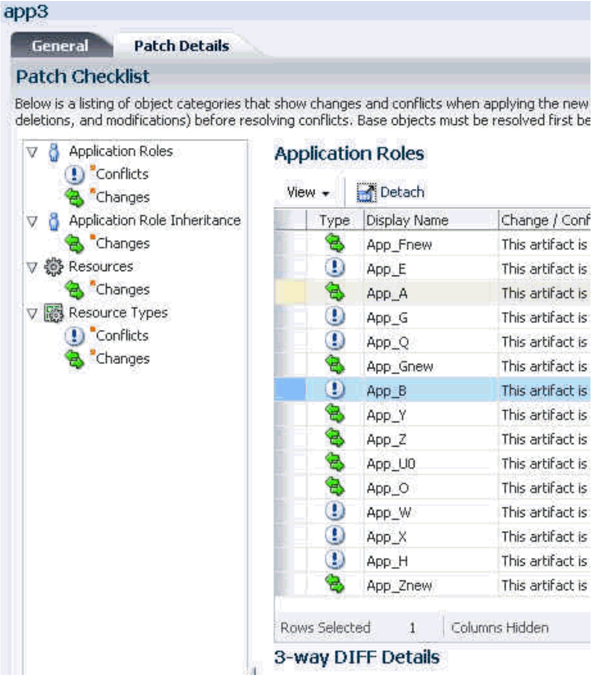
To terminate the current patching process and to delete the analysis data gathered thus far, click the button **Discard**; once the patch is discarded, the tab for the application is deleted from the Patching tab.

- **Patch Details** - This tab displays the proposed changes and conflicts encountered per security artifact. The details of this tab are explained in the next section.

7.4 Analyzing Patch Differences

The **Patch Details** tab, illustrated partially in Figure 7–5, contains two major areas: the left area displays a hierarchical overview of changes and conflicts per artifact that resulted from the comparisons; the right area displays the details of changes and conflicts for an artifact selected from the left area.

Figure 7–5 Patch Details Tab



app3

Patch Checklist

Below is a listing of object categories that show changes and conflicts when applying the new
deletions, and modifications) before resolving conflicts. Base objects must be resolved first be

- Application Roles
  - *Conflicts
  - *Changes
- Application Role Inheritance
  - *Changes
- Resources
  - *Changes
- Resource Types
  - *Conflicts
  - *Changes

Application Roles

Type	Display Name	Change / Conf
⇄	App_Fnew	This artifact is
!	App_E	This artifact is
⇄	App_A	This artifact is
!	App_G	This artifact is
!	App_Q	This artifact is
⇄	App_Gnew	This artifact is
!	App_B	This artifact is
⇄	App_Y	This artifact is
⇄	App_Z	This artifact is
⇄	App_U0	This artifact is
⇄	App_O	This artifact is
!	App_W	This artifact is
!	App_X	This artifact is
!	App_H	This artifact is
⇄	App_Znew	This artifact is

Rows Selected 1 Columns Hidden

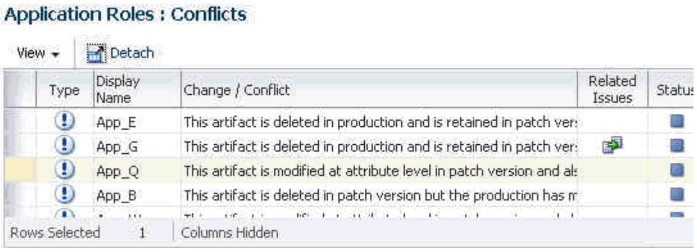
3-way DIFF Details

To display the specifics of an object's differences in the right area, click **Changes** or **Conflicts** under the object. Each row in the table has a type icon that indicates whether the difference is a change (double arrow icon) or a conflict (exclamation mark icon). (For details, see Section 7.5.1, "Changes and Conflicts.")

To view a change or conflict for a specific artifact, select the corresponding icon. All changes or conflicts are displayed in a table at the top of the page. The **Status** column shows whether a change or a conflict has been resolved (green check icon) or not (gray square icon). The **Related Issues** column shows whether a change or conflict has implied dependencies; click the icon in this column to display the **Patch Artifact Dependencies** dialog and see, among other details, the reasons why other artifacts would be affected when resolving a difference for this artifact. Figure 7–6 partially illustrates this page.

.

Figure 7–6 Viewing Artifact Conflicts



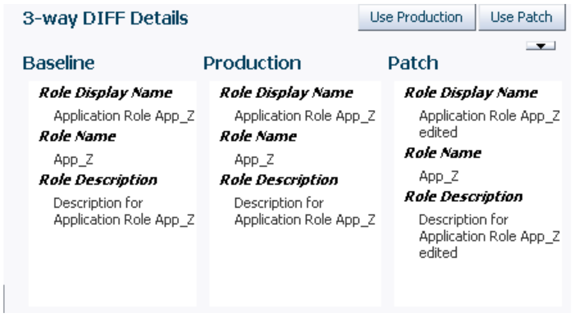
Application Roles : Conflicts

Type	Display Name	Change / Conflict	Related Issues	Status
!	App_E	This artifact is deleted in production and is retained in patch ver:		■
!	App_G	This artifact is deleted in production and is retained in patch ver:	⧉	■
!	App_Q	This artifact is modified at attribute level in patch version and al:		■
!	App_B	This artifact is deleted in patch version but the production has m		■

Rows Selected 1 Columns Hidden

To view conflict details for a specific item in the table, select the item to display the different specifications found in the 3-Way DIFF Details area. Figure 7–7 illustrates the differences for a role.

Figure 7–7 Displaying Difference Details



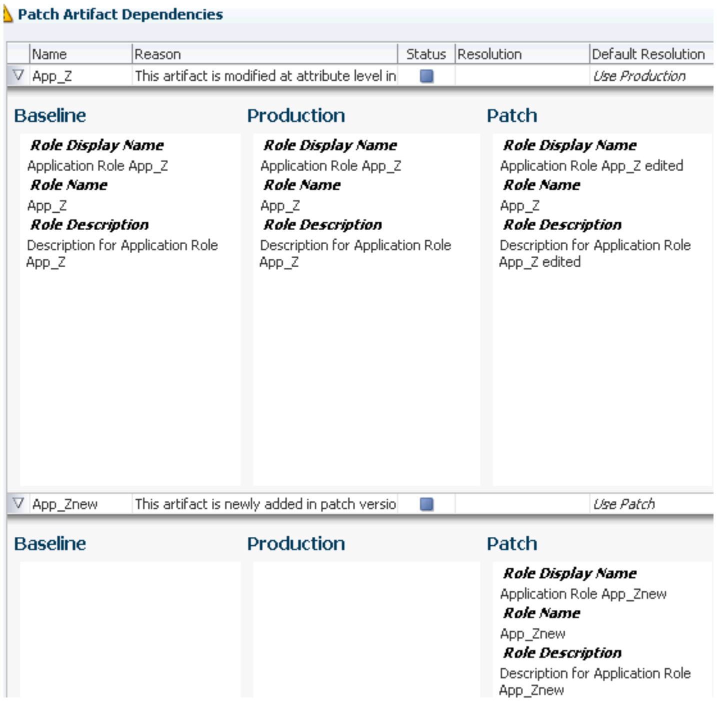
3-way DIFF Details

Baseline	Production	Patch
Role Display Name	***Role Display Name***	***Role Display Name***
Application Role App_Z	Application Role App_Z	Application Role App_Z edited
Role Name	***Role Name***	***Role Name***
App_Z	App_Z	App_Z
Role Description	***Role Description***	***Role Description***
Description for Application Role App_Z	Description for Application Role App_Z	Description for Application Role App_Z edited

Figure 7–8 partially illustrates the dependencies implied by differences in a pair of roles. Specifically, it illustrates a baseline role App_Z that is not modified in the production store but modified in the patch store as follows:

1. The display name and the role description are changed.
2. The new role App_Znew is a child of the role App_Z.

Figure 7–8 Viewing Dependencies Implied by a Conflict or Change



Patch Artifact Dependencies

Name	Reason	Status	Resolution	Default Resolution
▽ App_Z	This artifact is modified at attribute level in	■		*Use Production*
▽ App_Znew	This artifact is newly added in patch versio	■		*Use Patch*

App_Z:

Baseline	Production	Patch
Role Display Name	***Role Display Name***	***Role Display Name***
Application Role App_Z	Application Role App_Z	Application Role App_Z edited
Role Name	***Role Name***	***Role Name***
App_Z	App_Z	App_Z
Role Description	***Role Description***	***Role Description***
Description for Application Role App_Z	Description for Application Role App_Z	Description for Application Role App_Z edited

App_Znew:

Baseline	Production	Patch
		Role Display Name
		Application Role App_Znew
		Role Name
		App_Znew
		Role Description
		Description for Application Role App_Znew

7.5 Resolving Patch Differences

A patch difference identifies a disparity between the specifications of a security artifact in the some of the policy stores involved in the analysis. Oracle Authorization Policy Manager lists patch differences as changes or conflicts. These terms and how to resolve them are explained in the following sections:

- Changes and Conflicts
- Resolving Changes and Conflicts

7.5.1 Changes and Conflicts

To better explain the terminology used, assume that Abase, Aprod, and Apatch denote the states of an artifact in the baseline, production, and patch stores, respectively.

A patch difference is called a *change* when Abase and Apatch are equal, and Aprod is different to Apatch.

A patch difference is called a *conflict* when Abase and Apatch are different, and Aprod is different from Apatch.

7.5.2 Resolving Changes and Conflicts

Resolving an artifact change or conflict means choosing which specification to use: the one in the production store or the one in the patch store.

Even though there is a default resolution for each artifact change or conflict, it is recommended that all changes and conflicts be resolved manually before you proceed forward to applying the patch.

To resolve a change or conflict for an artifact, proceed as follows:

1. Select the artifact in the **Conflicts** table, to display the specifications for the artifact found in each of the three stores at the bottom of the page.
2. Inspect specification differences and decide which one to use; to use the production store, click the button **Use Production**; to use the specification in the patch store, click the button **Use Patch**.

> **Important Note:** The decision that you make in this step may imply necessary changes to other artifacts. These changes, necessary to preserve data consistency, are called *dependencies*.
>
> Oracle Authorization Policy Manager displays the dependencies that a decision implies and requests your confirmation before setting the value.

The decision value set for a change or conflict can be reset at any time. To any change or conflict left unresolved, Oracle Authorization Policy Manager sets one of the following default values:

- For a change, Use Patch.
- For a conflict, Use Production.

7.6 Applying a Patch

The procedure in this section assumes that:

- All changes and conflicts reported in the Patch Checklist of the Patch Details tab have been resolved (manually or by default).
- The prerequisites stated in Prerequisites to Patching Policies are met.

To apply a patch, proceed as follows:

1. Click the button **Apply Patch** in the application's patching tab to initiate the patching process, which will modify the application policy stripe in the domain LDAP store.

2. Once the application of the patch is completed, you are ready to deploy the new version of the application.

   Make sure that when deploying it, *the automatic migration of policies is turned off* so that the just patched application policies are not modified when the application is deployed.

   For details about how to manage the migration of policies when the application is deployed with Oracle Enterprise Manager Fusion Middleware Control, see *Oracle Fusion Middleware Application Security Guide*

8 Customizing the User Interface

This chapter explains several customizations available in Oracle Authorization Policy Manager in the following sections:

- Customizing Authorization Policy Manager
- Customizing Headers, Footers, and Logo
- Customizing Color Schemes
- Customizing the Login Page

8.1 Customizing Authorization Policy Manager

All customizations described in this chapter require modifying data in one or both of the following file archives:

```
$ORACLE_IDM_HOME$/apm/modules/oracle.security.apm_11.1.1/oracle.security.apm.ear
$ORACLE_IDM_HOME$/apm/modules/oracle.security.apm_
   11.1.1/oracle.security.apm.core.view.war
```

> **Note:** Before you begin, it is recommended that you backup these Authorization Policy Manager EAR and WAR files.

Any customizations applied to a version of Authorization Policy Manager must be specified again every time a new version is installed. The following procedure specifies, from a high level, how to customize Authorization Policy Manager.

1. Unzip the EAR, WAR and view WAR files using the following commands:

   ```
   $ unzip -d $tempDir/ear $ORACLE_HOME$/apm/modules/oracle.security.apm_
      11.1.1/oracle.security.apm.ear
   $ unzip -d $tempDir/war $tempDir/ear/oracle.security.apm.war
   $ unzip -d $tempDir/viewWar $ORACLE_HOME$/apm/modules/
      oracle.security.apm_11.1.1/oracle.security.apm.core.view.war
   ```

2. Modify one or more of the unzipped files as documented in one of the following sections of this chapter.

   - Section 8.2, "Customizing Headers, Footers, and Logo"
   - Section 8.3, "Customizing Color Schemes"
   - Section 8.4, "Customizing the Login Page"

3. Rearchive the modified EAR, WAR and view WAR files using the following commands:

```
$ zip $tempDir/ear/oracle.security.apm.war $tempDir/war/*
$ zip $ORACLE_HOME$/apm/modules/oracle.security.apm_
   11.1.1/oracle.security.apm.ear $tempDir/ear/*
$ zip $ORACLE_HOME$/apm/modules/oracle.security.apm_
   11.1.1/oracle.security.apm.core.view.war $temp/viewWar/*
```

4. Redeploy Authorization Policy Manager.

8.2 Customizing Headers, Footers, and Logo

Use the following procedure to customize headers, footers, and the logo.

1. Unzip the view WAR file.

```
$ORACLE_HOME$/apm/modules/oracle.security.apm_
   11.1.1/oracle.security.apm.core.view.war
```

2. Open `AuthPolicyMgr.jspx` file and apply any or all of the following modifications.

   - Specify a new branding title (header) by modifying the `branding` facet.

   ```
   <f:facet name="branding">
    <af:outputText value="My Custom Application Title" noWrap="true"
   id="ot1"/>
   </f:facet>
   ```

   - Specify a new footer by modifying the `appAbout` and `appCopyright` facets.

   ```
   <f:facet name="appAbout">
   <af:outputText value="My Custom Footer at Right" noWrap="true" id="ot2"/>
   </f:facet>
   <f:facet name="appCopyright">
   <af:outputText value="My Custom Footer at Left"  noWrap="true" id="ot3"/>
   </f:facet>
   ```

   - Specify a new logo image as follows:

     a. Insert your resource in the `metaContainer` facet.

     ```
     <f:facet name="metaContainer">
     ....
     <af:resource type="css">
     .MyCustomBrandingLogo {
     background-image:url(/apm/images/world_36x20.png);
      background-position:center;
      background-repeat:no-repeat; display:block;
     height:2.5em; width:119px;
                 }
      </af:resource>
     ...
     </f:facet>
     ```

     Be sure to leave all other content inside the `metaContainer` facet as is.

     b. Specify the style class name (defined in the previous step) as the attribute value of the `pageTemplate` tag.

     ```
     <af:pageTemplate viewId="/templates/IdmShell.jspx"
      value="#{bindings.pageTemplateBinding}" id="pt1">
     ```

```
...
<f:attribute name="brandingLogoCls" value="MyCustomBrandingLogo"/>
...
```

Be sure to leave all other content inside the `pageTemplate` tag as is.

3. Rearchive the view WAR file.

```
$ zip $ORACLE_HOME$/apm/modules/oracle.security.apm_
   11.1.1/oracle.security.apm.core.view.war $temp/viewWar/*
```

4. Redeploy Authorization Policy Manager.

8.3 Customizing Color Schemes

You can develop a new skin to apply to a web application. Use the following procedure to customize the Authorization Policy Manager color scheme. It assumes that you have a new skin available to reference.

> **Note:** Authorization Policy Manager uses the Oracle Application Development Framework (ADF) and supports ADF skinning. See the *Oracle Fusion Middleware Skin Editor User's Guide for Oracle Application Development Framework* for more information on ADF skins.

1. Unzip the EAR and WAR files.

```
$ unzip -d $tempDir/ear $ORACLE_HOME$/apm/modules/oracle.security.apm_
   11.1.1/oracle.security.apm.ear
$ unzip -d $tempDir/war $tempDir/ear/oracle.security.apm.war
```

2. Open the `Trinidad-config.xml` file.

   This file is typically located in the decompressed WAR's `WEB-INF` folder.

3. Specify the value of the new skin location in the `skin-family` tag.

```
<trinidad-config xmlns="http://myfaces.apache.org/trinidad/config">
...
<skin-family>MyCustomSkin</skin-family>
...
</trinidad-config>
```

4. Rearchive the modified EAR and WAR files using the following commands:

```
$ zip $tempDir/ear/oracle.security.apm.war $tempDir/war/*
$ zip $ORACLE_HOME$/apm/modules/oracle.security.apm_
   11.1.1/oracle.security.apm.ear $tempDir/ear/*
```

5. Redeploy Authorization Policy Manager.

8.4 Customizing the Login Page

Use the following procedure to customize the login and login error pages.

1. Unzip the EAR file.

```
$ unzip -d $tempDir/ear $ORACLE_HOME$/apm/modules/oracle.security.apm_
   11.1.1/oracle.security.apm.ear
```

2. Open the `web.xml` file.

This file is typically located in the decompressed EAR's `WEB-INF` folder.

3. Specify the appropriate values for the `form-login-page` and `form-error-page` under the element `form-login-config`.

```
<login-config>
 <form-login-config>
  <form-login-page>/MyCustomLoginPage.html</form-login-page>
  <form-error-page> MyCustomLoginErrorPage.html </form-error-page>
 </form-login-config>
</login-config>
```

4. Rearchive the modified EAR file using the following commands:

```
$ zip $ORACLE_HOME$/apm/modules/oracle.security.apm_
   11.1.1/oracle.security.apm.ear $tempDir/ear/*
```

5. Redeploy Authorization Policy Manager.

9 Managing Policy Distribution

Policy distribution comprises the process used to make configured policies and policy data available for evaluation. Evaluation of the policies will produce a *grant* or *deny* authorization decision in answer to an access request. This chapter contains the following sections.

- Understanding Policy Distribution
- Defining Distribution Modes
- Distributing Policies

9.1 Understanding Policy Distribution

Managing policies and distributing them are distinct operations. Policy management operations are used to define, modify and delete policies in the policy store. The Policy Distribution Component then makes the policies available to a Security Module where the data is used to grant or deny access to a protected resource. Policies are not enforced until they are distributed. Policy distribution may include any or all of the following actions:

- Reading policies from the policy store.
- Caching policy objects in the in-memory policy cache maintained by the Security Module for use during authorization request processing.
- Preserving policy objects in a file-based persistent cache, local to the Policy Distribution Component, that provides independence from the policy store.

Both the central Oracle Entitlements Server Administration Console and the locally-installed (to the protected application) Security Module contain the Policy Distribution Component. This architecture allows two deployment scenarios: the first involves a centralized Policy Distribution Component that can communicate with many Security Modules while the second involves a Policy Distribution Component that is local to, and communicates with, one Security Module.

> **Note:** For details on configuring a Security Module for policy distribution, see Section C.1, "Policy Distribution Configuration." For details on creating definitions and binding Security Modules, see Section 12.2, "Configuring Security Module Definitions."

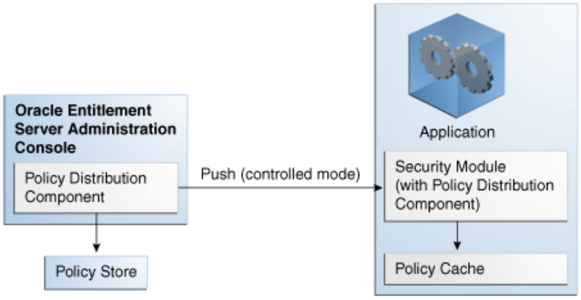
The following sections contain more information.

- Section 9.1.1, "Using a Central Policy Distribution Component"
- Section 9.1.2, "Using a Local Policy Distribution Component"

9.1.1 Using a Central Policy Distribution Component

The centralized Policy Distribution Component scenario involves the use of the Policy Distribution Component (within the Administration Console) to act as a server communicating with the Security Module's Policy Distribution Component client. Figure 9–1 illustrates how, in this scenario, the Security Module's Policy Distribution Component client does not communicate with the policy store. The distribution of policies is initiated by the Oracle Entitlements Server administrator and *pushed* to the Policy Distribution Component client. Currently, data can only be pushed in a *controlled* manner as described in Section 9.2.1, "Controlled Distribution." This scenario allows for a central Policy Distribution Component that can communicate with many Security Modules.

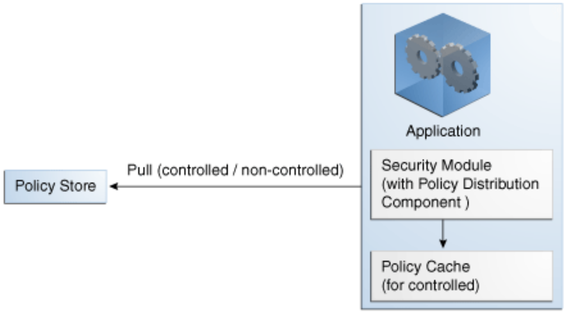
Figure 9–1 Using Oracle Entitlements Server Policy Distribution Component


Oracle Entitlement Server Administration Console
Policy Distribution Component
Policy Store
Push (controlled mode)
Application
Security Module (with Policy Distribution Component)
Policy Cache

9.1.2 Using a Local Policy Distribution Component

The local (to the Security Module) scenario involves the Security Module's Policy Distribution Component communicating directly with the policy store. This scenario allows for a local Policy Distribution Component to communicate with one Security Module only. The application administers management operations and decides when the Security Module instance of the Policy Distribution Component will distribute policies or policy deltas. In this deployment, as illustrated in Figure 9–2, the Policy Distribution Component *pulls* data from the policy store (by periodically checking the policy store for data to be distributed) and sends policy data from the policy store, making it available to the PDP after administrator-initiated policy distribution.

Figure 9–2 Using Security Module Policy Distribution Component


Policy Store
Pull (controlled / non-controlled)
Application
Security Module (with Policy Distribution Component)
Policy Cache (for controlled)

Currently, data can be pulled in either a controlled manner as described in Section 9.2.1, "Controlled Distribution" or a non-controlled manner as described in Section 9.2.2, "Non-controlled Distribution."

9.2 Defining Distribution Modes

Oracle Entitlements Server handles the task of distributing policies to individual Security Modules that protect applications and services. Policy data is distributed in either a controlled manner or a non-controlled manner. The distribution mode is defined in the `jps-config.xml` configuration file for each Security Module. The specified distribution mode is applicable for all Application objects bound to that Security Module. The following sections have more information on the distribution modes.

- Section 9.2.1, "Controlled Distribution"
- Section 9.2.2, "Non-controlled Distribution"

9.2.1 Controlled Distribution

Controlled distribution is the default distribution mode. It is initiated by the Policy Distribution Component, ensuring that the PDP client (Security Module) receives policy data that has been created or modified since the last distribution. In this respect, distribution is controlled by the policy administrator who takes explicit action to distribute the new or updated policy data. (The Policy Distribution Component maintains a versioning mechanism to keep track of policy changes and distribution.) When controlled distribution is enabled, the Security Module cannot request distribution of the Policy Distribution Component directly.

> **Note:** The exception is when a Security Module starts and registers itself with the Policy Distribution Component with a Configuration ID. The policies are distributed to the Security Module based on this registration.

With controlled distribution, the Policy Distribution Component distributes new and updated policy data to the Security Module where the data is stored in a local persistent cache, a file-based cache maintained by the PDP to store policy objects and provide independence from the policy store. The Policy Distribution Component does not maintain constant live connections to its Security Module clients; it will establish a connection before distributing policy to it. Thus, the Security Module is not dependent on the policy store for making policy decisions; it can use its own local cache if the policy store is offline. When the Security Module starts, it will check if the policy store is available. If it is not available, the Security Module will use policy data from the local persistent cache.

> **Caution:** Controlled distribution is supported only on database type policy stores - not on LDAP-based policy stores. If the distribution API is invoked for an LDAP policy store, it will be non-operable.

With controlled distribution, if any policy distribution operation fails, the entire policy distribution fails. By default, controlled distribution is disabled.

9.2.2 Non-controlled Distribution

When the PDP client (Security Module) periodically retrieves (or pulls) policies and policy modifications from a policy store, it is referred to as non-controlled distribution. Non-controlled distribution makes policy changes available as soon as they are saved to the policy store. Non-controlled distribution is initiated by the Security Module and may retrieve policies that are not yet complete. The policy store must be online and constantly available for non-controlled distribution. Non-controlled distribution is supported on any policy store type.

9.3 Distributing Policies

From a high level, the following steps are needed to get to the point where you can distribute policies.

1. Create a Security Module definition.

   See Section 12.2, "Configuring Security Module Definitions."

2. Bind the definition to the appropriate Application.

   See Section 12.2.2, "Binding an Application to a Security Module." To unbind the Security Module, see Section 12.2.3, "Unbinding an Application From a Security Module."

3. Open the Application in the Home area.

   See Section 9.3.1, "Distributing Policies Using the Administration Console."

4. Distribute the policies.

   See Section 9.3.1, "Distributing Policies Using the Administration Console."

9.3.1 Distributing Policies Using the Administration Console

Policies are distributed from within an Application. The following procedure documents how to distribute policies using the Administration Console.

1. Expand the Applications node in the Navigation Panel.
2. Select the Application to modify.
3. Right-click the Application name and select Open from the menu.

   The General tab, the Delegated Administrators tab and the Policy Distribution tab are all active.

4. Click the Policy Distribution tab.
5. Select the definition of the Security Module to which you will distribute policies.
6. Click Distribute.
7. Click Refresh to update the distribution progress.

10 Oracle Fusion Applications Data Role Templates

The information in this chapter is specific to Oracle Fusion Applications only.

This chapter describes what data role templates are and the procedures to create, run, and maintain them. It contains the following sections:

10.1 Using Data Role Templates

A template or data role template specifies key characteristics of external roles and data security policies. These characteristics include:

- A set of base external roles
- A set of dimension values
- A set of naming rules

When run, the data role template generates all the external roles and the data security policies that satisfy the values in the template. The external roles generated (by a template run) are stored in the domain identity store; the data security policies generated are stored in the data security store; templates are stored in the metadata storage (MDS).

The basic principle behind the generation of external roles and data policies is that one can take the cross product of the first two sets of characteristics (external roles times dimension values) to obtain a set of external roles named according to the third set (naming rules), and associate them with a set of permissions, for a given data stripe, in data security policies.

The external roles and the data security policies that a template run generates are specified as a set of external roles and a set of dimensions (rows or attributes returned by an SQL query). Each dimension attribute is associated with an alias, which is used (by the naming conventions) to generate names for the roles and data security policies generated.

A dimension attribute can be the attribute return by an SQL query, such as, the following:

```
where territory=US, business unit=Finance, and legal entity=North America
```

The number of external roles generated equals the number of specified external roles times the number of rows returned by the query (or number of dimensions). Each external role generated inherits from the corresponding specified external role.

For example, a template specifying the external roles Employee-Role and Manager-Role, the dimensions US and UK, and the naming rule [external role]:[dimension code name] would generate the following four external roles:

Employee-Role:US, Employee-Role:UK, Manager-Role:US, Manager-Role:UK

Each of the four generated role inherits from one of the specified external roles, Employee-Role or Manager-Role.

The list of external roles and data security policies that a template run generates can be previewed, that is, displayed *before* the actual creation of roles and associated data security policies takes place.

10.2 Before You Begin

In addition to the data sources listed in Section 1.2, "Installing and Configuring Authorization Policy Manager," the use of templates requires that two other data sources, described in Table 10–1, also be configured.

Table 10–1 Data Sources Required by Templates

Data Source Name	JNDI Name	Description
ApmRgxDimDBDS	jdbc/ApmRgxDimDBDS	Used by role templates to execute dimension SQLs.
ApplicationDB	jdbc/ApplicationDBDS	Stores role template records to create security artifacts.

All data sources can be configured with the WebLogic Console by navigating to **JDBC > Data Sources**. The data source ApmRgxDimDBDS must be created with a credential that includes the database writing privilege.

10.3 Creating a Template

To create a new template, proceed as follows:

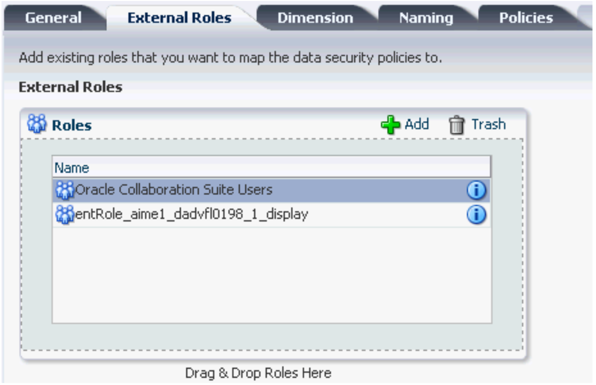
1. Select **Global > Role Templates**, in the left panel, and then click **New** to display an Untitled page in the right panel containing six tabs: General, External Roles, Dimension, Naming, Policies, and Summary.
2. In the **General** tab, enter the following data for the template being created:
    - A display name (required)
    - A name (required)
    - A description (optional)
    - A template group (optional) - This attribute allows searching templates by group and running simultaneously the set of templates in a group.

3. In the **External Roles** tab, specify the external roles for the template in one of the following ways:

   - Click **Add**, at the top of the Roles area, to display bring up the **Add External Role** dialog where you can search for external roles matching a given pattern; then select roles from the results of the query and click Add. The role(s) selected are displayed in the **Roles** table.
   - Perform a regular search for external roles and drag-and-drop the desired roles from the Search Results list into the **Roles** table.

   Figure 10–1 illustrates the Roles table in the External Roles tab after two external roles have been added to the table. When the mouse hovers the blue icon, at the right of a role row, the following information about the role is displayed: the role code, the role name, and the role description; these three attributes can always be used in the Naming tab to specify the names of generated roles.

Figure 10–1 Creating a Template, External Roles



Add existing roles that you want to map the data security policies to.

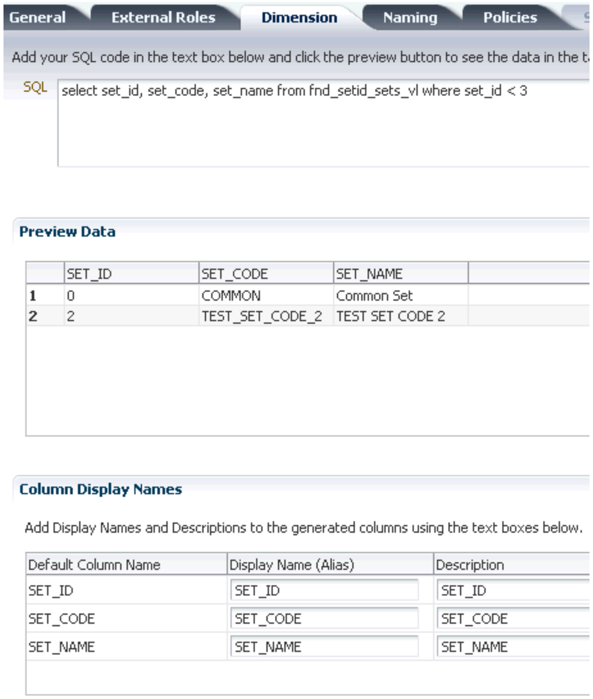
Name
Oracle Collaboration Suite Users
entRole_aime1_dadvfl0198_1_display

4. In the **Dimension** tab, specify the SQL that identifies the dimensions of the template.

   The user must have access privilege to the data queried. The data returned by that SQL is displayed in the **Preview Data** table. Optionally, enter aliases for the column names of the returned data in the **Column Display Names** table, at the bottom of the page.

   Figure 10–2 illustrates the Dimension tab with an SQL query, the data returned by it, and display name aliases; the attributes SET_ID, SET_CODE, and SET_NAME can be used in the Naming tab to specify the names of generated roles.

Figure 10–2 Creating a Template, Dimensions



Add your SQL code in the text box below and click the preview button to see the data in the t

SQL
```
select set_id, set_code, set_name from fnd_setid_sets_vl where set_id < 3
```

Preview Data

	SET_ID	SET_CODE	SET_NAME
1	0	COMMON	Common Set
2	2	TEST_SET_CODE_2	TEST SET CODE 2

Column Display Names

Add Display Names and Descriptions to the generated columns using the text boxes below.

Default Column Name	Display Name (Alias)	Description
SET_ID	SET_ID	SET_ID
SET_CODE	SET_CODE	SET_CODE
SET_NAME	SET_NAME	SET_NAME

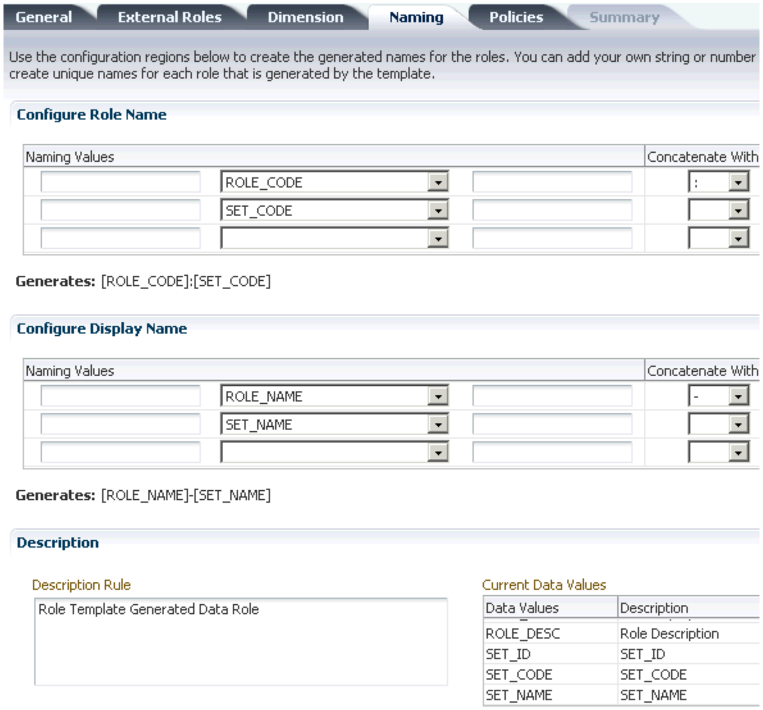
5. In the **Naming** tab, specify the rule to follow to generate names of the data roles created by the template. These names are put together by concatenating several strings that you specify in the area **Configure Role Name**. Typically, one chooses an attribute of the base role and an attribute of the dimension (such as SET_ID, SET_CODE, or SET_NAME in Figure 10–2); the role attributes Role_Code, Role_Name, and Role_Descrip are available by default. The resulting names must be unique.

   Similarly, specify the rule to follow to generated display names for the data roles created by the template. These names are put together by concatenating several strings that you specify in the area **Configure Display Name**. The resulting names need not be unique, but it is recommended that you specify enough attributes to make them unique too.

   Optionally, enter a description for the roles generated in the area **Description**.

   Figure 10–3 illustrates a portion of a Naming tab with naming values for the names and the display names for the external roles generated by the template. Note the following points: (a) the pattern of the concatenation is shown at the bottom of each area after the heading **Generates**; (b) the use of square brackets in the description to refer to data values.

Figure 10–3 Creating a Template, Role Naming



Use the configuration regions below to create the generated names for the roles. You can add your own string or number create unique names for each role that is generated by the template.

Configure Role Name

Naming Values			Concatenate With
	ROLE_CODE		:
	SET_CODE		

Generates: [ROLE_CODE]:[SET_CODE]

Configure Display Name

Naming Values			Concatenate With
	ROLE_NAME		-
	SET_NAME		

Generates: [ROLE_NAME]-[SET_NAME]

Description

Description Rule

Role Template Generated Data Role

Current Data Values

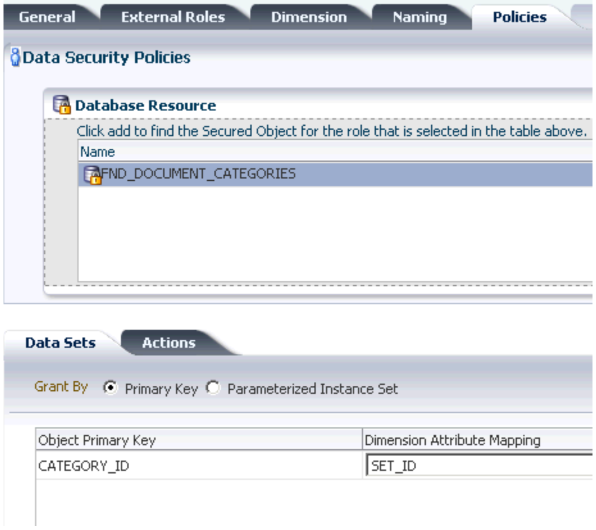
Data Values	Description
ROLE_DESC	Role Description
SET_ID	SET_ID
SET_CODE	SET_CODE
SET_NAME	SET_NAME

6. In the **Policies** tab, specify the rules to create data set grants, as follows:

   - In the **Database Resource** area, use the button **Add** to add a database resource, that is, the object to be secured by the generated data security grants.
   - In the **Data Sets** tab, specify wether the grant is using a **Primary Key** or an **Instance Set** (the instance set is selected from the available instance sets associated with the resource, which are defined at resource creation), and how the data set is mapped to a dimension attribute.
   - In the **Actions** tab, specify the actions allowed on the database resource.

   Figure 10–4 illustrates the specification of a data set by a primary key and the corresponding mapping to a dimension attribute; Figure 10–5 illustrates the specification of a data set by an instance set and the corresponding mapping to dimension attributes; and Figure 10–6 illustrates the selection of actions allowed on the database resource.

Figure 10–4 Creating a Template, Specify Data Set with Primary Key



Data Security Policies

Database Resource

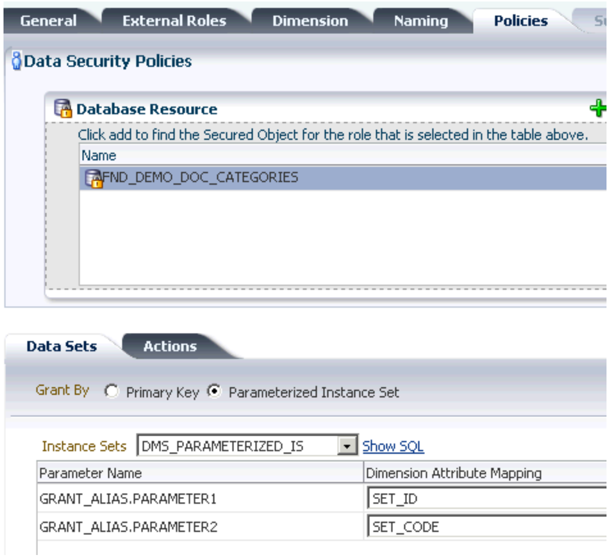
Click add to find the Secured Object for the role that is selected in the table above.

Name
FND_DOCUMENT_CATEGORIES

Grant By ● Primary Key ○ Parameterized Instance Set

Object Primary Key	Dimension Attribute Mapping
CATEGORY_ID	SET_ID

Figure 10–5 Creating a Template, Specify Data Set with Instance Set



Data Security Policies

Database Resource

Click add to find the Secured Object for the role that is selected in the table above.

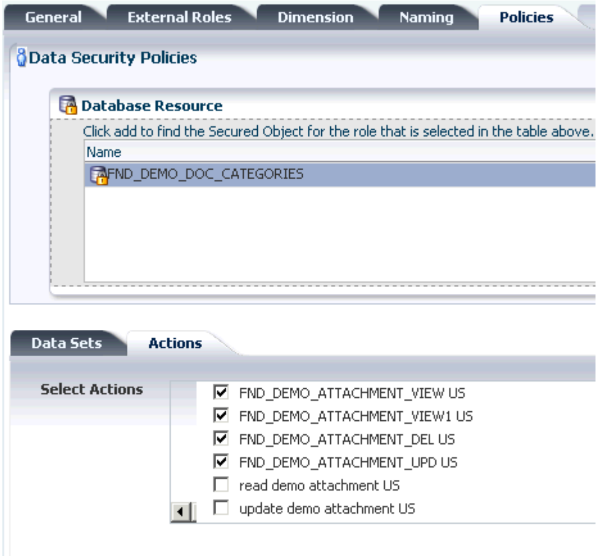
Name
FND_DEMO_DOC_CATEGORIES

Grant By ○ Primary Key ● Parameterized Instance Set

Instance Sets DMS_PARAMETERIZED_IS

Parameter Name	Dimension Attribute Mapping
GRANT_ALIAS.PARAMETER1	SET_ID
GRANT_ALIAS.PARAMETER2	SET_CODE

Figure 10–6 Creating a Template, Specifying Actions



Click add to find the Secured Object for the role that is selected in the table above.

Name
FND_DEMO_DOC_CATEGORIES

Select Actions

☑ FND_DEMO_ATTACHMENT_VIEW US
☑ FND_DEMO_ATTACHMENT_VIEW1 US
☑ FND_DEMO_ATTACHMENT_DEL US
☑ FND_DEMO_ATTACHMENT_UPD US
☐ read demo attachment US
☐ update demo attachment US

7. Click **Save**. Oracle Authorization Policy Manager validates the information supplied and, if all data passes validation, the template is saved and the tab **Summary** becomes available.

10.4 Running a Template

The roles that a template run generates can be previewed *before* the creation of security artifacts takes place. The procedures in this section assume that the template (mentioned in the procedures) has been created and saved.

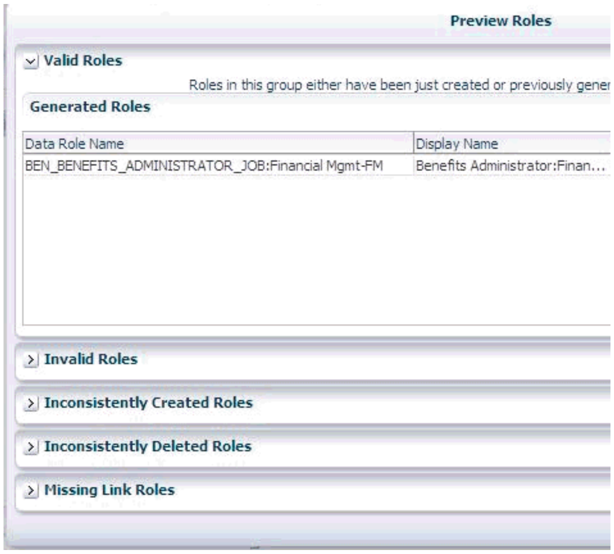
A template or a set of templates can also be run programmatically via web-services. For details, see Section 10.4.1, "Running Templates Programmatically."

To preview the external roles that a template run would generate, proceed as follows:

1. Open the template and bring the **Summary** tab to the foreground (this tab is available since the template has been saved).
2. Click the button **Preview Roles**, near the top of the page, to display the **Preview Roles** dialog, where the external roles that would be generated by an actual template run are grouped in the following five disjoint categories:
   - **Valid Roles** - Set of roles with no issues.
   - **Invalid Roles** - Set of roles with no base role in the identity store.
   - **Inconsistently Created Roles** - Set of roles with identical names to existing roles in the identity store. These roles, typically, get to be included in this category because of a change or deletion in records from where the dimensions are computed.
   - **Inconsistently Deleted Roles** - Set of roles that have been deleted from the identity store.
   - **Missing Link Roles** - Set of roles that are missing the link to the parent base role.

   Figure 10–7 illustrates a portion of the Preview Roles dialog with the category Valid Roles expanded.

Figure 10–7 Previewing Roles, Five Categories



Roles in this group either have been just created or previously gener

Data Role Name	Display Name
BEN_BENEFITS_ADMINISTRATOR_JOB:Financial Mgmt-FM	Benefits Administrator:Finan...

To run a template, proceed as follows:

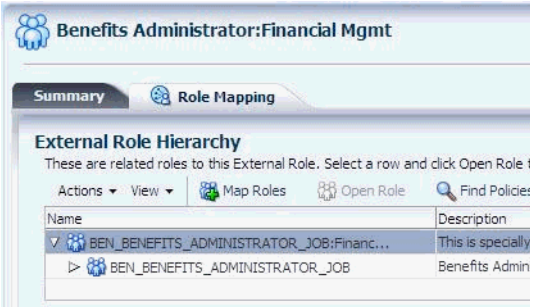
1. Open the template and bring the **Summary** tab to the foreground (this tab is available since the template has been saved).
2. Click the button **Generate Roles**. The roles generated are displayed in the five disjoint categories mentioned in the preceding procedure. Each external role generated by the run inherits from the corresponding parent external role.
3. Reconcile roles in the following four categories, as appropriate:
   - Invalid Roles - A role in this category is a role for which the base role is not found in the identity store. Delete or allow roles in this set; deleting an invalid role:
     - Removes the role, if it is not being used by any policy.
     - Removes the data security generated for the role.
   - Inconsistently Created Roles - A role in this category is a role with a name identical the name of some other role already in the identity store. Typically, these roles show up because of a change or deletion in records from where the dimensions are computed. Delete or reuse roles in this set; reusing an inconsistently created role:
     - Overwrites the existing role with the generated one.
     - Adds a link between the base role and the role.
     - Refreshes the role’s display name and description.
     - Adds the data security for the role.
     - Does not affect data securities defined by other templates.
   - Inconsistently Deleted Roles - Delete or recreate roles in this set; recreating an inconsistently deleted role:
     - Creates the role in the identity store using the template’s naming definition.
     - Adds the data security for the role.

  - Adds a link between the base role and the role, if it was not already in place.

- Missing Link Roles - A role in this category is missing the required link to a base role. Relink roles in this set; relinking a missing link:
  - Adds a link between the base role and the role.
  - Updates the grant associated with the role.

Once external roles and data policy grants have been generated, you can verify that they have been properly created by searching and opening a particular role or policy. Figure 10–8 illustrates how the generated external role *Benefits Administrator:Finalcial Mgnt* inherits, as expected, from the base external role *Benefits Administrator* (the names displayed in the External Role Hierarchy table are the role display names, not role names):

Figure 10–8 Generated Role Inheriting from a Based Role



Benefits Administrator:Financial Mgmt

External Role Hierarchy

These are related roles to this External Role. Select a row and click Open Role t

Name	Description
▽ BEN_BENEFITS_ADMINISTRATOR_JOB:Financ...	This is specially
▷ BEN_BENEFITS_ADMINISTRATOR_JOB	Benefits Admin

10.4.1 Running Templates Programmatically

The following two functions support running a single template or the collection of templates with a given group id via web-services:

```
public String executeTemplate(String TemplateName)
public String executeTemplateByGroupId(String GroupId)
```

The string returned by either of them describes the status of the run. If successful, it identifies the template(s) that were run; otherwise, it identifies the error that was encountered.

10.5 Updating a Template

There are rigorous restrictions on how a template can be changed once it has been run.

- The name of a template cannot be updated.
- The SQL that defines the template dimensions cannot be changed. The data that this SQL accesses, however, can change and, therefore, a new template run may return a different set of dimensions than those returned by the last run.
- When a dimension is added (to the set of dimensions of the last run), the template run creates external roles for the added dimension only.
- When a dimension is deleted (from the set of dimensions of the last run), the administrator can either deactivate the external roles involving the deleted dimension or left them unchanged.
- After execution, the template’s naming cannot be updated.

On the other hand, external roles can be added or deleted from a template at any time.

- When an external role is added to a template, a template run creates external roles for the added role and each of the dimensions.
- When an external role is deleted from a template, then the administrator can either deactivate the external roles involving the deleted role or left them unchanged.

Use the following procedure to update a template.

1. Locate the template to update by performing a regular search or an advanced search.

   Data Role Templates can be searched by specifying a template name, display name, and group id.

   a. Select **Global** > **Role Templates** in the navigation panel and click Open (the folder icon on top of the panel) to display the Search - Role Templates page.

   b. Enter an operator and a string to match for the template name, an operator and a string to match for the template display name, and an operator and a string to match for the template group id.

   c. Click Search to trigger the search and to display the templates that match the entered specification in the Search Results table.

   d. Double-click an item in the Search Results table to open it.

      Alternately, select the template in the Search Results table and click Open.
2. Click **Edit** to open the template for editing in the Home area.
3. Modify fields as appropriate and as allowed in the page tabs.
4. Click **Apply** to save changes.

10.6 Importing and Exporting a Template

A data role template can be imported to or exported from the Oracle Authorization Policy Manager environment with the use of the following two utilities: `importMetadata` and `exportMetadata`. Both utilities require establishing a connection to the Oracle WebLogic server before they can be used. The following code illustrates how to establish a connection to a WebLogic server:

```
> connect ('aUser','aPassword','t5://localhost:7133')
```

In the code, the first argument is the user name, the second is the password for that user, and the third is the connection URL to the server. The connection so established is terminated with the command `exit()`.

Use the following procedure to import one or more data role templates.

1. Connect to the server.
2. Execute the utility `importMetadata`, as illustrated in the following sample (the arguments are listed in different lines only for clarity of exposition):

   ```
   > importMetadata(application='oracle.security.apm',
                   server='AdminServer',
                   fromLocation='/myLocation/myRoleTemplates',
                   docs='/oracle/apps/apm/**',
                   restrictCustTo='site')
   ```

   The meaning of the arguments is as follows:

- `application` specifies the owner of the data role template to be imported.
- `server` specifies the name of the WebLogic server to which one is connected.
- `fromLocation` specifies the directory where the data role template to be imported is located.
- `docs` specifies the template in the directory `fromLocation` to be imported. To import all templates (including template subdirectories) in the specified directory, use **, as illustrated in the example above.
- `restrictCustTo` is an argument that should always be set to `site`.

To export a data role template, proceed as follows:

1. Connect to the server.
2. Execute the utility `exportMetadata`, as illustrated in the following sample (the arguments are listed in different lines only for clarity of exposition):

   ```
   > exportMetadata(application='oracle.security.apm',
                    server='AdminServer',
                    toLocation='/myLocation/myRoleTemplates',
                    docs='/oracle/apps/apm/**',
                    restrictCustTo='site')
   ```

   The meaning of the arguments is identical to those used for importing, except for `toLocation`, which specifies the location where the data role template(s) should be downloaded.

11 Managing Oracle Fusion Applications Data Security Policies

The information in this chapter is specific to Oracle Fusion Applications only.

This chapter describes the procedures an administrator follows to manage security policies on database resources, and it is divided into the following sections:

- Database Resources and Policies Overview
- Searching Database Resources and Policies
- Managing Database Resources
- Managing Data Security Policies

11.1 Database Resources and Policies Overview

Data security policies determine who can do what on which set of data. A data security policy defines the enterprise or application roles that provides members of those roles access rights to specific data.

The Oracle Fusion security reference implementation provides enterprise roles and access to data through a comprehensive set of predefined data security policies applicable to application roles that are members of those enterprise roles.

Data security consists of privileges conditionally granted to a role, and these grants are used to control access to instance sets of a business object. A privilege is a single action corresponding to an operation on a single business object. Instance sets are rows of a database resource returned by a user-defined SQL WHERE clause; instance sets may be a single row of data, multiple rows of a single table, or all rows of a single table. A data security policy is, therefore, a set of privileges to a principal on a business object for a given instance set.

The security administrator uses Oracle Authorization Policy Manager to create and administer data security policies. A data security policy involves the following security artifacts:

- A database resource that references a primary key corresponding to the database table or view of the business object to be secured
- A role that has been provisioned with the users who can perform the granted actions
- A rule (also known as a condition) to define the available row instances in the form of an SQL predicate or simple filter (stored as XML) defined on the rows of the database resource

- One or more actions (such as view, edit, or delete) performed on database records that correspond to the operations supported by the business object, and which may include custom operations

By default, when a business object is registered as a database resource in Authorization Policy Manager, users are denied access to all data of that business object. A data security policy makes data available to users based on the roles they are members of according to the actions and conditions specified in the policy:

- Actions determine whether the user has the right to perform a given operation.
- Condition evaluation for actions (and their corresponding operations) specify the set of rows on which those operations can be applied.

11.1.1 Prerequisites and Best Practices for Creating Data Security Policies

Data security policies secure the database resources of an enterprise. The Oracle Fusion security reference implementation provides a comprehensive set of predefined data security policies for database resources that involve database tables and views that correspond to business objects; it is recommended that these database resources not be changed.

In cases where custom database resources must be secured, the security administrator can manage the predefined database resources or create new data security policies. Before modifying any data security policy, it is important to understand the predefined data security policies provided by the security reference implementation. As a general guideline, security policies assigned to duty roles of the reference implementation should not be changed, only their participation in role hierarchies. Details about the Oracle Fusion security reference implementation can be found in the Oracle Fusion Security Reference Manual and are also available for review Authorization Policy Manager.

Important: Review but do not modify data security policies from the Oracle Fusion security reference implementation in Authorization Policy Manager except as a custom implementation to create data security policies.

For example, in the security reference implementation, the `IT Security Manager` job role hierarchy includes the `Application Data Security Administration Duty` duty role, which is entitled to manage data security policies (the entitlement is `Manage Data Security Policy`). This entitlement provides necessary privileges to perform the **Manage Data Security Policies** task in Authorization Policy Manager.

Before creating a data security policy with Authorization Policy Manager, the security administrator should collect the following information for custom security policies:

- The actions corresponding to the operations that the business object to be secured defines; these actions can be obtained from the developer who implemented the business function.
- The primary key of the database table or view that the business object represents; this key can be obtained from the developer who implemented the business function.
- The application roles for which the policy is created; these roles can be obtained from an Oracle Identity Manager administrator, or they can be queried with Authorization Policy Manager.

11.1.2 Process Overview for Creating Data Security Policies

To define an Oracle Fusion Data Security policy, proceed as follows:

1. Identify the business object that you want to secure and register its backend database table or view as a database resource.

   A table or view is registered by its primary key columns.

2. Identify and define all of the conditions that you want to make available on the registered database resource.

   Conditions define an instance set of rows specified either by simple filters (XML defined) or complex SQL queries whose values can be parameterized. No condition definition is needed in the case of a single row instance or all row instances.

3. Identify and register the actions that you want to secure for this database resource.

   Action names should match the names of the operations the business object supports (for example, view_US_ONLY, edit_US_ONLY, delete_US_ONLY for custom operations).

4. Identify the Oracle Platform Security Services (OPSS) role for which you want to create the policy.

   OPSS roles and the role inheritance hierarchy are managed by Oracle Identity Manager.

5. Define a rule to specify the values (data) that you want to make available on the registered database resource for a particular role.

   A rule can be a row instance of the database resource (when a single value is desired), the entire resource (when all values are desired), or a condition that had been defined for the resource (when multiple values are needed).

6. Grant one or more actions on the database resource to the role for the specified rule.

   Available actions will be limited to the actions that had been defined for the database resource.

11.2 Searching Database Resources and Policies

Data security policies are displayed in Authorization Policy Manager by the database resource they secure, as explained in the following sections:

- Searching Database Resources
- Locating Policies Associated with a Database Resource

11.2.1 Searching Database Resources

Database resources can be queried with a simple or an advanced search.

To specify a simple search, proceed as follows:

1. Select **Global** from the pull-down list at the top of the navigation panel.
2. Select **Database Resources** from the pull-down list second from the top.
3. Enter a string to match in the text box, possibly using the wildcard characters % or * (the wild character matches any character in the pattern).

The search returns all names and display names of database resources that match the specified string; leave this box empty to obtain the list of all objects of the specified type.

4. Click the Go button to trigger the search and to display the results in the tab **Search Results**, which is automatically brought to the foreground when the search is completed. Positioning the cursor on the blue information button next to an item displays the item details. The Search Results tab shows at most the first 200 matches found by the search.

5. Once an item is selected in the Search Results, it can be opened or edited by clicking **Open** or **Edit** at the top of the table.

To specify a database resource advanced search, proceed as follows:

1. Expand the hierarchy it to expose all nodes in the hierarchy.

2. Double-click **Database Resources** under the Global node to display the **Manage Database Resources and Policies** tab.

3. In the Search area of that tab, enter the query parameters as follows:

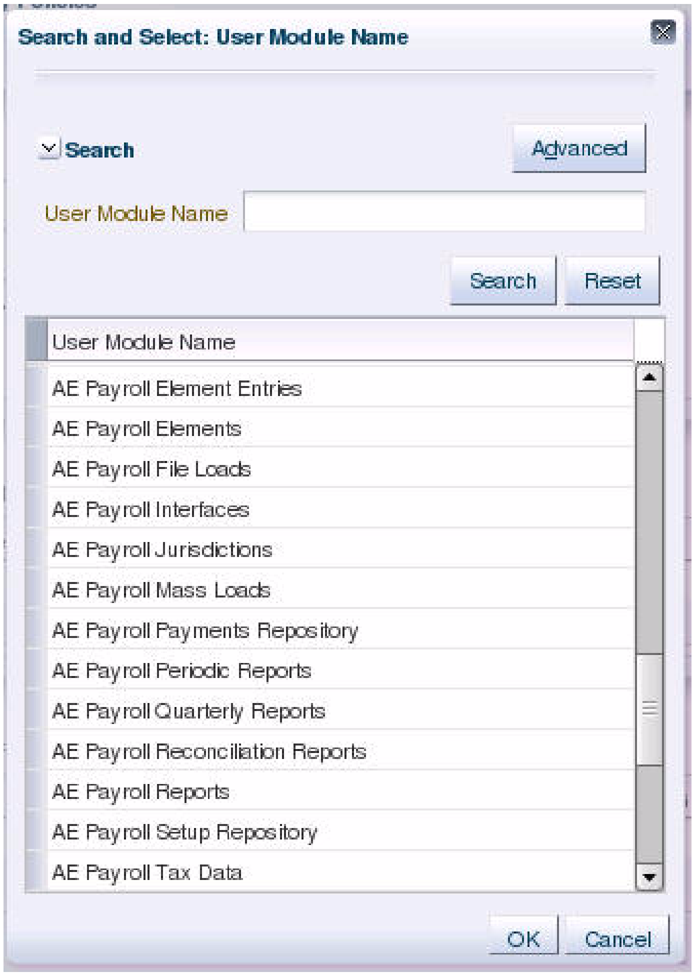
   - In the **Object Name** box, enter the string to match the database resource name, possibly including the wildcard characters % or * (the wild character matches any character in the pattern). To match all strings, leave the box empty.

   - In the **Display Name** box, enter the string to match the database resource display name, possibly including the wildcard characters % or * (the wild character matches any character in the pattern). To match all strings, leave the box empty.

   - In the **User Module Name** pull-down box, select a module where to look for database resources. To locate a module, optionally select the Search item (at the bottom of the pull-down list) to bring up the **Search and Select: User Module Name** dialog, illustrated in Figure 11–1. In **User Module Name** box in that dialog, enter the string to match module and select one from the result list returned and then click OK.

   - Optionally, click **Reset** to set the parameter values to the values they had before you entered the current values.

Figure 11–1 Searching for a User Module



Search and Select: User Module Name

User Module Name
AE Payroll Element Entries
AE Payroll Elements
AE Payroll File Loads
AE Payroll Interfaces
AE Payroll Jurisdictions
AE Payroll Mass Loads
AE Payroll Payments Repository
AE Payroll Periodic Reports
AE Payroll Quarterly Reports
AE Payroll Reconciliation Reports
AE Payroll Reports
AE Payroll Setup Repository
AE Payroll Tax Data

4. Optionally, click **Save...** to save the current query parameters. The name of the saved collection then appears in the pull-down list **Saved Search**. Selecting a saved search from this pull-down list fills in the query parameters automatically with the saved values.

5. Click **Search** to trigger the search. All database resources matching the entered specifications are displayed in the table **Search Results**.

   The actions at the top of this table allow:

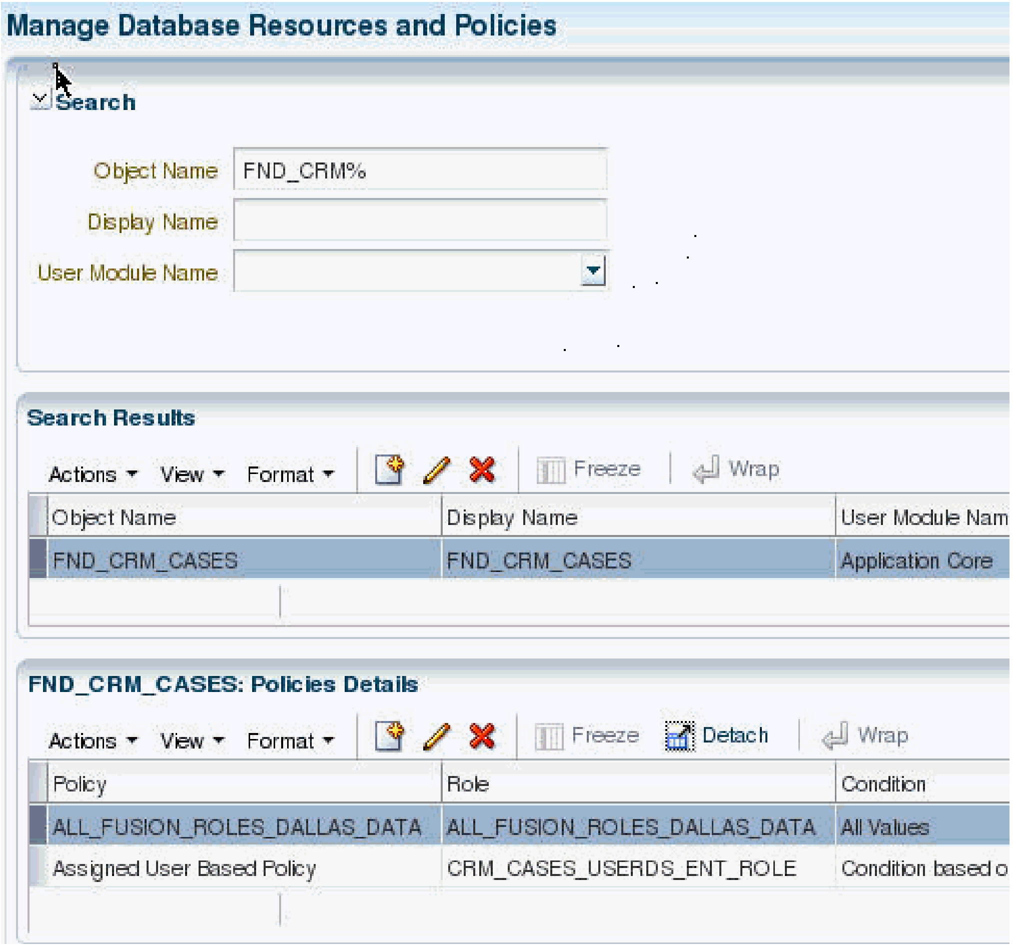
   - Creating a database resource
   - Editing a database resource
   - Deleting a database resource

   In addition, the table below the Search Results area, displays the list of policies associated with a database resource selected in the Search Results table.

Figure 11–2 illustrates the results of an advanced search on database resources and the policies associated with a database resource selected from the Search Results table.

The button Follow is used send out notifications about the activity stream. For details about Activity Stream, see *Oracle Fusion Applications Developer's Guide*.

Figure 11–2 Manage Database Resources and Policies Tab



Object Name: FND_CRM%

Display Name:

User Module Name:

Object Name	Display Name	User Module Nam
FND_CRM_CASES	FND_CRM_CASES	Application Core

Policy	Role	Condition
ALL_FUSION_ROLES_DALLAS_DATA	ALL_FUSION_ROLES_DALLAS_DATA	All Values
Assigned User Based Policy	CRM_CASES_USERDS_ENT_ROLE	Condition based o

11.2.2 Locating Policies Associated with a Database Resource

To locate a policy (global or associated with an application) associated with a database resource, first identify the database resource with a search as described in section Searching Database Resources, and then inspect the list of policies associated with the database resource in the **Policies Details** table. Figure 11–2 illustrates the policies associated with a database resource FND_CRM_CASES.

For alternative ways to locate policies, see Chapter 4, "Searching for Security Objects."

11.3 Managing Database Resources

The following sections describe how to specify what portion of the database resource is secured by a data security policy:

- Specifying Database Resource Column Details
- Managing Database Resource Conditions
- Managing Database Resource Actions

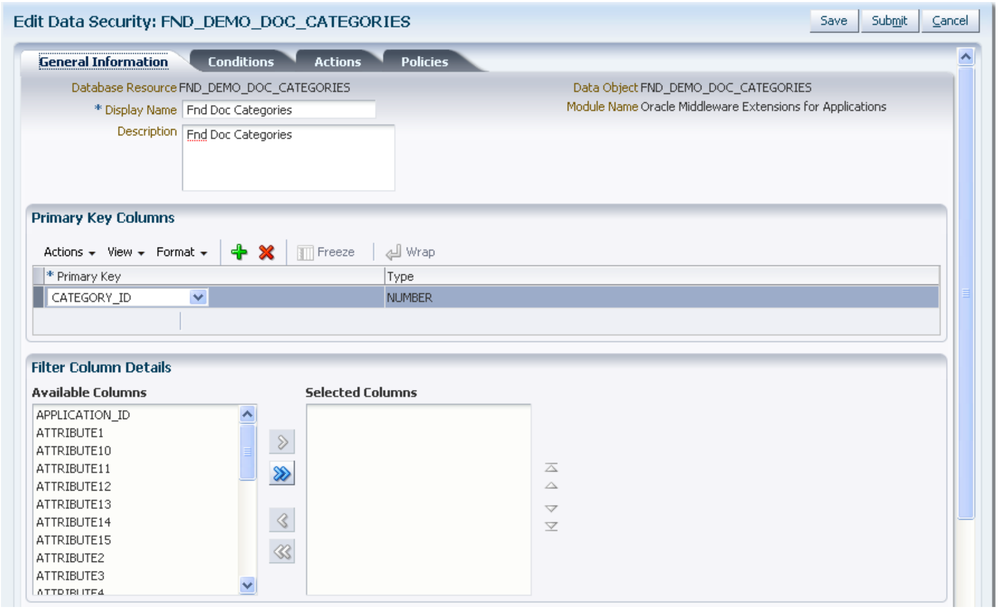
11.3.1 Specifying Database Resource Column Details

The following sections describe how to manage the available columns of a database resource for which security policies may be defined:

- Specifying the Primary Key Columns of the Policy's Database Resource
- Filtering Columns of the Policy's Database Resource

Figure 11–3 illustrates the **General Information** tab in the Edit Data Security page after the FND_DOC_CATEGORIES table has been registered as a database resource with the primary key CATEGORY_ID and no columns filtered.

Figure 11–3 Creating a Database Resource - Specifying the Primary Key Columns



Edit Data Security: FND_DEMO_DOC_CATEGORIES

Database Resource FND_DEMO_DOC_CATEGORIES

* Display Name Fnd Doc Categories

Description Fnd Doc Categories

Data Object FND_DEMO_DOC_CATEGORIES

Module Name Oracle Middleware Extensions for Applications

Primary Key Columns

* Primary Key	Type
CATEGORY_ID	NUMBER

Filter Column Details

Available Columns

APPLICATION_ID
ATTRIBUTE1
ATTRIBUTE10
ATTRIBUTE11
ATTRIBUTE12
ATTRIBUTE13
ATTRIBUTE14
ATTRIBUTE15
ATTRIBUTE2
ATTRIBUTE3
ATTRIBUTE4

Selected Columns

11.3.1.1 Specifying the Primary Key Columns of the Policy's Database Resource

The database resource is a database table or view. You use the table or view's primary key column(s) to register it as a database resource.

To specify the primary key of the database resource, proceed as follows:

1. Identify the database resource by matching the name of the database resource that the policy will secure. For details, see Section 11.2.1, "Searching Database Resources."
2. In the **General Information** tab, click **Add** and choose the database resource's primary key from the dropdown list. You can add additional key columns when more than one key column is defined by the resource.
3. Click **Save** to complete the specification of the primary key.

11.3.1.2 Filtering Columns of the Policy's Database Resource

You can filter columns at the level of the database resource when you want to exclude columns from the row instance sets defined by data security policies. Additionally, the data from filtered columns will not be accessible by the user.

To filter the list of columns that the database resource defines, proceed as follows:

1. Identify the database resource by matching the name of the database resource that the policy will secure. For details, see Section 11.2.1, "Searching Database Resources."
2. In the **General Information** tab, move available columns to the **Selected Column** list when you want to exclude that column from the database resource.

   Excluded columns will not be available when defining database resource conditions and the data of these columns will not be accessible to any user.
3. Click **Save** to complete the filtering the list of columns.

11.3.2 Managing Database Resource Conditions

You define conditions on the database resource to specify what portions of the database resource may be secured by data security policies. A condition is a group of row instances that are determined by a simple XML filter or an SQL predicate (`WHERE` clause) that queries the attributes of the resource itself. Conditions are always defined on a single table or view.

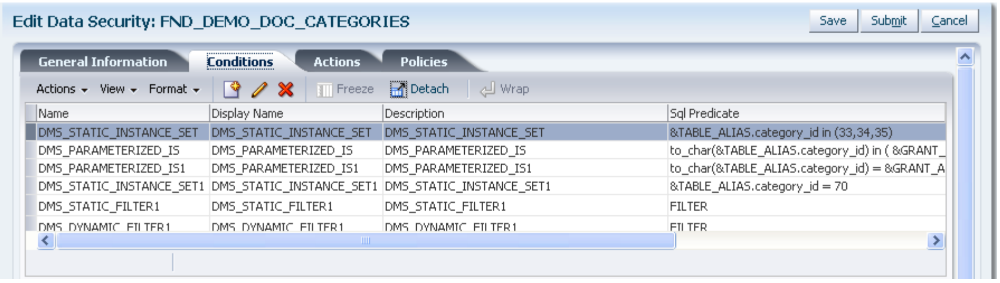
You can define a condition to specify multiple row instance sets using a parameterized SQL WHERE clause. For example, the condition may be defined by the predicate `REGION=&PARAM` where the parameter `PARAM` is associated with different regions. When an action is granted for a condition, it may be done for a particular value of the parameter, such as a "sales manager" in the West region may have an action granted for a **Region** condition with the parameter value `West`.

You do not need to define a condition for single row instance condition (single value) or for all row instances conditions (all values). Both the single-value case and the all-values case may be easily defined when you create the data security policy. Internally, Oracle Authorization Policy Manager will save these as conditions with the appropriate SQL query clause.

Figure 11–4 illustrates the **Conditions** tab in the Edit Data Security page after several row instance sets have been defined as conditions of the database resource. You can perform these operations using the **Conditions** tab:

- Click **New** to define a new condition.
- Select an existing condition and click **Edit** to edit the condition details.
- Select an existing condition and click **Delete** to delete the condition.

Figure 11–4 Creating a Database Resource - Adding to the Available Conditions List



Edit Data Security: FND_DEMO_DOC_CATEGORIES

Name	Display Name	Description	Sql Predicate
DMS_STATIC_INSTANCE_SET	DMS_STATIC_INSTANCE_SET	DMS_STATIC_INSTANCE_SET	&TABLE_ALIAS.category_id in (33,34,35)
DMS_PARAMETERIZED_IS	DMS_PARAMETERIZED_IS	DMS_PARAMETERIZED_IS	to_char(&TABLE_ALIAS.category_id) in (&GRANT_
DMS_PARAMETERIZED_IS1	DMS_PARAMETERIZED_IS1	DMS_PARAMETERIZED_IS1	to_char(&TABLE_ALIAS.category_id) = &GRANT_A
DMS_STATIC_INSTANCE_SET1	DMS_STATIC_INSTANCE_SET1	DMS_STATIC_INSTANCE_SET1	&TABLE_ALIAS.category_id = 70
DMS_STATIC_FILTER1	DMS_STATIC_FILTER1	DMS_STATIC_FILTER1	FILTER
DMS_DYNAMIC_FILTER1	DMS_DYNAMIC_FILTER1	DMS_DYNAMIC_FILTER1	FILTER

To define a new database resource condition, proceed as follows:

1. Identify the database resource to secure by matching the name of the database resource that the policy secures. For details, see Section 11.2.1, "Searching Database Resources."

2. In the **Conditions** tab, click **New** and define what portions of the database resource may be secured by data security policies. For details, see Section 11.3.2, "Managing Database Resource Conditions."

3. In the **Create Database Resource Condition** dialog, enter the following information:

   - A name (required)
   - A display name (required)
   - A description (optional)
   - A condition type (required)
     - When you what to use the attribute tree picker user interface to define a simple condition, choose **Filter**.
     - When you know the attributes names of your condition and you want to define an SQL WHERE clause, for example to specify a dynamic condition, using a parameterized SQL predicate, choose **SQL Predicate**.

4. If you chose a **Filter** condition type, then define the condition as follows:

   a. Click **Add** and choose the column name from the dropdown list that you want to define the filter on.

   b. Choose the tree operator for the selected column.

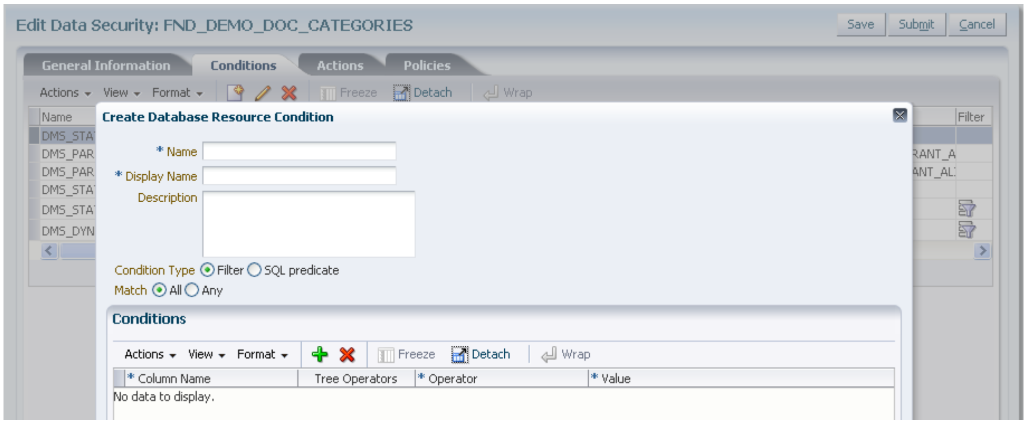
   c. Enter a value as the test for the operator.

   d. Add additional columns as needed.

   e. Select **Match All** or **Match Any** depending on whether you want the filter conditions to be ANDed (match all) or ORed (match any).

   Figure 11–5 illustrates the **Create Database Resources Condition** dialog in the Edit Data Security page when creating an XML filter condition.

Figure 11–5 Creating a Database Resource Condition - Defining an XML Filter Condition



Edit Data Security: FND_DEMO_DOC_CATEGORIES

Create Database Resource Condition

* Name

* Display Name

Description

Condition Type ● Filter ○ SQL predicate

Match ● All ○ Any

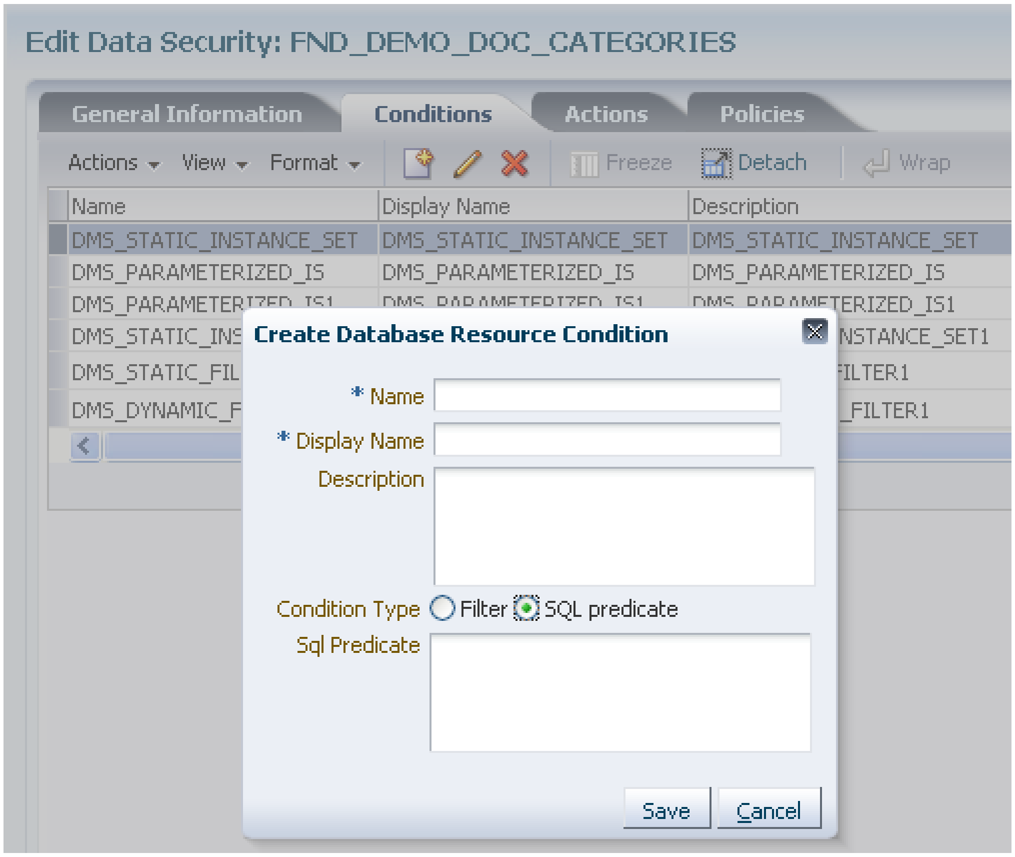
Conditions

* Column Name	Tree Operators	* Operator	* Value
No data to display.			

5. If you chose a **SQL Predicate** condition type, then enter the SQL predicate consisting of a query on the table or view named by the database resource.

   Figure 11–6 illustrates the **Create Database Resources Condition** dialog in the Edit Data Security page where you enter an SQL predicate condition.

Figure 11–6 Creating a Database Resource Condition



Name	Display Name	Description
DMS_STATIC_INSTANCE_SET	DMS_STATIC_INSTANCE_SET	DMS_STATIC_INSTANCE_SET
DMS_PARAMETERIZED_IS	DMS_PARAMETERIZED_IS	DMS_PARAMETERIZED_IS
DMS_PARAMETERIZED_IS1	DMS_PARAMETERIZED_IS1	DMS_PARAMETERIZED_IS1
DMS_STATIC_INS		NSTANCE_SET1
DMS_STATIC_FIL		FILTER1
DMS_DYNAMIC_F		_FILTER1

Create Database Resource Condition

* Name

* Display Name

Description

Condition Type ○ Filter ● SQL predicate

Sql Predicate

6. Click **Save** to complete the creation of a database source condition.

11.3.3 Managing Database Resource Actions

You define actions on the database resource to specify what kind of access data security policies will secure on a business object. For example, you can specify whether a user might have read, update, or delete access by naming actions for each of these and granting them in a data security policy to a particular role.

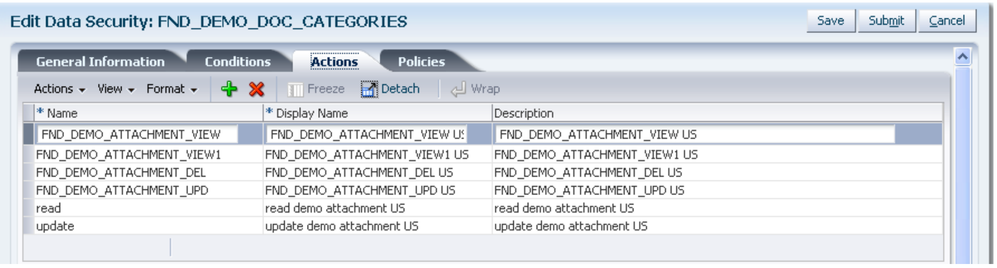
An action corresponds one to one with an operation that the business object implements. Action names must match the corresponding business object operation names established by the business object developer. Actions may correspond to either standard operations or custom operations. For example, a business object might define custom read operations based on the regions West and East, which allows you to create the corresponding actions **read_WEST** and **read_EAST**. Alternatively, actions that you define, such as **read** and **update**, may correspond to the standard read and update operations of the same business object, when no region is specified.

Actions act on the row instance sets of the database resource conditions that you define in a data security policy. When the user invokes an operation on the business object, the system will act on the row set instances defined by the condition and the corresponding action of the security policy in effect for that business object. The system will perform the operation only if the policy grants the user a privilege for the corresponding action.

Figure 11–7 illustrates the **Actions** tab in the Edit Data Security page after several actions have been defined on the database resource. You can perform these operations using the **Actions** tab:

- Click **Add** to define a new action.
- Click in any field of an existing action and edit the details; do not change the name of an action unless the name of the corresponding business object operation should be changed too.
- Select an existing action and click **Delete** to delete the action.

Figure 11–7 Creating a Database Resource - Adding to the Available Actions List



Edit Data Security: FND_DEMO_DOC_CATEGORIES

* Name	* Display Name	Description
FND_DEMO_ATTACHMENT_VIEW	FND_DEMO_ATTACHMENT_VIEW U:	FND_DEMO_ATTACHMENT_VIEW US
FND_DEMO_ATTACHMENT_VIEW1	FND_DEMO_ATTACHMENT_VIEW1 US	FND_DEMO_ATTACHMENT_VIEW1 US
FND_DEMO_ATTACHMENT_DEL	FND_DEMO_ATTACHMENT_DEL US	FND_DEMO_ATTACHMENT_DEL US
FND_DEMO_ATTACHMENT_UPD	FND_DEMO_ATTACHMENT_UPD US	FND_DEMO_ATTACHMENT_UPD US
read	read demo attachment US	read demo attachment US
update	update demo attachment US	update demo attachment US

To define a new database resource action, proceed as follows:

1. Identify the database resource to secure by matching the name of the database resource that the policy secures. For details, see Section 11.2.1, "Searching Database Resources."
2. In the **Actions** tab, click **New** and enter the following information in the list of actions table:
   - An action name (required) - the name must match the corresponding operation of the business object. When defining actions for custom operations, consult the developer for the names of the operations.
   - A display name (required)
   - A description (optional)
3. Click **Save**.

11.4 Managing Data Security Policies

The following sections describe how to determine the roles that can access a database resource and the type of actions that those roles may perform on the data:

- Creating a Data Security Policy
- Modifying a Custom Data Security Policy

11.4.1 Creating a Data Security Policy

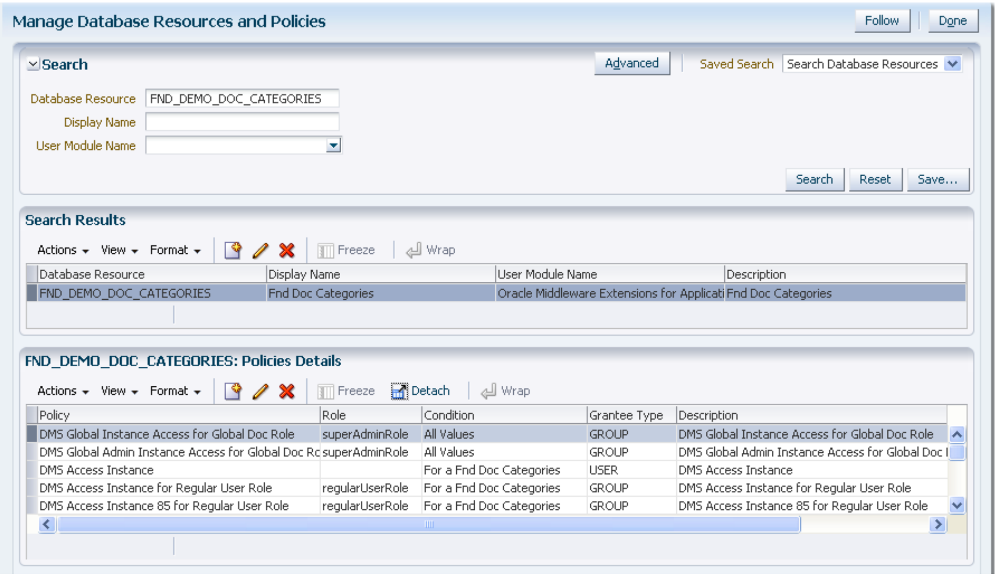
When you register a new business object as a database resource, users will initially be prevented from invoking the operations of the business object. They will also be prevented from accessing the data of the resource. You define data security policies to make data of a custom business object available to the users of the application.

Figure 11–8 illustrates the **Policies Details** tab in the Manage Database Resources and Polices page after several data security policies have been created for the database resource. You can perform these operations using the **Policies** tab:

- Click **New** to define a new policy.
- Select an existing policy and click **Edit** to edit the details in the **Details** tab.
- Select an existing policy and click **Delete** to delete the policy.

  Important: Duty roles in security policies in the Oracle Fusion reference implementation should not be edited or deleted; only their role hierarchies should be modified.

Figure 11–8 Creating a Data Security Policy - Adding to the Policy List



Manage Database Resources and Policies

Database Resource: FND_DEMO_DOC_CATEGORIES
Display Name:
User Module Name:

Database Resource	Display Name	User Module Name	Description
FND_DEMO_DOC_CATEGORIES	Fnd Doc Categories	Oracle Middleware Extensions for Applicati	Fnd Doc Categories

FND_DEMO_DOC_CATEGORIES: Policies Details

Policy	Role	Condition	Grantee Type	Description
DMS Global Instance Access for Global Doc Role	superAdminRole	All Values	GROUP	DMS Global Instance Access for Global Doc Role
DMS Global Admin Instance Access for Global Doc Rc	superAdminRole	All Values	GROUP	DMS Global Admin Instance Access for Global Doc I
DMS Access Instance		For a Fnd Doc Categories	USER	DMS Access Instance
DMS Access Instance for Regular User Role	regularUserRole	For a Fnd Doc Categories	GROUP	DMS Access Instance for Regular User Role
DMS Access Instance 85 for Regular User Role	regularUserRole	For a Fnd Doc Categories	GROUP	DMS Access Instance 85 for Regular User Role

Before you begin

Before you create a data security policy, perform the following tasks:

1. Register the business object as a database resource, as described in Section 11.3, "Managing Database Resources."
2. Define the conditions that you want to apply for specific actions of the policy, as described in Section 11.3.2, "Managing Database Resource Conditions." Conditions determine the row instance set available to a user for a given operation.
3. Define the actions to grant to the role, as described in Section 11.3.3, "Managing Database Resource Actions." Actions correspond to the operations of the business object that the user may invoke.
4. Obtain the name of the application role or enterprise role for which you want to create the policy.

To create a new data security policy, proceed as follows:

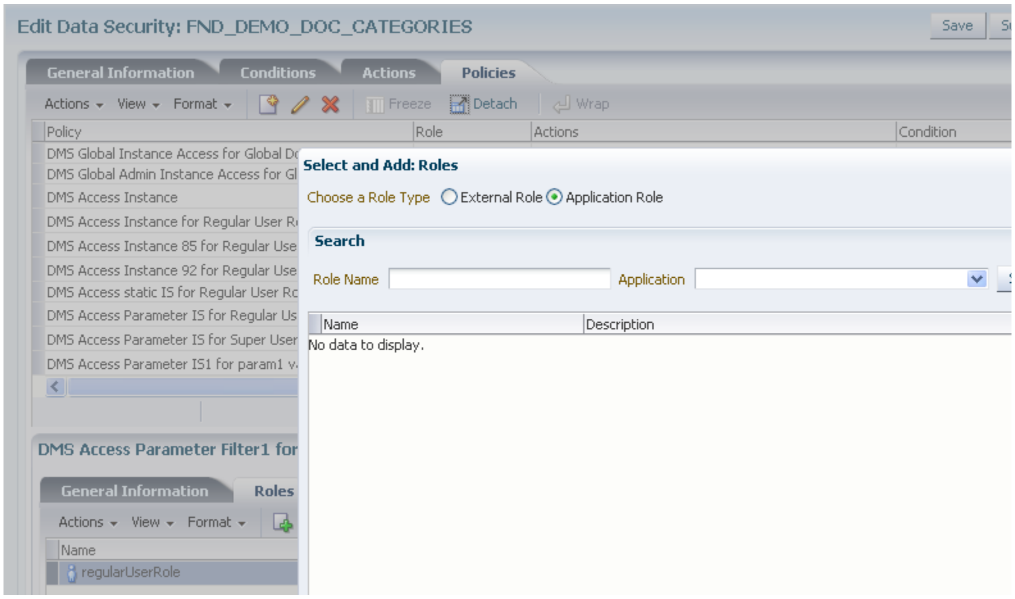
1. Identify the database resource to secure by matching the name of the database resource that the policy secures. For details, see Section 11.2.1, "Searching Database Resources."
2. In the **Policies** tab, click **New**.
3. In the **General Information** tab of the **Details** section, enter the following information for the data security policy being created:
   - A name (required)

- A module (required)
- A start date for the policy to become effective (required)
- An end date for the policy to cease to be effective (optional)
- A description (optional)

4. In the **Roles** tab of the **Details** section, select the role to which the policy grants access. The roles you add entitle all users assigned to those roles with access to the data.

   Figure 11–9 illustrates the **Select and Add: Roles** dialog in the Edit Data Security page.

Figure 11–9 Creating a Data Security Policy, Selecting a Role



Policy	Role	Actions	Condition
DMS Global Instance Access for Global Dc			
DMS Global Admin Instance Access for Gl			
DMS Access Instance			
DMS Access Instance for Regular User Ri			
DMS Access Instance 85 for Regular Use			
DMS Access Instance 92 for Regular Use			
DMS Access static IS for Regular User Rc			
DMS Access Parameter IS for Regular Us			
DMS Access Parameter IS for Super User			
DMS Access Parameter IS1 for param1 v.			

Choose a Role Type ○ External Role ● Application Role

Role Name Application

Name	Description

No data to display.

Name
regularUserRole

5. In the **Rule** tab of the **Details** section, specify the rows of the database resource on which the policy applies in the following ways:

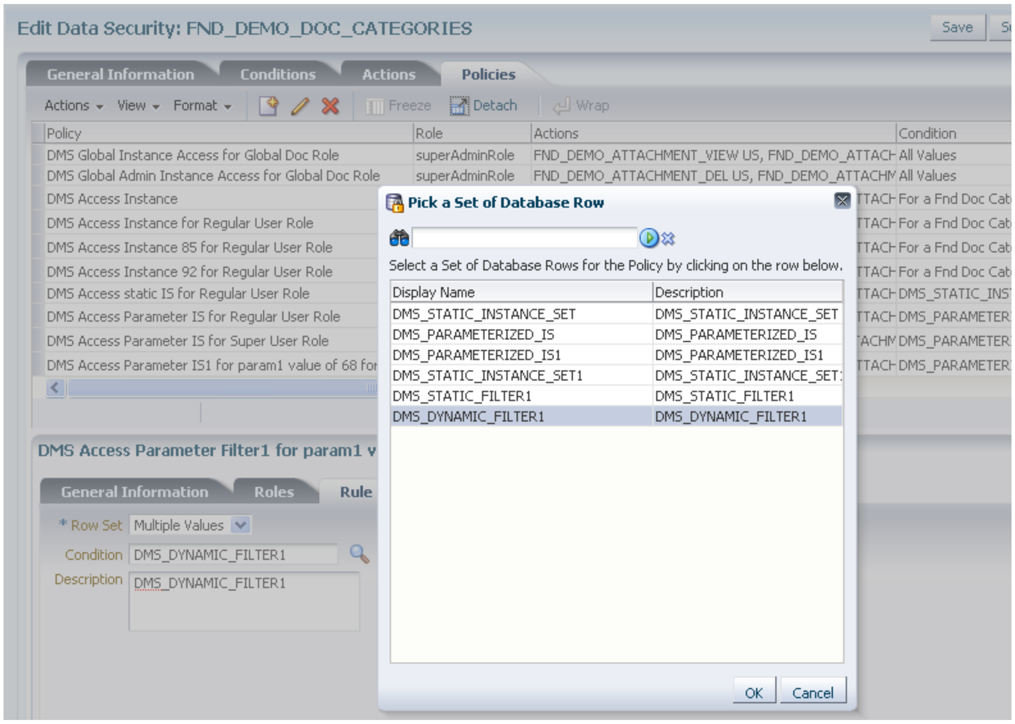
   – When you want to secure a specific row, select **Single Value**.

   – When you want to secure all rows, select **All Values**.

   – When you want to change the condition in order to change the secured rows of the database resource, select **Multiple Values** and click the **Search** icon and choose the desired condition. To create a new condition, see Section 11.3.2, "Managing Database Resource Conditions."

   Figure 11–10 illustrates the **Pick a Set of Database Row** dialog in the Edit Data Security page after several conditions have been selected from the list of available conditions.

Figure 11–10 Creating a Data Security Policy, Selecting Database Row



Policy	Role	Actions	Condition
DMS Global Instance Access for Global Doc Role	superAdminRole	FND_DEMO_ATTACHMENT_VIEW US, FND_DEMO_ATTACH	All Values
DMS Global Admin Instance Access for Global Doc Role	superAdminRole	FND_DEMO_ATTACHMENT_DEL US, FND_DEMO_ATTACHM	All Values
DMS Access Instance		TTACH	For a Fnd Doc Cat
DMS Access Instance for Regular User Role		TTACH	For a Fnd Doc Cat
DMS Access Instance 85 for Regular User Role		TTACH	For a Fnd Doc Cat
DMS Access Instance 92 for Regular User Role		TTACH	For a Fnd Doc Cat
DMS Access static IS for Regular User Role		TTACH	DMS_STATIC_INS
DMS Access Parameter IS for Regular User Role		TTACH	DMS_PARAMETER
DMS Access Parameter IS for Super User Role		ACHM	DMS_PARAMETER
DMS Access Parameter IS1 for param1 value of 68 for		TTACH	DMS_PARAMETER

DMS Access Parameter Filter1 for param1 v

* Row Set: Multiple Values

Condition: DMS_DYNAMIC_FILTER1

Description: DMS_DYNAMIC_FILTER1

Pick a Set of Database Row

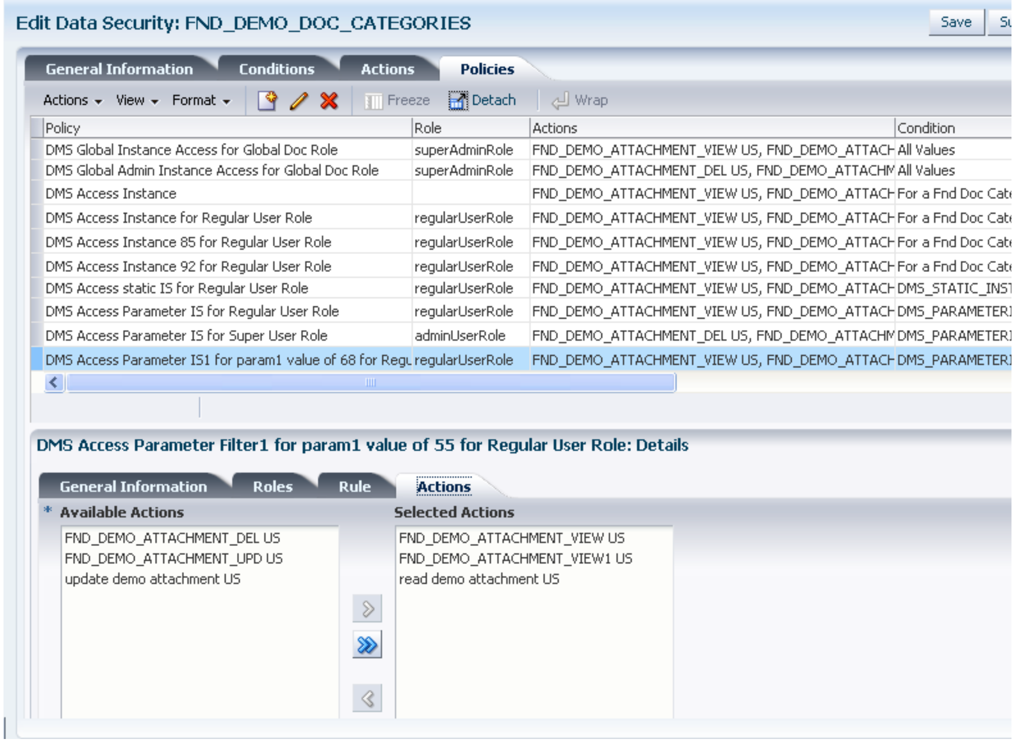
Select a Set of Database Rows for the Policy by clicking on the row below.

Display Name	Description
DMS_STATIC_INSTANCE_SET	DMS_STATIC_INSTANCE_SET
DMS_PARAMETERIZED_IS	DMS_PARAMETERIZED_IS
DMS_PARAMETERIZED_IS1	DMS_PARAMETERIZED_IS1
DMS_STATIC_INSTANCE_SET1	DMS_STATIC_INSTANCE_SET:
DMS_STATIC_FILTER1	DMS_STATIC_FILTER1
DMS_DYNAMIC_FILTER1	DMS_DYNAMIC_FILTER1

6. In the **Action** tab of the **Details** section, click **New** and specify what kind of access data security policies will secure on the database resource. For details, see Section 11.3.3, "Managing Database Resource Actions."

   Figure 11–11 illustrates the **Actions** tab in the Edit Data Security page after several actions have been selected.

Figure 11–11 Creating a Data Security Policy, Selecting Actions



Edit Data Security: FND_DEMO_DOC_CATEGORIES

Policy	Role	Actions	Condition
DMS Global Instance Access for Global Doc Role	superAdminRole	FND_DEMO_ATTACHMENT_VIEW US, FND_DEMO_ATTACH	All Values
DMS Global Admin Instance Access for Global Doc Role	superAdminRole	FND_DEMO_ATTACHMENT_DEL US, FND_DEMO_ATTACHM	All Values
DMS Access Instance		FND_DEMO_ATTACHMENT_VIEW US, FND_DEMO_ATTACH	For a Fnd Doc Cat
DMS Access Instance for Regular User Role	regularUserRole	FND_DEMO_ATTACHMENT_VIEW US, FND_DEMO_ATTACH	For a Fnd Doc Cat
DMS Access Instance 85 for Regular User Role	regularUserRole	FND_DEMO_ATTACHMENT_VIEW US, FND_DEMO_ATTACH	For a Fnd Doc Cat
DMS Access Instance 92 for Regular User Role	regularUserRole	FND_DEMO_ATTACHMENT_VIEW US, FND_DEMO_ATTACH	For a Fnd Doc Cat
DMS Access static IS for Regular User Role	regularUserRole	FND_DEMO_ATTACHMENT_VIEW US, FND_DEMO_ATTACH	DMS_STATIC_INST
DMS Access Parameter IS for Regular User Role	regularUserRole	FND_DEMO_ATTACHMENT_VIEW US, FND_DEMO_ATTACH	DMS_PARAMETERI
DMS Access Parameter IS for Super User Role	adminUserRole	FND_DEMO_ATTACHMENT_DEL US, FND_DEMO_ATTACHM	DMS_PARAMETERI
DMS Access Parameter IS1 for param1 value of 68 for Regu	regularUserRole	FND_DEMO_ATTACHMENT_VIEW US, FND_DEMO_ATTACH	DMS_PARAMETERI

DMS Access Parameter Filter1 for param1 value of 55 for Regular User Role: Details

* Available Actions	Selected Actions
FND_DEMO_ATTACHMENT_DEL US	FND_DEMO_ATTACHMENT_VIEW US
FND_DEMO_ATTACHMENT_UPD US	FND_DEMO_ATTACHMENT_VIEW1 US
update demo attachment US	read demo attachment US

7. Click **Save** to compete the creation of the data security policy.

11.4.2 Modifying a Custom Data Security Policy

Data security policies provided in the Oracle Fusion security reference implementation can be viewed but it is recommended that they not be modified; other data security policies, that is, those created with Oracle Authorization Policy Manager, can be modified.

To modify a data security policy, proceed as follows:

1. Identify the data security policy to modify or view in either of the following ways:

   - By matching the name of the policy. For details, see Section 11.2.2, "Locating Policies Associated with a Database Resource."

   - By matching the name of the database resource that the policy secures. For details, see Section 11.2.1, "Searching Database Resources."

2. In the **Policies** tab, select the policy to modify from the **Policy** list and modify the following details for the data security policy:

   a. In the **General Information** tab, you can modify the policy start and end dates, as well as change the name of the policy and its description.

   b. In the **Roles** tab, you can change the roles to which the policy grants access. You can add a new role to the policy when you want to entitle all users who belong to that role with access to the data. You can also remove an existing role from the policy.

   c. In the **Rule** tab, you can change the rows of the database resource on which the policy applies in the following ways:

- When you want to secure a specific row, select **Single Value**.
- When you want to secure all rows, select **All Values**.
- When you want to change the condition in order to change the secured rows of the database resource, select **Multiple Values** and click the **Search** icon and choose the desired condition. To create a new condition, see Section 11.3.2, "Managing Database Resource Conditions."

d. In the **Actions** tab, you can change the actions on the database resource's records secured by the policy. To create a new action, see Section 11.3.3, "Managing Database Resource Actions."

3. Click **Save** to complete the modification of the data security policy.

12 Managing System Configurations

Security Module definitions and administrator configurations are defined within the top-level System Configuration tab in the Authorization Policy Manager Administration Console. This chapter contains the following topics:

- Delegating With Administrators
- Configuring Security Module Definitions

12.1 Delegating With Administrators

Administrator Roles can be created to delegate management operations for policy objects. For example, Application and Policy Domain delegating administrators can be defined by creating an Administrator Role at the appropriate level and assigning the role Administration Privileges as well as a user, group, or another role. See Chapter 6, "Delegating With Administrator Roles" for more information. It includes a section on creating System Administrator Roles which can manage other types of Administrator Roles in any Application or Policy Domain.

12.2 Configuring Security Module Definitions

A Security Module is an Oracle Entitlements Server client that plays a key role in authorization. After an authorization request is generated, the Security Module evaluates policy data to determine if access to the resource will be granted or denied. An Application (the Oracle Entitlements Server object that represents the protected resource) must be *bound* to the Security Module that protects it. Binding Security Modules enables policy data to be transmitted to it for evaluation. The Policy Distribution Component (discussed in Chapter 9, "Managing Policy Distribution") is the mechanism used to transmit policy data to the Security Modules.

Note: For more information about the authorization process, see Section 2.2, "How Oracle Entitlements Server Evaluates Policies."

The following sections document how to bind (and unbind) Security Module definitions to (and from) Application objects.

- Section 12.2.1, "Creating a Security Module Definition"
- Section 12.2.2, "Binding an Application to a Security Module"
- Section 12.2.3, "Unbinding an Application From a Security Module"
- Section 12.2.4, "Deleting a Security Module Definition"

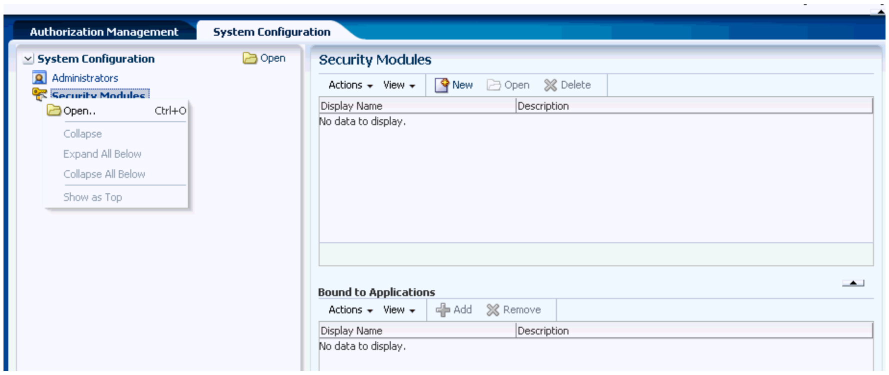
12.2.1 Creating a Security Module Definition

To create a security module, proceed as follows.

1. Select the **System Configuration** tab from the Home area.
2. Double-click **Security Modules** in the Navigation Panel.

   Alternately, right-click Security Modules and select Open. The Security Modules page is displayed as in Figure 12–1.

Figure 12–1 Security Modules in Home Area



3. Click **New** to create a new Security Module definition.

   Alternately, select **New** from the Actions menu. The Security Module dialog is displayed.
4. Provide the following values for the new Security Module.
   - **Name**: The entry must be a unique.
   - **Display Name**
   - **Description**
5. Click **Save**.

12.2.2 Binding an Application to a Security Module

To bind an Application to a Security Module, proceed as follows.

1. Select the **System Configuration** tab from the Home area.
2. Double-click **Security Modules** in the Navigation Panel.

   Alternately, right-click Security Modules and select Open. The Security Modules page is displayed.
3. Select the name of the Security Module definition from the table.
4. Click **Add** in the Bound to Applications table, or select **Add** from the **Actions** menu.

   Alternately, select Add from the Bound to Applications Actions menu. The Add Applications dialog displays.
5. Enter a search string in the text box and click the arrow to search.

   Alternately, click the arrow with no search string to return all available Applications.

6. Select one or more applications from the list returned.
7. Click **Add**.

   The selected applications are bound to the selected Security Module and displayed in the Bound to Applications table.

12.2.3 Unbinding an Application From a Security Module

To unbind an application from a Security Module, proceed as follows.

1. Select the **System Configuration** tab from the Home area.
2. Double-click **Security Modules** in the Navigation Panel.

   Alternately, right-click Security Modules and select Open. The Security Modules page is displayed.
3. Select the name of the applicable Security Module definition in the table.
4. Select the name of the applicable Application in the Bound to Applications table.
5. Click **Remove** or select **Remove** from the Actions menu.

   A confirmation dialog is displayed.
6. Click Unbind.

12.2.4 Deleting a Security Module Definition

To remove a Security Module definition, proceed as follows.

1. Select the **System Configuration** tab from the Home area.
2. Double-click **Security Modules** in the Navigation Panel.

   Alternately, right-click Security Modules and select Open. The Security Modules page is displayed.
3. Select the name of the applicable Security Module definition in the table.
4. Click **Delete** or select **Delete** from the Actions menu.

   A confirmation dialog is displayed.
5. Click Remove.

13 Management Tasks

This chapter contains information on several management tasks including configuring the cache, auditing, and migrating policies from different types of stores. It contains the following sections:

- Integrating with WebLogic Server
- Managing Audit Tasks
- Migrating Policies
- Configuring Cache
- Debugging

13.1 Integrating with WebLogic Server

WebLogic Server can automatically intercept authorization requests after enabling the Role Mapping and Authorization providers. The following procedure explains how to do this; it assumes the WebLogic Server is installed in the `$WLS` directory in the `$DOMAIN` domain. Replace the values from your installation when following the procedure.

1. Copy the `jps-atz-wls-proxyproviders.jar` to the WebLogic Server provider definition directory using the following command.

   ```
   cp jps-atz-wls-proxyproviders.jar $WLS/server/lib/mbeantypes
   ```

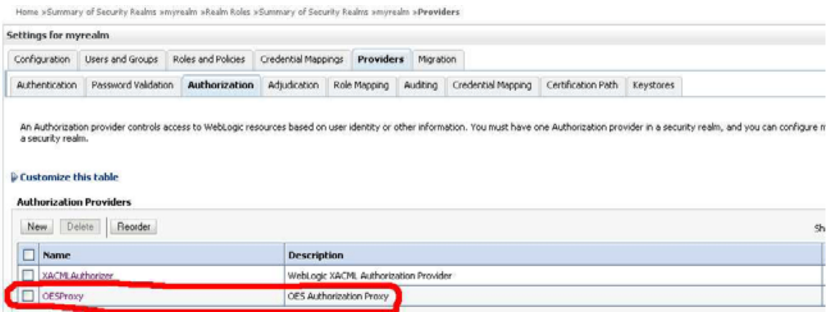
2. Start the `$DOMAIN` domain using the following command.

   ```
   $DOMAIN/startWeblogic.sh
   ```

3. Add the Authorization Proxy and Role Mapping providers to the realm that protects the domain.

   Figure 13–1 is a screenshot of the WebLogic Server console that illustrates this.

Figure 13–1 Adding Providers to the WebLogic Server Domain's Realm



An Authorization provider controls access to WebLogic resources based on user identity or other information. You must have one Authorization provider in a security realm, and you can configure m a security realm.

Authorization Providers

□	Name	Description
□	XACMLAuthorizer	WebLogic XACML Authorization Provider
□	OESProxy	OES Authorization Proxy

4. Restart the domain.

After enabling the providers, see Section C.2.4, "WebLogic Server Security Module" for the configuration parameters.

13.2 Managing Audit Tasks

Oracle Entitlements Server audits all administrative activities and authorization requests, optionally recording the information to a file. The auditing framework is based on the framework developed for Oracle Platform Security Services. An overview of the Oracle Platform Security Services auditing framework can be found in *Oracle Fusion Middleware Application Security Guide*. The following information is more specific to the auditing functionality in Oracle Entitlements Server.

- Section 13.2.1, "Auditing Events"
- Section 13.2.2, "Configuring Auditing"
- Section 13.2.3, "Additional Auditing Information"

> **Note:** Oracle Entitlements Server will audit decisions resulting from policies configured by itself, Oracle Platform Security Services or any combination thereof.

13.2.1 Auditing Events

Table 13–1 lists the events (organized by functional category) that are audited by Oracle Entitlements Server. Audit logging is disabled by default.

Table 13–1 Events Audited in Oracle Entitlements Server

Functional Category	Functional Task
Administration Role Management	■ AdminRoleCreation ■ AdminRoleDeletion ■ AdminRoleGrant ■ AdminRoleRevoke ■ AdminRoleResActionGrant ■ AdminRoleResActionRevoke
Application Management	■ ApplicationDeletion
Grant Management	■ PermissionSetGrant ■ PermissionSetRevocation

Table 13–1 (Cont.) Events Audited in Oracle Entitlements Server

Functional Category	Functional Task
PermissionSetManagement	■ PermissionSetCreation ■ PermissionSetModification ■ PermissionSetDeletion
PolicyDomainManagement	■ PolicyDomainCreation ■ PolicyDomainDeletion
PolicyManagement	■ PolicyCreation ■ PolicyModification ■ PolicyDeletion ■ PolicyGrant ■ PolicyRevoke
ResourceManagement	■ ResourceCreation ■ ResourceModification ■ ResourceDeletion
Role Management	■ RoleCreation ■ RoleModification ■ RoleDeletion ■ RoleMembershipAdd ■ RoleMembershipRemove
RolePolicyManagement	■ RolePolicyCreation ■ RolePolicyModification ■ RolePolicyDeletion
Authorization	■ CheckPermission ■ IsAccessAllowed ■ CheckSubject
ConfigurationBindingManagement	■ SecurityModuleBinding ■ SecurityModuleUnbinding
ConfigurationManagement	■ SecurityModuleCreation ■ SecurityModuleModification ■ SecurityModuleDeletion
PolicyDistributionManagement	■ PolicyDistribution ■ PdpDeregistration ■ purgeDistributionStatus

13.2.2 Configuring Auditing

Auditing is configured in `jps-config.xml`, the configuration file used by Java EE containers. It is located in the *$DOMAIN_HOME*`/config/fmwconfig` directory. You can define a filterPreset level, a repository type and other information as illustrated in Example 13–1.

Example 13–1 Audit Service Configuration Parameters in jps-config.xml

```
<!-- JPS Audit Service Instance-->
<serviceInstance name="audit" provider="audit.provider">
```

```
        <property name="audit.filterPreset" value="None"/>
        <property name="audit.maxDirSize" value ="0"/>
        <property name="audit.maxFileSize" value ="104857600"/>
        <property name="audit.loader.jndi" value="jdbc/AuditDB"/>
        <property name="audit.loader.interval" value="15" />
        <property name="audit.loader.repositoryType" value="File" />
    </serviceInstance>
```

Table 13–2 contains details about the configuration parameters.

Table 13–2 Auditing Parameters in jps-config.xml

Parameter	Description
audit.filterPreset	None (default), Low, Medium, All or Custom
audit.maxDirSize	Controls the size of the directory in which the audit files are written. Takes an integer in bytes.
audit.maxFileSize	Controls the size of the bus stop file in which audit events are written. Takes an integer in bytes.
audit.loader.jndi	When a database is in use, takes a path to the JNDI data source to which audit events are uploaded.
audit.loader.interval	When a database is in use, controls the frequency of the audit loader's upload. Takes an integer in seconds.
audit.loader.RepositoryType	Defines the audit repository type. Takes a value of **File** or **Db**. If type is database (Db), `audit.loader.jndi` must also be defined.

13.2.3 Additional Auditing Information

The following list collects chapter links in other documents with information regarding the auditing framework.

- Introductory material can be found in *Oracle Fusion Middleware Security Guide*.
- You can manage audit policies with the Enterprise Manager user interface or with the WebLogic Scripting Tool (WLST) command-line interface. See *Oracle Fusion Middleware Application Security Guide* for guidance.
- The Oracle Fusion Middleware Audit Framework Reference is in *Oracle Fusion Middleware Security Guide*.
- Additional configuration information is in *Oracle Fusion Middleware Security Guide*.

13.3 Migrating Policies

This section contains information regarding migrating policies from one type of store to another. It contains procedures for the following:

- Section 13.3.1, "Migrating From XML to LDAP"
- Section 13.3.2, "Migrating From LDAP to XML"
- Section 13.3.3, "Migrating From XML to Database"
- Section 13.3.4, "Migrating From Database to XML"

13.3.1 Migrating From XML to LDAP

Following is the procedure to migrate policies from an XML-based policy store to an LDAP-based directory.

1. Modify `jps-config.xml` as described in this sub procedure.

   a. Create a `serviceInstance` for both the source and destination policy stores as illustrated in Example 13–2.

Example 13–2 XML to LDAP serviceInstances for Source and Destination Policy Stores

```
<!-- Source XML-based policy store instance -->
   <serviceInstance name="src.xml" provider="policystore.xml.provider"
      location="mydir/jazn-data.xml">
    <description>File Based Policy Store Service Instance</description>
   </serviceInstance>

<!-- Destination LDAP-based policy store instance -->
   <serviceInstance provider="ldap.policystore.provider"
      name="policystore.ldap.destination">
    <description>Replace: A. myDestDomain and myDestRootName to appropriate
       values according to your destination LDAP directory structure;
       B. ldap://myDestHost.com:3060 with the URL and port
       number of your destination LDAP</description>
       <property value="OID" name="policystore.type"/>
       <property value="bootstrap" name="bootstrap.security.principal.key"/>
       <property value="cn=myDestDomain" name="oracle.security.jps.farm.name"/>
       <property value="cn=myDestRootName"
         name="oracle.security.jps.ldap.root.name"/>
       <property value="ldap://myDestHost.com:3060" name="ldap.url"/>
   </serviceInstance>
```

   b. Create a `serviceInstance` corresponding to the bootstrap credential used to access the destination LDAP directory as illustrated in Example 13–3.

Example 13–3 XML to LDAP serviceInstance for Bootstrap Credential

```
<!-- Bootstrap credentials to access destination LDAP -->
   <serviceInstance location="./bootstrap" provider="credstoressp"
       name="bootstrap.cred">
     <description>Replace location with the full path of the directory
      where the bootstrap file cwallet.sso is located;
      typically found in destinationDomain/config/fmwconfig/</description>
   </serviceInstance>
```

   c. Create a `jpsContext` for both source and destination stores as illustrated in Example 13–4.

Example 13–4 XML to LDAP jpsContext for Source and Destination Policy Stores

```
<jpsContext name="sourceContext">
   <serviceInstanceRef ref="src.xml"/>
</jpsContext>

<jpsContext name="destinationContext">
   <serviceInstanceRef ref="policystore.ldap.destination"/>
</jpsContext>

<jpsContext name="bootstrap_credstore_context">
   <serviceInstanceRef ref="bootstrap.cred"/>
</jpsContext>
```

2. Start the WebLogic Scripting Tool.

There is no need to connect the WebLogic Scripting Tool to the WebLogic Server as the migration command is an offline command.

3. Run the WebLogic Scripting Tool `migrateSecurityStore` command to migrate the policy store and application as follows.

   - To migrate the policy store, run:

     ```
     migrateSecurityStore
        (type="policyStore", src="sourceContext",
         dst="destinationContext",
         configFile="myDir/jps-config.xml")
     ```

     where the following applies:

     – The name of the corresponding `jpsContext` (previously created in Step 1) should be passed to the `src` and `dst` parameters.

     – The name of the jps-config.xml file (previously modified in Step 1) should be passed to the `configFile` parameter. The value must be a fully qualified file name with complete path information.

   - To migrate the application, run:

     ```
     migrateSecurityStore
        (type="appPolicies", src="sourceContext",
         dst="destinationContext",
         configFile="myDir/jps-config.xml",
         srcApp="sourceApplication", dstApp="destinationApplication",
         overWrite="true")
     ```

     where the following applies:

     – The name of the corresponding `jpsContext` (previously created in Step 1) should be passed to the `src` and `dst` parameters.

     – The name of the `jps-config.xml` file (previously modified in Step 1) should be passed to the `configFile` parameter. The value must be a fully qualified file name with complete path information.

     – The name of the application being migrated is the value of the `srcApp` parameter. If this parameter is not passed, all applications with the same application name in both the source and destination policy stores will be migrated.

     – The name that is assigned to the application in the destination policy store is the value of the `dstApp` parameter. If this parameter is not passed, the name of the application in the destination store is the same as the name used in the source store.

     – If the `overWrite` parameter is defined as true, policies specific to the destination application are replaced by policies from the source application. The default value of this parameter is false.

13.3.2 Migrating From LDAP to XML

Following is the procedure to migrate policies from an LDAP-based directory to an XML-based policy store.

1. Modify `jps-config.xml` as described in this sub procedure.

   a. Create a `serviceInstance` for both the source and destination policy stores as illustrated in Example 13–5.

Example 13–5 LDAP to XML serviceInstances for Source and Destination Policy Stores

```
<!-- Source LDAP-based policy store instance -->
   <serviceInstance provider="ldap.policystore.provider"
      name="policystore.ldap.source">
    <description></description>
    <property value="OID" name="policystore.type"/>
    <property value="bootstrap" name="bootstrap.security.principal.key"/>
    <property value="cn=mySourceDomain" name="oracle.security.jps.farm.name"/>
    <property value="cn=mySourceRootName"
      name="oracle.security.jps.ldap.root.name"/>
    <property value="ldap://mySourceHost.com:3060" name="ldap.url"/>
   </serviceInstance>

<!-- Destination XML-based policy store instance -->
   <serviceInstance name="dst.xml" provider="policystore.xml.provider"
    location="/scratch/divyasin/WithPSR/jazn-data-fscm.xml">
    <description>File Based Policy Store Service Instance</description>
   </serviceInstance>
```

b. Create a `serviceInstance` corresponding to the bootstrap credential used to access the destination LDAP directory as illustrated in Example 13–6.

Example 13–6 LDAP to XML serviceInstance for Bootstrap Credential

```
<!-- Bootstrap credentials to access source LDAP -->
   <serviceInstance location="./bootstrap" provider="credstoressp"
      name="bootstrap.cred">
   <description>Replace location with the full path of the directory where the
      bootstrap file cwallet.sso is located; typically found in
      destinationDomain/config/fmwconfig/</description>
   </serviceInstance>
```

c. Create a `jpsContext` for both source and destination stores as illustrated in Example 13–7.

Example 13–7 LDAP to XML jpsContext for Source and Destination Policy Stores

```
<jpsContext name="sourceContext">
   <serviceInstanceRef ref="policystore.ldap.source"/>
</jpsContext>

<jpsContext name="destinationContext">
   <serviceInstanceRef ref="dst.xml"/>
</jpsContext>

<jpsContext name="bootstrap_credstore_context">
   <serviceInstanceRef ref="bootstrap.cred"/>
</jpsContext>
```

2. Start the WebLogic Scripting Tool.

   There is no need to connect the WebLogic Scripting Tool to the WebLogic Server as the migration command is an offline command.

3. Run the WebLogic Scripting Tool `migrateSecurityStore` command to migrate the policy store and application as follows.

   - To migrate the policy store, run:

     ```
     migrateSecurityStore
        (type="policyStore", src="sourceContext",
     ```

```
    dst="destinationContext",
    configFile="/scratch/divyasin/WithPSR/jps-config.xml")
```

where the following applies:

– The name of the corresponding `jpsContext` (previously created in Step 1) should be passed to the `src` and `dst` parameters.

– The name of the jps-config.xml file (previously modified in Step 1) should be passed to the `configFile` parameter. The value must be a fully qualified file name with complete path information.

- To migrate the application, run:

```
migrateSecurityStore
   (type="appPolicies", src="sourceContext",
    dst="destinationContext",
    configFile="/scratch/divyasin/WithPSR/jps-config.xml",
    srcApp="sourceApplication", dstApp="destinationApplication",
    overWrite="true")
```

where the following applies:

– The name of the corresponding `jpsContext` (previously created in Step 1) should be passed to the `src` and `dst` parameters.

– The name of the `jps-config.xml` file (previously modified in Step 1) should be passed to the `configFile` parameter. The value must be a fully qualified file name with complete path information.

– The name of the application being migrated is the value of the `srcApp` parameter. If this parameter is not passed, all applications with the same application name in both the source and destination policy stores are migrated.

– The name that is assigned to the application in the destination policy store is the value of the `dstApp` parameter. If this parameter is not passed, the name of the application in the destination store is the same as the name used in the source store.

– If the `overWrite` parameter is defined as true, policies specific to the destination application are replaced by policies from the source application. The default value of this parameter is false.

13.3.3 Migrating From XML to Database

Following is the procedure to migrate policies from an XML-based policy store to a database.

1. Modify `jps-config.xml` as described in this sub procedure.

   a. Create a `serviceInstance` for both the source and destination policy stores as illustrated in Example 13–8.

Example 13–8 XML to Database serviceInstances for Source and Destination Policy Stores

```
<!-- Source XML-based policy store instance -->
   <serviceInstance name="src.xml" provider="policystore.xml.provider"
     location="/scratch/divyasin/WithPSR/jazn-data-fscm.xml">
    <description>File Based Policy Store Service Instance</description>
</serviceInstance>
```

```
<!-- Destination DB-based policy store instance -->
   <serviceInstance provider="ldap.policystore.provider"
     name="policystore.db.destination">
    <description>DB Based Policy Store Service Instance</description>
     <property name="policystore.type" value="DB_ORACLE"/>
     <property name="jdbc.url"
       value="jdbc:oracle:thin:@hostname.example.com:1722:orcl"
    <jdbc:oracle:thin:@hostname.example.com:1722:orcl>/>
     <property name="jdbc.driver" value="oracle.jdbc.driver.OracleDriver"/>
     <property name="bootstrap.security.principal.key"
        value="bootstrap_DWgpEJgXwhDIoLYVZ2OWd4R8wOA=" />
     <property name="oracle.security.jps.ldap.root.name" value="cn=jpsTestNode"/>
     <property name="oracle.security.jps.farm.name" value="cn=view_steph.atz"/>
   </serviceInstance>
```

b. Create a `serviceInstance` corresponding to the bootstrap credential used to access the destination LDAP directory as illustrated in Example 13–9.

Example 13–9 XML to Database serviceInstance for Bootstrap Credential

```
<!-- Bootstrap credentials to access source DB -->
   <serviceInstance location="./bootstrap" provider="credstoressp"
     name="bootstrap.cred">
     <description>Replace location with the full path of the directory
        where the bootstrap file cwallet.sso is located;
        typically found in destinationDomain/config/fmwconfig/</description>
   </serviceInstance>
```

c. Create a `jpsContext` for both source and destination stores as illustrated in Example 13–10.

Example 13–10 XML to Database jpsContext for Source and Destination Policy Stores

```
<jpsContext name="sourceContext">
   <serviceInstanceRef ref="src.xml"/>
</jpsContext>

<jpsContext name="destinationContext">
   <serviceInstanceRef ref="policystore.db.destination"/>
</jpsContext>

<jpsContext name="bootstrap_credstore_context">
   <serviceInstanceRef ref="bootstrap.cred"/>
</jpsContext>
```

2. Start the WebLogic Scripting Tool.

   There is no need to connect the WebLogic Scripting Tool to the WebLogic Server as the migration command is an offline command.

3. Run the WebLogic Scripting Tool `migrateSecurityStore` command to migrate the policy store and application as follows.

   - To migrate the policy store, run:

     ```
     migrateSecurityStore
        (type="policyStore", src="sourceContext",
         dst="destinationContext",
         configFile="/scratch/divyasin/WithPSR/jps-config.xml")
     ```

     where the following applies:

– The name of the corresponding jpsContext (previously created in Step 1) should be passed to the src and dst parameters.

– The name of the jps-config.xml file (previously modified in Step 1) should be passed to the configFile parameter. The value must be a fully qualified file name with complete path information.

- To migrate the application, run:

```
migrateSecurityStore
   (type="appPolicies", src="sourceContext",
    dst="destinationContext",
    configFile="/scratch/divyasin/WithPSR/jps-config.xml",
    srcApp="sourceApplication", dstApp="destinationApplication",
    overWrite="true")
```

where the following applies:

– The name of the corresponding jpsContext (previously created in Step 1) should be passed to the src and dst parameters.

– The name of the jps-config.xml file (previously modified in Step 1) should be passed to the configFile parameter. The value must be a fully qualified file name with complete path information.

– The name of the application being migrated is the value of the srcApp parameter. If this parameter is not passed, all applications with the same application name in both the source and destination policy stores are migrated.

– The name that is assigned to the application in the destination policy store is the value of the dstApp parameter. If this parameter is not passed, the name of the application in the destination store is the same as the name used in the source store.

– If the overWrite parameter is defined as true, policies specific to the destination application are replaced by policies from the source application. The default value of this parameter is false.

13.3.4 Migrating From Database to XML

Following is the procedure to migrate policies from a database to an XML-based policy store.

1. Modify jps-config.xml as described in this sub procedure.

   a. Create a serviceInstance for both the source and destination policy stores as illustrated in Example 13–11.

Example 13–11 Database to XML serviceInstances for Source and Destination Policy Stores

```
<!-- Source DB-based policy store instance -->
   <serviceInstance provider="ldap.policystore.provider"
     name="policystore.db.source">
    <description>DB Based Policy Store Service Instance</description>
    <property name="policystore.type" value="DB_ORACLE"/>
    <property name="jdbc.url"
      value="jdbc:oracle:thin:@hostname.example.com:1722:orcl"
<jdbc:oracle:thin:@hostname.example.com:1722:orcl>/>
    <property name="jdbc.driver" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="bootstrap.security.principal.key"
```

```
        value="bootstrap_DWgpEJgXwhDIoLYVZ2OWd4R8wOA=" />
    <property name="oracle.security.jps.ldap.root.name" value="cn=jpsTestNode"/>
    <property name="oracle.security.jps.farm.name" value="cn=view_steph.atz"/>
</serviceInstance>

<!-- Destination XML-based policy store instance -->
   <serviceInstance name="dst.xml" provider="policystore.xml.provider"
      location="/scratch/divyasin/WithPSR/jazn-data-fscm.xml">
    <description>File Based Policy Store Service Instance</description>
   </serviceInstance>
```

b. Create a `serviceInstance` corresponding to the bootstrap credential used to access the destination LDAP directory as illustrated in Example 13–12.

Example 13–12 Database to XML serviceInstance for Bootstrap Credential

```
<!-- Bootstrap credentials to access source and destination LDAPs -->
   <serviceInstance location="./bootstrap" provider="credstoressp"
      name="bootstrap.cred">
     <description>Replace location with the full path of the directory where
       the bootstrap file cwallet.sso is located; typically found in
       destinationDomain/config/fmwconfig/</description>
   </serviceInstance>
```

c. Create a `jpsContext` for both source and destination stores as illustrated in Example 13–13.

Example 13–13 Database to XML jpsContext for Source and Destination Policy Stores

```
<jpsContext name="sourceContext">
   <serviceInstanceRef ref="policystore.db.source"/>
</jpsContext>

<jpsContext name="destinationContext">
   <serviceInstanceRef ref="dst.xml"/>
</jpsContext>

<jpsContext name="bootstrap_credstore_context">
   <serviceInstanceRef ref="bootstrap.cred"/>
</jpsContext>
```

2. Start the WebLogic Scripting Tool.

   There is no need to connect the WebLogic Scripting Tool to the WebLogic Server as the migration command is an offline command.

3. Run the WebLogic Scripting Tool `migrateSecurityStore` command to migrate the policy store and application as follows.

   - To migrate the policy store, run:

     ```
     migrateSecurityStore
        (type="policyStore", src="sourceContext",
         dst="destinationContext",
         configFile="/scratch/divyasin/WithPSR/jps-config.xml")
     ```

     where the following applies:

     – The name of the corresponding `jpsContext` (previously created in Step 1) should be passed to the `src` and `dst` parameters.

  - The name of the jps-config.xml file (previously modified in Step 1) should be passed to the `configFile` parameter. The value must be a fully qualified file name with complete path information.

- To migrate the application, run:

```
migrateSecurityStore
   (type="appPolicies", src="sourceContext",
    dst="destinationContext",
    configFile="/scratch/divyasin/WithPSR/jps-config.xml",
    srcApp="sourceApplication", dstApp="destinationApplication",
    overWrite="true")
```

where the following applies:

  - The name of the corresponding `jpsContext` (previously created in Step 1) should be passed to the `src` and `dst` parameters.
  - The name of the `jps-config.xml` file (previously modified in Step 1) should be passed to the `configFile` parameter. The value must be a fully qualified file name with complete path information.
  - The name of the application being migrated is the value of the `srcApp` parameter. If this parameter is not passed, all applications with the same application name in both the source and destination policy stores are migrated.
  - The name that is assigned to the application in the destination policy store is the value of the `dstApp` parameter. If this parameter is not passed, the name of the application in the destination store will be the same as the name used in the source store.
  - If the `overWrite` parameter is defined as true, policies specific to the destination application are replaced by policies from the source application. The default value of this parameter is false.

13.4 Configuring Cache

Oracle Entitlements Server offers caching capabilities. The cache settings are configured in the `jps-config.xml` file. The following sections contain the appropriate information.

- Section 13.4.1, "Configuring Decision Caching"
- Section 13.4.2, "Configuring Attribute Caching"

13.4.1 Configuring Decision Caching

Authorization decision caching allows Oracle Entitlements Server to cache the result of an authorization call and use that decision in the future, if an identical call is made. The decision cache consists of two hierarchical levels.

- The first level (L1) caches subjects used in the authorization calls.
- The second level (L2) caches authorization and role mapping decisions for the given subject.

Note: The decision cache automatically invalidates itself if there is a change in the policy.

The key of the cache is the incoming Subject, Permission and attributes used during policy evaluation. The value of the cache is the decision and obligations.

All parameter names are prefixed with oracle.security.jps.pdp. Example 13–14 illustrates how the decision cache parameters might be set in jps-config.xml.

Example 13–14 XML To Configure Decision Caching

```
<serviceInstance name="pdp.service" provider="pdp.service.provider">
           …
 <property name="oracle.security.jps.pdp.AuthorizationDecisionCacheEnabled"
   value="true"/>
 <property name="oracle.security.jps.pdp.
   AuthorizationDecisionCacheEvictionCapacity"
   value="1000"/>
 <property name="oracle.security.jps.pdp.
   AuthorizationDecisionCacheEvictionPercentage"
   value="15"/>
 <property name=" oracle.security.jps.pdp.AuthorizationDecisionCacheTTL"
   value="180"/>
           …
</serviceInstance>
```

Table 13–3 documents the decision caching parameters.

Table 13–3 Decision Caching Parameters

Name	Description	Accepted Values
oracle.security.jps.pdp.AuthorizationDecisionCacheEnabled	Optional parameter that specifies whether the policy decision cache should be enabled.	true (default) false
oracle.security.jps.pdp.AuthorizaionDecisionCacheEvictionCapacity	Optional parameter that specifies the maximum capacity of the L1 cache. If the number of entries exceeds the value, some entries are evicted.	Integer representing number of entries 500 (default)
oracle.security.jps.pdp.AuthorizationDecisionCacheEvictionPercentage	Optional parameter that specifies the percentage of entries in L1 cache that have to be evicted when the maximum capacity has been reached. For example, if the maximum capacity is 200 and the value of this parameter is 10 then 20 entries are evicted from the cache.	Integer representing percent of entries 10 (default equals 10%)
oracle.security.jps.pdp.AuthotizationDecisionCacheTTL	Optional parameter that specifies a time-to-live value (in seconds) for entries in the L2 cache. It defines how long an authorization decision is cached.	Integer representing time in seconds 60 (default equals 1 minute)

13.4.2 Configuring Attribute Caching

Each passed attribute can be cached if the cached property is defined for it. A corresponding time-to-live (TTL) value must also be defined if cached is enabled. The key of the cache is the attribute URI. The value of the cache is the attribute object. Example 13–15 illustrates how the attribute cache might be set in jps-config.xml.

Example 13–15 XML To Configure Attribute Caching

```
<propertySet name="ootb.pip.attribute.age.based.on.myattr.rdbms">
    <property name="ootb.pip.attr.type" value="OOTB_PIP_ATTRIBUTE"/>
    <property name="ootb.pip.ref" value="pip.service.ootb.db"/>
    <property name="name" value=" myattr"/>
    <property name="query" value="select value from table"/>
    <property name="cached" value="true"/>
    <property name="TTL" value="60"/>
</propertySet>
```

Note: If `cached` is not defined for the attribute, it will not be cached.

13.5 Debugging

The following sections contain information on how to debug Authorization Policies created using Oracle Entitlements Server as well as the Policy Distribution Component.

- Section 13.5.1, "Configuring Logging for Debugging"
- Section 13.5.2, "Searching Logs to Debug Authorization Policies"
- Section 13.5.3, "Debugging Policy Distribution"

13.5.1 Configuring Logging for Debugging

Oracle Entitlements Server uses the standard Java logging framework. Logging is the process of notifying an entity of a particular event. In the case of Oracle Entitlements Server, the entity can be a file or the Administration Console, and the event can be debugging information, runtime exceptions, or a record of actions taken by a user. The logging framework is configured based on the Oracle Entitlements Server deployment. More information is in the following sections.

- Section 13.5.1.1, "Configuring Logging for a Java Security Module Deployment"
- Section 13.5.1.2, "Configuring Logging for a WebLogic Server Security Module Deployment"

Note: The `java.util.logging` package provides the classes and interfaces of the platform's core logging facilities.

13.5.1.1 Configuring Logging for a Java Security Module Deployment

The following configurations must be made to enable logging when using the Java Security Module in your deployment.

- Run the following command when you start the Security Module to specify the logging configuration file:

  ```
  -Djava.util.logging.config.file=logging.properties
  ```

- Set the logging level by adding the following lines to the configuration file:

  ```
  oracle.jps.authorization.level=FINEST
  oracle.jps.openaz.level=FINEST
  ```

  Logging levels define the complexity of the logging record and include (from least to most) VERBOSE (simple information), WARNING, INFO, CONFIG, FINE, FINER and FINEST (complex information).

If you don't specify a configuration file, the `logging.properties` file in *$JAVA_HOME*`/jre/lib/` is used. Example 13–16 illustrates how to configure `logging.properties` to log information to the Administration Console.

Example 13–16 Configuration for Administration Console Logging

```
#The messages will we printed to the standard output
handlers=java.util.logging.ConsoleHandler

#The default level for all loggers is INFO
.level=INFO

#Override the default level for OES authorization to FINEST
oracle.jps.authorization.level=FINEST
oracle.jps.openaz.level=FINEST

#Use default formatter to print the messages
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter
```

Example 13–17 illustrates how to configure `logging.properties` to log information to a file.

Example 13–17 Configuration for File Logging

```
#The messages will be written to a file
handlers=java.util.logging.FileHandler

#The default level for all loggers is INFO
.level=INFO

#Override the default level for OES authorization to FINEST
oracle.jps.authorization.level=FINEST
oracle.jps.openaz.level=FINEST

#Configure file information. %h - is the user home directory
java.util.logging.FileHandler.pattern = %h/java%u.log
java.util.logging.FileHandler.limit = 50000
java.util.logging.FileHandler.count = 1
java.util.logging.FileHandler.formatter = java.util.logging.SimpleFormatter
```

13.5.1.2 Configuring Logging for a WebLogic Server Security Module Deployment

To enable logging when using the WebLogic Server Security Module in your deployment, run the following command to specify the logging configuration file when you start the WebLogic Server domain.

```
startWeblogic.sh -Djava.util.logging.config.file=logging.properties
```

> **Tip:** If you specify a relative path, the base directory is the domain home - not the directory where `startWeblogic.sh` is located

Other configurations relevant to the WebLogic Server Security Module are similar to those defined in Section 13.5.1.1, "Configuring Logging for a Java Security Module Deployment."

13.5.2 Searching Logs to Debug Authorization Policies

The following sections explain how to search for information recorded to the logging file. They include the commands to be run and, in many sections, sample output.

- Section 13.5.2.1, "Searching for PEP Request Information"
- Section 13.5.2.2, "Searching for Security Module Cache Configuration Parameters"
- Section 13.5.2.3, "Searching for Principals"
- Section 13.5.2.4, "Searching for Resources and Actions"
- Section 13.5.2.5, "Searching for the Value of an Attribute"
- Section 13.5.2.6, "Searching for an Authorization Decision"
- Section 13.5.2.7, "Searching for the Value of an Obligation"
- Section 13.5.2.8, "Searching for Static Application Roles"

13.5.2.1 Searching for PEP Request Information

Run the following command against the logging file to output PEP Request related information (including the Authentic Identity, the Runtime Resource, the Runtime Action and the Application Context).

```
grep "PepRequestImpl"
```

13.5.2.2 Searching for Security Module Cache Configuration Parameters

Run the following command against the logging file to output the cache configuration parameters for a particular Security Module.

```
grep "AuthotizationDecisionCacheTTL"
```

The following properties may be returned for this search. If a property does not appear in the log, it is not specified in `jps-config.xml`. In cases like this, the default value of the property is used.

- `AuthorizationDecisionCacheTTL` defines the time-to-live (in seconds) for the Authorization Decision cache. The default value is 60.
- `AuthorizationDecisionCacheEvictionPercentage` defines the percentage of authorization decisions to drop when the Authorization Decision cache has reached maximum capacity. The default value is 10.
- `AuthorizationDecisionCacheEvictionCapacity` defines the number used to evict the Authorization Decision cache if the decision cache size reaches this size. The default value is 500.
- `AuthorizationDecisionCacheEnabled` specifies whether the Authorization Decision cache is enabled. The default value is *true*.

Example 13–18 illustrates output for this search.

Example 13–18 Sample Output for Cache Configuration Parameters Search

```
oracle.security.jps.az.internal.runtime.service.AbstractPDPService

FINE: properties : {
oracle.security.jps.pdp.AuthotizationDecisionCacheTTL=60,
oracle.security.jps.pdp.AuthorizationDecisionCacheEvictionPercentage=10,
oracle.secuirty.jps.pdp.AuthorizationDecisionCacheEvictionCapacity=1000,
oracle.security.jps.pdp.AuthorizationDecisionCacheEnabled=true}
```

13.5.2.3 Searching for Principals

Run the following command against the logging file to output the names of Principals that have been received by Oracle Entitlements Server in the form of an authorization request.

```
grep "Principal:"
```

Example 13–19 illustrates the output for a Principal search.

Example 13–19 Sample Output for Principal Search

```
com.bea.security.providers.authorization.asi.AuthorizationProviderImpl
isAccessAllowed

FINE:subject: Subject:
     Principal: John
     Principal: Employee
     Principal: Administrator
     Principal: Principal Developer
```

13.5.2.4 Searching for Resources and Actions

Run the following command against the logging file to output the Resources and Actions that have been received by Oracle Entitlements Server in the form of an authorization request.

```
grep "Resource:"
```

Example 13–20 illustrates how the information is returned. The defined values are the name of the policy object.

- Application = Lib
- Resource Type = libraryresourcetype
- Resource = Book
- Action = borrow

Example 13–20 Sample Output for Resource and Action Search

```
com.bea.security.providers.authorization.asi.AuthorizationProviderImpl
isAccessAllowed

FINE: Resource: resource=Lib/libraryresourcetype/Book, action=borrow
```

13.5.2.5 Searching for the Value of an Attribute

Run the following command against the logging file to output the value of an attribute that has been received by Oracle Entitlements Server in the form of an authorization request.

```
grep "<name-of-the-attribute>:"
EXAMPLE: grep "getAttributeInternal:"
```

Example 13–21 illustrates the returned information where the name of the attribute is `NumberOfBorrowedBooksAttribute` and the value is 2.

Example 13–21 Sample Output for the Value of an Attribute Search

```
com.bea.security.providers.authorization.asi.ARME.evaluator.EvalSession logDebug
FINE: getAttributeInternal: name: NumberOfBorrowedBooksAttribute; value: 2; type:
3
```

13.5.2.6 Searching for an Authorization Decision

Run the following command against the logging file to retrieve an authorization decision that has been stored.

```
grep "AccessResultLogger"
```

Example 13–22 illustrates the returned information and confirm that the authorization decision was affirmative.

Example 13–22 Sample Output for Authorization Decision Search

```
com.bea.security.providers.authorization.asi.AccessResultLogger log
FINE: Subject Subject:
Principal: John
Principal: Employee
Principal: Administrator
Principal: Principal Developer
 privilege borrow resource //app/policy/Lib/Book result PERMIT
```

13.5.2.7 Searching for the Value of an Obligation

Run the following command against the logging file to output the value of a specific obligation.

```
grep "adding response attribute:" | grep  "obligations"
```

Example 13–23 illustrates the returned information indicating that the obligation (named `DenyObligation`) denies the request when the amount of library books the Principal currently has checked out is more than three; in this case, the Principal has five books checked out.

Example 13–23 Sample Output for Obligation Value Search

```
com.bea.security.providers.authorization.asi.AuthorizationProviderImpl
ARMEisAccessAllowed
FINE: adding response attribute: namespace=oracle.security.oes.authorization.
 name=obligations value={DenyObligation=
   { reason_part1=Too many borrowed books (max=3), reason_part2=5, }}
```

13.5.2.8 Searching for Static Application Roles

Run the following command against the logging file to output the names of Application Roles granted statically.

```
grep "AbstractRoleManager" | grep "getGrantedStaticAppRoles"
```

Example 13–24 illustrates how two static roles are added to the list of principals: an authenticated-role – build-in role and Reader, an Application Role defined in the Application named Library.

Example 13–24 Sample Output for Static Role Search

```
oracle.security.jps.az.internal.runtime.entitymanager.AbstractRoleManager
getGrantedStaticAppRoles(Set)
FINER: RETURN [authenticated-role,
  ApplicationRoleLibrary/Readeruname:
  cn=Writer,cn=Roles,cn=Lib,cn=akapisni_dwps1_
  view1.atzsrg,cn=JPSContext,cn=jpsTestNode,guid:
  411EBF807CD411E0BF887FB1A0F3878F]
```

13.5.3 Debugging Policy Distribution

The Policy Distribution Component uses the policy management `Logger` interface. To enable debugging for the Policy Distribution Component, change the logging level of the `oracle.jps.policymgmt.level` property in the logging configuration file to FINEST. The procedure is documented in Section 13.5.1, "Configuring Logging for Debugging."

A

Using an OpenLDAP Identity Store

This appendix describes the special set up required in case the domain APM is running uses an OpenLDAP 2.2 identity store.

A.1 Using an OpenLDAP Identity Store

To use OpenLDAP 2.2 as a domain identity store with Authorization Policy Manager, proceed as follows:

1. Use the WebLogic Server administration console to create a new authenticator provider. For this new provider:
   - Select OpenLDAPAuthenticator from the list of authenticators.
   - Set the control flag of the OpenLDAPAuthenticator to SUFFICIENT.
   - Set the control flag of the DefaultAuthenticator to SUFFICIENT.
   - Change the order of authenticators to make the OpenLDAPAuthenticator the first in the list.
   - In the Provider Specific page for the OpenLDAPAuthenticator, enter User Base DN and Group Base DN, and set the value of the objectclass in the Group From Name Filter to something other than groupofnames.
2. From the Home directory of the OpenLDAP installation:
   - Open the file `slapd.conf` for edit.
   - In that file, insert the following line in the "include" section at the top:

     ```
     include ./schema/inetorgperson.schema
     ```

   - Save the file, and restart the OpenLDAP.

The above settings make possible adding the object class `inetorgperson` to every new external role you create in the OpenLDAP; this object class is required to map the external role to an application role.

B

Troubleshooting Oracle Authorization Policy Manager

This appendix describes common problems that you may encounter when configuring and using Authorization Policy Manager and explains how to solve them. It contains the following sections:

- Section B.1, "Unable to Login"
- Section B.2, "Need Further Help?"

B.1 Unable to Login

This section explains one of the reasons why logging in Authorization Policy Manager may fail.

Symptom

Authorization Policy Manager logging in fails and the system outputs a message that contains a line similar to the following:

```
Cannot obtain connection: driverURL = jdbc:weblogic:pool:mds-ApplicationMDSDB,
props = {EmulateTwoPhaseCommit=false, connectionPoolID=mds-ApplicationMDSDB,
jdbcTxDataSource=true, LoggingLastResource=false,
dataSourceName=mds-ApplicationMDSDB}.
```

Diagnosis

The above message indicates that Authorization Policy Manager cannot establish a connection with the database `mds-ApplicationMDSBD`. Authorization Policy Manager requires that this database be present for a successful logging in. For information on the databases required by Authorization Policy Manager, see Chapter 1, "Getting Started With Oracle Authorization Policy Manager."

Solution

Verify that referenced database is up, running, and available; then retry logging in.

B.2 Need Further Help?

You can find more solutions on My Oracle Support (formerly MetaLink) at `http://myoraclesupport.oracle.com`. If you do not find a solution to your problem, log a service request.

C

Configuration Parameters

This Appendix lists the parameters and accepted values that may be defined for Oracle Entitlements Server services using `jps-config.xml`, the configuration file used by Java EE containers. It is located in the *$DOMAIN_HOME*`/config/fmwconfig` directory. This Appendix is comprised of the following sections:

- Section C.1, "Policy Distribution Configuration"
- Section C.2, "Security Module Configuration"
- Section C.3, "PDP Proxy Configuration"
- Section C.4, "Policy Store Service Configuration"

C.1 Policy Distribution Configuration

The Policy Distribution Component is responsible for distributing policy objects and policies from the policy store to one or more Security Modules. It can distribute in a controlled-push mode, a controlled-pull mode and a non-controlled mode. Each mode entails different configurations.

- Section C.1.1, "Policy Distribution Component Server Configuration"
- Section C.1.2, "Policy Distribution Component Client Configuration"

C.1.1 Policy Distribution Component Server Configuration

Typically, configuration for the Policy Distribution Component (in a scenario when it runs within Oracle Entitlements Server) is associated with the Policy Store configuration in the `jps-config.xml` file to fetch policies and policy objects for distribution. Only in cases when data is pulled in a controlled manner (*controlled-pull mode*) is the Policy Distribution Component associated with the PDP Service configuration on the Security Module side. Table C–1 contains the configuration parameters.

Table C–1 Policy Distribution Server Configuration

Name	Information
oracle.security.jps.pd.server. transactionalScope	**Description:** Defines the scope of the policy distribution as either to one Security Module or to all Security Modules. If distribution fails when it involves only one Security Module, it does not affect distributions to other Security Modules. **Optional** **Accepted Values:** All (default), One

C.1.2 Policy Distribution Component Client Configuration

The Policy Distribution Component client is responsible for making policies available to the Security Module. Thus, the Policy Distribution Client configuration is always associated with the PDP Service configuration portion of the `jps-config.xml` file on the Security Module side. Configuration is different depending on the mode of distribution and the environment in which the Security Module is running. The following sections contain descriptions of the applicable configuration parameters.

- Section C.1.2.1, "Policy Distribution Component Client Java Standard Edition Configuration (Controlled Push Mode)"
- Section C.1.2.2, "Policy Distribution Component Client Java Enterprise Edition Container Configuration (Controlled Push Mode)"
- Section C.1.2.3, "Policy Distribution Client Configuration (Controlled Pull Mode)"
- Section C.1.2.4, "Policy Distribution Client Configuration (Non-controlled Mode)"

C.1.2.1 Policy Distribution Component Client Java Standard Edition Configuration (Controlled Push Mode)

Table C–2 compiles the parameters for the Policy Distribution Component client configuration when the Oracle Entitlements Server is running in a Java Standard Edition (JSE) environment and is configured to distribute data in the controlled-push mode.

Table C–2 Policy Distribution Client Configuration, JSE, Controlled Push Mode

Name	Information
oracle.security.jps.runtime.pd.client.policyDistributionMode	**Description:** Specifies the mode of policy distribution. *Controlled distribution* is initiated by the Policy Distribution Component, ensuring that the Security Module receives policy data that has been created or modified since the last distribution. **Mandatory** **Accepted Value:** controlled-push
oracle.security.jps.runtime.pd.client.sm_name	**Description:** Defines the name of the Security Module. **Mandatory** **Accepted Value:** Name of the Security Module
oracle.security.jps.runtime.pd.client.localpolicy.work_folder	**Description:** Defines the name of any directory in which local cache files are stored. This directory must have read and write privileges. **Optional** **Accepted Value:** The name of any directory in which local cache files will be stored. This directory must have read and write privileges.

Table C–2 (Cont.) Policy Distribution Client Configuration, JSE, Controlled Push Mode

Name	Information
oracle.security.jps.runtime.pd.client.incrementalDistribution	**Description:** Defines whether the distribution is incremental or flush. *Incremental distribution* is when only new and modified data is distributed. *Flush distribution* is when the Policy Distribution Component notifies the Security Module to cleanup locally stored policies in preparation for a complete re-distribution of all policy objects in the policy store. **Optional** **Accepted Values:** ■ false (policy distribution is flush for this Security Module) ■ true (default value; policy distribution is incremental for this Security Module if the required change logs are kept in the policy store)
oracle.security.jps.runtime.pd.client.registrationRetryInterval	**Description:** When a Security Module starts, it registers itself with the Policy Distribution Component to ensure the local policy cache is up to date. If registration fails, it will retry each time this interval of time passes until successful. **Optional** **Accepted Value:** time in seconds (default value is 5)
oracle.security.jps.runtime.pd.client.waitDistributionTime	**Description:** If this value is defined and not equal to zero, it specifies the amount of time that a Security Module will wait for initial policy distribution to happen. During this wait period, authorization requests are blocked until either the initial policy distribution completes or the configured period expires. **Optional** **Accepted Value:** time in seconds (default value is 60)
oracle.security.jps.runtime.pd.client.RegistrationServerURL	**Description:** Defines the URL of the Oracle Entitlements Server Administration Server. Used by the Security Module to register itself with Oracle Entitlements Server when it starts. **Mandatory** **Accepted Value:** URL
oracle.security.jps.runtime.pd.client.backupRegistrationServerURL	**Description:** Defines a backup URL for the Oracle Entitlements Server Administration Server. Used by the Security Module to register itself with Oracle Entitlements Server when it starts if the primary URL (parameter above) is unavailable. **Optional** (although if not configured Oracle Entitlements Server failover will not work) **Accepted Value:** URL
oracle.security.jps.runtime.pd.client.DistributionServicePort	**Description:** Defines the port to which a remote Policy Distributor will push policy updates. **Mandatory** **Accepted Value:** port number
oracle.security.jps.pd.client.sslMode	**Description:** Defines whether communication between the Policy Distribution Component server and client will use the Secure Sockets Layer (SSL) protocol or not. **Mandatory** **Accepted Values:** none, two-way (default value)

Table C–2 (Cont.) Policy Distribution Client Configuration, JSE, Controlled Push Mode

Name	Information
oracle.security.jps.pd.client.ssl.identityKeyStoreFileName	**Description:** Defines the name of the Identity Key Store file in which client certificates are stored. Used for SSL communication between the Security Module and the Policy Distribution Component. **Mandatory** **Accepted Value:** the name of the keystore file
oracle.security.jps.pd.client.ssl.trustKeyStoreFileName	**Description:** Defines the name of the Trust Key Store file where Certificate Authority (CA) certificates are stored. Used for SSL communication between the Security Module and the Policy Distribution Component. **Mandatory** **Accepted Value:** the name of the identity key store file
oracle.security.jps.pd.client.ssl.identityKeyStoreKeyAlias	**Description:** Defines an Identity Key alias to identify the client certificate used for SSL communication between the Security Module and the Policy Distribution Component. **Optional** (if only one alias exists in the identity keystore there is no need to specify this value) **Accepted Value:** the identity key alias
oracle.security.jps.runtime.pd.client.SMinstanceType	**Description:** Defines the type of Security Module to which the Policy Distribution Component client is connecting. **Mandatory** **Accepted Value:** java (Other accepted values include wls, RMI and ws. Because this table covers the Java Security Module only, the value must be java.)

C.1.2.2 Policy Distribution Component Client Java Enterprise Edition Container Configuration (Controlled Push Mode)

Table C–3 compiles the parameters for the Policy Distribution Component client configuration when the Oracle Entitlements Server is running in a Java Enterprise Edition (JEE) environment and is configured to distribute data in the controlled-push mode.

Table C–3 Policy Distribution Client Configuration, JEE, Controlled Push Mode

Name	Information
oracle.security.jps.runtime.pd.client.policyDistributionMode	**Description:** Specifies the mode of policy distribution. *Controlled distribution* is initiated by the Policy Distribution Component, ensuring that the Security Module receives policy data that has been created or modified since the last distribution. **Mandatory** **Accepted Value:** controlled-push
oracle.security.jps.runtime.pd.client.sm_name	**Description:** Defines the name of the Security Module. **Mandatory** **Accepted Value:** Name of the Security Module

Table C–3 (Cont.) Policy Distribution Client Configuration, JEE, Controlled Push Mode

Name	Information
oracle.security.jps.runtime.pd.client.localpolicy.work_folder	**Description:** Defines the name of any directory in which local cache files are stored. This directory must have read and write privileges. **Optional** **Accepted Value:** The name of any directory in which local cache files will be stored. This directory must have read and write privileges.
oracle.security.jps.runtime.pd.client.incrementalDistribution	**Description:** Defines whether the distribution is incremental or flush. *Incremental distribution* is when new and modified data is distributed. *Flush distribution* is when the Policy Distribution Component notifies the Security Module to cleanup locally stored policies in preparation for a complete re-distribution of all policy objects in the policy store. **Optional** **Accepted Values:** ■ false (policy distribution is flush for this Security Module) ■ true (default value; policy distribution is incremental for this Security Module if the required change logs are kept in the policy store)
oracle.security.jps.runtime.pd.client.registrationRetryInterval	**Description:** When a Security Module starts, it registers itself with the Policy Distribution Component to ensure the local policy cache is up to date. If registration fails, it will retry each time this interval of time passes until successful. **Optional** **Accepted Value:** time in seconds (default value is 5)
oracle.security.jps.runtime.pd.client.waitDistributionTime	**Description:** If this value is defined and not equal to zero, it specifies the amount of time that a Security Module will wait for initial policy distribution to happen. During this wait period, authorization requests are blocked until either the initial policy distribution completes or the configured period expires. **Optional** **Accepted Value:** time in seconds (default value is 60)
oracle.security.jps.runtime.pd.client.RegistrationServerURL	**Description:** Defines the URL of the Oracle Entitlements Server Administration Server. Used by the Security Module to register itself with Oracle Entitlements Server when it starts. **Mandatory** **Accepted Value:** URL
oracle.security.jps.runtime.pd.client.backupRegistrationServerURL	**Description:** Defines a backup URL for the Oracle Entitlements Server Administration Server. Used by the Security Module to register itself with Oracle Entitlements Server when it starts if the primary URL (parameter above) is unavailable. **Optional** (although if not configured Oracle Entitlements Server failover will not work) **Accepted Value:** URL

Table C–3 (Cont.) Policy Distribution Client Configuration, JEE, Controlled Push Mode

Name	Information
oracle.security.jps.runtime.pd.client.SMinstanceType	**Description:** Defines the type of Security Module to which the Policy Distribution Component client is connecting. **Mandatory** **Accepted Values:** ■ was ■ wls
oracle.security.jps.runtime.pd.client.DistributionServiceURL	**Description:** Defines the URL to which the remote Policy Distributor will push policy updates. **Mandatory** **Accepted Values:** URL

C.1.2.3 Policy Distribution Client Configuration (Controlled Pull Mode)

Table C–4 compiles the parameters for the Policy Distribution Component client configuration when the Oracle Entitlements Server is running in either a JEE or a JSE environment and is configured to distribute data in the controlled-pull mode.

Table C–4 Policy Distribution Client Configuration, Controlled Pull Mode

Name	Information
oracle.security.jps.runtime.pd.client.policyDistributionMode	Specifies the mode of policy distribution. *Controlled distribution* is initiated by the Policy Distribution Component, ensuring that the Security Module receives policy data that has been created or modified since the last distribution. **Mandatory** **Accepted Value:** controlled-pull
oracle.security.jps.runtime.pd.client.sm_name	**Description:** Defines the name of the Security Module. **Mandatory** **Accepted Value:** the name of the Security Module
oracle.security.jps.runtime.pd.client.localpolicy.work_folder	**Description:** Defines the name of any directory in which local cache files are stored. This directory must have read and write privileges. **Optional** **Accepted Value:** The name of any directory in which local cache files will be stored. This directory must have read and write privileges.
oracle.security.jps.runtime.pd.client.incrementalDistribution	**Description:** Defines whether the distribution is incremental or flush. *Incremental distribution* is when new and modified data is distributed. *Flush distribution* is when the Policy Distribution Component notifies the Security Module to cleanup locally stored policies in preparation for a complete re-distribution of all policy objects in the policy store. **Optional** **Accepted Values:** ■ false (policy distribution is flush for the Security Module) ■ true (default value; policy distribution is incremental for this Security Module if the required change logs are kept in the policy store)

Table C–4 (Cont.) Policy Distribution Client Configuration, Controlled Pull Mode

Name	Information
oracle.security.jps.runtime.pd.client.waitDistributionTime	**Description:** If this value is defined and not equal to zero, it specifies the amount of time that a Security Module will wait for initial policy distribution to happen. During this wait period, authorization requests are blocked until either the initial policy distribution completes or the configured period expires. **Optional** **Accepted Value:** time in seconds (default value is 60)
oracle.security.jps.runtime.pd.client.PollingTimerEnabled	**Description:** Enables a periodic check for policy updates in the Policy Store. Can be set to false to disable polling for environment when policies are not expected to be modified. **Optional** **Accepted Values:** - false - true (default value)
oracle.security.jps.runtime.pd.client.PollingTimerInterval	**Description:** Defines the interval of time in which the Policy Distribution Component will check for policy data changes. **Optional** **Accepted Value:** time in seconds (default value of 600)
oracle.security.jps.ldap.root.name	**Description:** Defines the top (root) entry of the LDAP policy store directory information tree (DIT). **Mandatory** **Accepted Value:** the top (root) entry of the LDAP policy store directory information tree (DIT)
oracle.security.jps.farm.name	**Description:** Defines the RDN format of the domain node in the LDAP policy store. **Mandatory** **Accepted Value:** name of the domain
jdbc.url	**Description:** Takes a URL that points to the database. **Mandatory** (if using Java Database Connectivity API to connect to policy store) **Accepted Value:** URL
jdbc.driver	**Description:** Location of the driver if using Java Database Connectivity API to connect to an Apache Derby database. **Mandatory** **Accepted Value:** driver
datasource.jndi.name	**Description:** The JNDI name of the JDBC data source instance. The instance may correspond to a single source or multi-source datasource. Valid in only JEE applications. Applies only to database stores. **Mandatory** **Accepted Value:** name of JNDI data source; for example, `jdbc/APMDBDS`.
bootstrap.security.principal.key	**Description:** The key for the password credentials to access the policy store. Credentials are stored in the Credential Store Framework (CSF) store. **Mandatory** **Accepted Value:** CSF credential key

Table C–4 (Cont.) Policy Distribution Client Configuration, Controlled Pull Mode

Name	Information
bootstrap.security.principal.map	**Description:** The map for the password credentials to access the policy store. Credentials are stored in the CSF store. **Mandatory** **Accepted Value:** name of the CSF credential map

C.1.2.4 Policy Distribution Client Configuration (Non-controlled Mode)

Table C–5 compiles the parameters for Policy Distribution Component client configuration when the Oracle Entitlements Server is running in either a JEE or a JSE environment and is configured to distribute data in the non-controlled mode.

Table C–5 Policy Distribution Client COnfiguration, Non-controlled Mode

Name	Information
oracle.security.jps.runtime.pd.client.policyDistributionMode	**Description:** Specifies the mode of policy distribution. *Non-controlled distribution* is when the Security Module initiates the periodic retrieval of policy data from a policy store (or from a component that serves as an intermediary between the two). **Optional** **Accepted Value:** non-controlled (default value)

C.2 Security Module Configuration

This section covers the configurations for the various types of Security Modules and their proxy clients.

- Section C.2.1, "Java Security Module"
- Section C.2.2, "Web Services Security Module"
- Section C.2.3, "RMI Security Module"
- Section C.2.4, "WebLogic Server Security Module"

C.2.1 Java Security Module

Table C–6 compiles the parameters to configure the Java Security Module embedded in either a JSE or a JEE container.

Table C–6 Java Security Module Configuration Parameters

Name	Information
oracle.security.jps.policystore.rolemember.cache.type	**Description:** Defines the role member cache type. Valid in J2EE and J2SE applications. Applies to LDAP and database stores. **Optional** **Accepted Values** ■ SOFTHASH (cleaning of a cache of this type relies on the garbage collector when there is a memory crunch) ■ WEAK (behavior of a cache of this type is similar to a cache of type SOFT but the garbage collector cleans it more frequently) ■ STATIC (default value; cache objects are statically cached and can be cleaned explicitly only according to the applied cache strategy, such as FIFO; the garbage collector does not clean a cache of this type)
oracle.security.jps.policystore.rolemember.cache.strategy	**Description:** Defines the type of strategy used in the role member cache. Valid in J2EE and J2SE applications. Applies to LDAP and database stores. **Optional** **Accepted Values** ■ NONE (all entries in the cache grow until a refresh or reboot occurs; there is no control over the size of the cache; not recommended but typically efficient when the policy footprint is very small) ■ FIFO (default value; the cache implements the first-in-first-out strategy)
oracle.security.jps.policystore.rolemember.cache.size	**Description:** Defines the number of roles kept in the role member cache. Valid in J2EE and J2SE application. Applies to LDAP and database stores. **Optional** **Accepted Value:** number (default value is 1000)
oracle.security.jps.policystore.rolemember.cache.warmup.enable	**Description:** Controls the way the Application Role membership cache is created. Valid in J2EE and J2SE applications. Applies to LDAP and database stores. **Optional** **Accepted Values** ■ true (the cache is created at server startup; use when the number of users and groups is significantly higher than the number of Application Roles) ■ false (default value; the cache is created on demand - lazy loading; use when the number of Application Roles is very high)
oracle.security.jps.policystore.policy.lazy.load.enable	**Description:** Enables or disables the policy lazy load. Valid in J2EE and J2SE applications. Applies to LDAP and database stores. **Optional** **Accepted Values** ■ false ■ true (default value)

Table C–6 (Cont.) Java Security Module Configuration Parameters

Name	Information
oracle.security.jps.policystore.policy.cache.strategy	**Description:** Defines the type of strategy used in the permission cache. Valid in J2EE and J2SE applications. Applies to LDAP and database stores. **Optional** **Accepted Values** ■ NONE (all entries in the cache grow until a refresh or reboot occurs; there is no control over the size of the cache; not recommended but typically efficient when the policy footprint is very small.) ■ PERMISSION_FIFO (default value; the cache implements the first-in-first-out strategy)
oracle.security.jps.policystore.policy.cache.size	**Description:** Defines the number of permissions kept in the permission cache. Valid in J2EE and J2SE applications. Applies to LDAP and database stores. **Optional** **Accepted Value:** number (default value is 1000)
oracle.security.jps.policystore.policy.cache.updateable	**Description:** Defines whether the policy cache is incrementally updated for management operations on policy data. **Optional** **Accepted Values** ■ false ■ true (default value)
oracle.security.jps.policystore.refresh.enable	**Description:** Enables or disables the policy store refresh. If this property is set, `oracle.security.jps.ldap.cache.enable` cannot be set. Valid in J2EE and J2SE applications. Applies to LDAP and database stores. **Optional** **Accepted Values:** ■ false ■ true (default value)
oracle.security.jps.policystore.refresh.purge.timeout	**Description:** Defines the time in milliseconds after which the policy store cache is purged. Valid in J2EE and J2SE applications. Applies to LDAP and database stores. **Optional** **Accepted Value:** time in milliseconds; default value is 43200000 which equals 12 hours
oracle.security.jps.ldap.policystore.refresh.purge.interval	**Description:** Defines the interval of time in which the policy store is polled for changes. Valid in J2EE and J2SE applications. Applies to LDAP and database stores. **Optional** **Accepted Value:** time in milliseconds; default value is 600000 which equals 10 minutes
oracle.security.jps.pdp.missingAppPolicyQueryTTL	**Description:** Defines the interval of time to avoid frequently querying a non-exist Application (`ApplicationPolicy`) object. **Optional** **Accepted Value:** time to live in milliseconds (default value is 60000)

Table C–6 (Cont.) Java Security Module Configuration Parameters

Name	Information
oracle.security.jps.pdp.AuthorizationDecisionCacheEnabled	**Description:** Specifies whether the authorization cache should be enabled. Valid in J2EE and J2SE applications. Applies to XML, LDAP, and database stores. **Optional** **Accepted Values** ■ false ■ true (default value)
oracle.security.jps.pdp.AuthorizationDecisionCacheEvictionCapacity	**Description:** Defines the maximum number of authorization and role mapping sessions to maintain. When the maximum is reached, old sessions are dropped and reestablished when needed. Valid in J2EE and J2SE applications. Applies to XML, LDAP, and database stores. **Optional** **Accepted Value:** number (default value is 500)
oracle.security.jps.pdp.AuthorizationDecisionCacheEvictionPercentage	**Description:** Defines the percentage of sessions to drop when the eviction capacity is reached. Valid in J2EE and J2SE applications. Applies to XML, LDAP, and database stores. **Optional** **Accepted Value:** number (default value is 10)
oracle.security.jps.pdp.AuthorizationDecisionCacheTTL	**Description:** Defines the number of seconds during which session data is cached. Valid in J2EE and J2SE applications. Applies to XML, LDAP, and database stores. **Optional** **Accepted Value:** time in seconds (default value is 60)
oracle.security.jps.pdp.anonymousrole.enable	**Description:** Specifies whether anonymous role has to be added to anonymous subject for policy matching. **Optional** **Accepted Values** ■ false ■ true (default value)
oracle.security.jps.pdp.authenticatedrole.enable	**Description:** Specifies whether authenticated role has to be added to authenticated subject for policy matching. **Optional** **Accepted Values** ■ false ■ true (default value)

C.2.2 Web Services Security Module

Table C–7 compiles the parameters to configure the Web Services Security Module embedded in either a JSE or a JEE container.

Table C–7 Web Services Security Module Configuration Parameters

Name	Information
oracle.security.jps.pdp.wssm.WSServiceRegistryPortNumber	**Description:** Defines the port on which the Web Services Security Module listens. **Mandatory** **Accepted Value:** port number
oracle.security.jps.pdp.wssm.WSServiceRegistryHost	**Description:** Defines the name of the server on which the Web Services Security Module is running. **Optional** **Accepted Value:** server name (default value is localhost)
oracle.security.jps.pdp.wssm.Protocol	**Description:** Defines the transport protocol used between the Policy Distribution Component client and server. **Optional** **Accepted Values** ■ https ■ http (default value)
oracle.security.jps.pdp.sm.IdentityMaxCacheSize	**Description:** Specifies the maximum number of users for which information is cached. When the maximum is reached, old records are dropped and reestablished when needed. **Optional** **Accepted Value:** number
oracle.security.jps.pdp.sm.IdentityCacheEvictionPercentage	**Description:** Specifies percentage of identities that must be evicted when cache has reached the maximum size. **Optional** **Accepted Value:** number indicating percentage
oracle.security.jps.pdp.sm.IdentityCachedEntryTTL	**Description:** Specifies time-to-live of an identity cache record. **Optional** **Accepted Value:** time in seconds
oracle.security.jps.pdp.wssm.responseContext	**Description:** Specifies whether to merge data from many AppContext responses into a single AppContext response. **Optional** **Accepted Values** ■ Merged ■ Unmerged (default value)
oracle.security.jps.pdp.wssm.ssl.identityKeyStoreFileName	**Description:** Defines the name of the Identity Key Store file where client certificates are stored for the Web Services Security Module. Used for SSL communications between the remote client and the Web Services Security Module. **Optional** **Accepted Value:** name of the Identity Key Store file
oracle.security.jps.pdp.wssm.ssl.trustKeyStoreFileName	**Description:** Defines the name of the Trust Key Store file in which CA certificates are stored. Used for SSL communications between the remote client and the Web Services Security Module. **Optional** **Accepted Value:** name of the Trust Key Store file

Table C–7 (Cont.) Web Services Security Module Configuration Parameters

Name	Information
oracle.security.jps.pdp.wssm.ssl.identityKeyStoreKeyAlias	**Description:** Specifies the Identity Key alias used to identify the Web Services Security Module client certificate used for SSL communication between the Web Services Security Module and the remote client. Accepted value: Identity key alias **Optional** **Accepted Value:** Identity Key alias

C.2.3 RMI Security Module

Table C–8 compiles the parameters to configure the RMI Security Module embedded in either a JSE or a JEE container.

> **Note:** Currently this configuration is for a standalone deployment. We need to add the Container based configuration later.

Table C–8 RMI Security Module Configuration Parameters

Name	Information
oracle.security.jps.pdp.rmism.RMIRegistryPortNumber	**Description:** Defines the port on which the RMI Security Module listens to the RMI server. **Mandatory** **Accepted Value:** port number.
oracle.security.jps.pdp.rmism.UseSSL	**Description:** Defines whether the SSL protocol is used for secure communication between the RMI Security Module and RMI server. **Optional** **Accepted Values** ▪ true ▪ false (default)
oracle.security.jps.pdp.sm.IdentityMaxCacheSize	**Description:** Specifies the maximum number of users for which information is cached. When the maximum is reached, old records are dropped and reestablished when needed. **Optional** **Accepted Value:** number
oracle.security.jps.pdp.sm.IdentityCacheEvictionPercentage	**Description:** Specifies percentage of identities that must be evicted when cache has reached the maximum size. **Optional** **Accepted Value:** number representing percentage
oracle.security.jps.pdp.sm.IdentityCachedEntryTTL	**Description:** Specifies the time-to-live of an identity cache record. **Optional** **Accepted Value:** time in seconds

C.2.4 WebLogic Server Security Module

Table C–9 compiles the parameters to configure the WebLogic Server (WLS) Security Module embedded in a JEE container. These parameters are used only when the WLS Security Module is configured to be used as a PEP.

- See *Oracle Fusion Middleware Administrator's Guide for Oracle Entitlements Server* for architectural details regarding the Security Module.
- See Section 13.1, "Integrating with WebLogic Server" to enable the WebLogic Server Security Module.

Table C–9 WebLogic Server Security Module Configuration Parameters

Name	Information
oracle.security.jps.pdp.wlssm.UndefinedApplicationEffect	**Description:** Specifies the effect that the provider has to return if an application is not defined in the policy store. **Optional** **Accepted Values** ▪ permit ▪ abstain ▪ deny
oracle.security.jps.pdp.wlssm.NoApplicablePolicyEffect	**Description:** Specifies the effect that the provider has to return if no applicable policies have been found. **Optional** **Accepted Values** ▪ permit (represents an open system) ▪ abstain ▪ deny (represents a closed system)

C.3 PDP Proxy Configuration

This section contains information regarding configuration for the Security Module proxies.

- Section C.3.1, "Web Services Security Module Proxy Client"
- Section C.3.2, "RMI Security Module Proxy Client"

C.3.1 Web Services Security Module Proxy Client

Table C–10 compiles the parameters to configure the Web Services Security Module proxy client.

Table C–10 Web Services Proxy Client Configuration Parameters

Name	Information
oracle.security.jps.pdp.PDPTransport	**Description:** Specifies the underlying protocol to be used by Multi-protocol Security Module to communicate with Oracle Entitlements Server. **Mandatory** **Accepted Values**: no default value; XACML is always available in the Web Services Security Module. ▪ WS ▪ RMI
oracle.security.jps.pdp.proxy.PDPAddress	**Description:** Specifies the host and port number of either the Web Services Security Module. For example, `http://dadvml0134:9015` **Optional** **Accepted Value:** a comma separated list of URIs (if more then one address is specified the first is considered the primary, and the rest as backups)
oracle.security.jps.pdp.proxy.RequestTimeoutMilliSecs	**Description:** Defines the interval of time in which an authorization request times out when the remote PDP (RMI or Web Services Security Module) is not responding. **Optional** **Accepted Value:** time in milliseconds (default value is 10000)
oracle.security.jps.pdp.proxyFailureRetryCount	**Description:** Specifies the number of attempts to make before attempting the alternate failover server. **Optional** **Accepted Value:** number (default value is 3)
oracle.security.jps.pdp.proxy.FailbackTimeoutMilliSecs	**Description:** Specifies the interval of time after which a failed primary server is tried again for failover. **Optional** **Accepted Value:** time in milliseconds (default value is 180000)
oracle.security.jps.pdp.proxy.SynchronizationIntervalMilliSecs	**Description:** Defines how often the PDP Proxy polls the PDP server in order to synchronize its state. For example, the interval is used to periodically check whether the authorization cache has to be flushed. **Optional** **Accepted Value:** time in seconds (default value is 60)
oracle.security.jps.pdp.proxy.wssm.ssl.identityKeyStoreFileName	**Description:** Defines the name of the Identity Key Store file where client certificates for the Web Services Security Module are stored. Used for SSL communication between a client and the Web Services Security Module. **Optional** **Accepted Value:** name of the Identity Key Store file
oracle.security.jps.pdp.proxy.wssm.ssl.trustKeyStoreFileName	**Description:** Defines the name of the Trust Key Store file where CA certificates for Web Services Security Module are stored. Used for SSL communication between a client and the Web Services Security Module. **Optional** **Accepted Value:** the name of the Trust Key Store file.

Table C–10 (Cont.) Web Services Proxy Client Configuration Parameters

Name	Information
oracle.security.jps.pdp.proxy.wssm.ssl.identityKeyStoreKeyAlias	**Description:** Specifies the alias name of the Web Services client certificate. Used for SSL communication between a client and the Web Services Security Module. **Optional** **Accepted Value:** alias of the identity key store (if only one alias exists in the identity key store, no need to specify this value)
oracle.security.jps.pdp.proxy.wssm.protocol	**Description:** Defines the transport protocol used between the Policy Distribution Component client and server. **Optional** **Accepted Values** - https - http (default value)

C.3.2 RMI Security Module Proxy Client

Table C–11 compiles the parameters to configure the RMI Security Module Proxy Client.

Table C–11 PDP RMI Proxy Client Configuration Parameters

Name	Information
oracle.security.jps.pdp.PDPTransport	**Description:** Specifies the underlying protocol to be used by Multi-protocol Security Module to communicate with Oracle Entitlements Server. **Mandatory** **Accepted Values:** no default value; XACML is always available in the RMI Security Module. - WS - RMI
oracle.security.jps.pdp.proxy.PDPAddress	**Description:** Specifies the host and port number of the RMI Security Module. For example, `rmi://localhost:9400` **Mandatory** **Accepted Value:** a comma separated list of URIs (if more then one address is specified the first is considered the primary, and the rest as backups)
oracle.security.jps.pdp.proxy.RequestTimeoutMilliSecs	**Description:** Defines the interval of time in which an authorization request times out when the remote PDP (RMI or Web Services Security Module) is not responding. **Optional** **Accepted Value:** time in milliseconds (default value is 10000)
oracle.security.jps.pdp.proxyFailureRetryCount	**Description:** Specifies the number of attempts to make before attempting the alternate failover server. **Optional** **Accepted Value:** number (default value is 3)
oracle.security.jps.pdp.proxy.FailbackTimeoutMilliSecs	**Description:** Specifies the interval of time after which a failed primary server is tried again for failover. **Optional** **Accepted Value:** time in milliseconds (default value is 180000)

Table C–11 (Cont.) PDP RMI Proxy Client Configuration Parameters

Name	Information
oracle.security.jps.pdp.proxy.SynchronizationIntervalMilliSecs	**Description:** Defines how often the PDP Proxy polls the PDP server in order to synchronize its state. For example, the interval is used to periodically check whether the authorization cache has to be flushed. **Optional** **Accepted Value:** time in seconds (default value is 60)

C.4 Policy Store Service Configuration

Table C–12 compiles the configuration parameters for the Policy Store Service.

Table C–12 Policy Store Service Configuration Parameters

Name	Information
ldap.url	**Description:** Defines the URL of the LDAP policy store. Valid in JEE and JSE applications and only applies to LDAP stores. **Mandatory** **Accepted Value:** URI of the LDAP policy store in the format `ldap://host:port`.
max.search.filter.length	**Description:** Defines the maximum length of a search filter. **Mandatory** **Accepted Value:** integer defining the maximum length of a search filter; for example, 1024
oracle.security.jps.ldap.root.name	**Description:** Defines the RDN format of the root node in the LDAP policy store. Valid in JEE and JSE applications. Applies to LDAP and database stores. **Mandatory** **Accepted Value:** root name of jps context; for example, `cn=jpsroot`.
oracle.security.jps.farm.name	**Description:** Defines the RDN format of the root node in the LDAP policy store. Valid in JEE and JSE applications. Applies to LDAP and database stores. **Mandatory** **Accepted Value:** farm name of the domain; for example, `cn=base_domain`.

Table C–12 (Cont.) Policy Store Service Configuration Parameters

Name	Information
oracle.security.jps.policystore.resourcetypeenforcementmode	**Description:** Controls the throwing of exceptions if any of the following checks fail: ■ Verify that if two resource types share the same permission class, that permission must be either ResourcePermission or extend AbstractTypedPermission, and this last resource type cannot be created. ■ Verify that all permissions have resource types defined, and that the resource matcher permission class and the permission being granted match. Valid in JEE and JSE applications. Applies to LDAP and database stores. **Optional** **Accepted Values** ■ strict (when any of the above checks fail, the system throws an exception and the operation is aborted) ■ lenient (default value; when any of the above checks fail, the system does not throw any exceptions, the operation continues without disruption, and any discrepancies encountered are logged)
bootstrap.security.principal.key	**Description:** Defines the key for the password credentials to access the LDAP policy store, stored in the CSF store. Valid in JEE and JSE applications. Applies to LDAP and database stores. **Mandatory** **Accepted Value:** the key name of the credential; for example, `oes_sm_key`. The out-of-the-box value is `bootstrap`.
bootstrap.security.principal.map	**Description:** Defines the map for the password credentials to access the LDAP policy store, stored in the CSF store. Valid in JEE and JSE applications. Applies to LDAP and database stores. **Mandatory** **Accepted Value:** map name of the credential; for example, `oes_sm_map`. The default value is `BOOTSTRAP_JPS`.
jdbc.driver	**Description:** Defines the name of the JDBC driver. **Mandatory** **Accepted Value:** name of the JDBC driver.
datasource.jndi.name	**Description:** The JNDI name of the JDBC data source instance. The instance may correspond to a single source or multi-source datasource. Valid in only JEE applications. Applies only to database stores. **Mandatory** **Accepted Value:** name of JNDI data source; for example, `jdbc/APMDBDS`.
jdbc.url	**Description:** Defines the JDBC driver connection URL. **Mandatory** **Accepted Value:** the JDBC driver connection URL.

Table C–12 (Cont.) Policy Store Service Configuration Parameters

Name	Information
oracle.security.jps.pd.localMode	**Description:** Defines whether the policy store is running in local mode. **Mandatory** **Accepted Values** ■ true ■ false

Index