BEA AquaLogic Enterprise Security™®

Integrating ALES with Application Environments

Version 2.1
Document Revised: December 19, 2005

Copyright

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Third-Party Software License Agreement

Sun Microsystems, Inc.’s XACML implementation v2.0
Copyright © 2003-2004 Sun Microsystems, Inc. All Rights Reserved.

This product includes Sun Microsystems, Inc.’s XACML implementation v2.0, which is governed by the following terms:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors maybe used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL,

For all third-party software license agreements, see the 3rd_party_licenses.txt file, which is placed in the \ales21-admin directory when you install the AquaLogic Enterprise Security Administration Server.

Trademarks or Service Marks

About This Document

1. Using the Administration Console

2. Securing ALES Components

3. Setting Up Application Security Administrators

4. Integrating ALES with Applications

5. Enabling SAML-based Single Sign-On

6. Enabling SPNEGO-based Single Sign-on

7. Configuring Metadirectories

8. Authorization Caching

About This Document

This document describes important tasks associated with integrating ALES into application environments. It is organized as follows:

- Chapter 1, "Using the Administration Console," provides a short introduction to the ALES Administration Console.
- Chapter 2, "Securing ALES Components," describes how to control access to ALES using the Administration Console.
- Chapter 3, "Setting Up Application Security Administrators," describes how to establish application-security administrators using the Administration Console.
- Chapter 4, "Integrating ALES with Applications," describes how to configure SSMs and bind them to application servers.
- Chapter 5, "Enabling SAML-based Single Sign-On," describes how to configure SSMs to provide SSO using the SAML 1.1 Browser POST Profile.
- Chapter 6, "Enabling SPNEGO-based Single Sign-on," describes the setup steps necessary to achieve Single Sign-On (SSO) integration with .NET based web services clients, as well as Internet Explorer browser clients.
- Chapter 8, "Authorization Caching," describes how to configure and manage authorization caching.
- Chapter 8, "Authorization Caching," discusses issues of performance when implementing authorization policies and how to configure your providers for the improved performance.

Audience

This document distinguishes between two levels of security administrators.

- Application Security Administrators — these administrators are responsible for integrating ALES into application environments, managing interaction between an applications and ALES, and setting up application-level security administrators.

  Typical tasks include modifying deployment descriptors, managing security providers and other security configurations, managing single sign-on scripts, setting up application-level security administrators.

- Application-level Security Administrators — these administrators are responsible for securing applications using ALES policies.

  The primary task is to create and deploy the policies securing application resources.

This document is intended for Application Security Administrators.

Product Documentation

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

`http://dev2dev.bea.com`

To view the documentation for a particular product, select that product from the Product Centers menu on the left side of the screen on the dev2dev page. Select More Product Centers. From the BEA Products list, choose AquaLogic Enterprise Security 2.1. The home page for this product is displayed. From the Resources menu, choose Documentation 2.1. The home page for the complete documentation set for the product and release you have selected is displayed.

Related Information

The BEA corporate web site provides all documentation for BEA AquaLogic Enterprise Security. Other BEA AquaLogic Enterprise Security documents that may be of interest to the reader include:

- *Introduction to AquaLogic Enterprise Security*—This document summarizes the features of the BEA AquaLogic Enterprise Security products and presents an overview of the architecture and capabilities of the security services. It provides a starting point for understanding the family of BEA AquaLogic Enterprise Security products.

- *BEA AquaLogic Enterprise Administration and Deployment Guide* —This document describes tasks associated with deploying and managing ALES
- *Programming Security for Java Applications*—This document describes how to implement security in Java applications. It includes descriptions of the Security Service Application Programming Interfaces and programming instructions.
- *Developing Security Providers for BEA AquaLogic Enterprise Security* —This document provides security vendors and security and application developers with the information needed to develop custom security providers.
- *BEA AquaLogic Enterprise Security Policy Managers Guide*—This document defines the policy model used by BEA AquaLogic Enterprise Security, and describes how to import and export policy data.
- *Javadocs for Java API—*This document provides reference documentation for the Java Application Programming Interfaces that are provided with and supported by this release of BEA AquaLogic Enterprise Security.
- *Javadocs for Security Service Provider Interfaces—*This document provides reference documentation for the Security Service Provider Interfaces that are provided with and supported by this release of BEA AquaLogic Enterprise Security.

Using the Administration Console

Many of the tasks described in the document are performed using the Administration Console. This chapter describes how to access the console and provides a brief introduction to the console interface.

For more detailed information about using the Administration Console, consult its help system.

- "Accessing the Administration Console" on page 1-1
- "Setting Administration Console Preferences" on page 1-2

Accessing the Administration Console

To access the Administration Console, make sure the Administration Server is running. Then enter: `https://<hostname>:<port>/asi`

`where`:

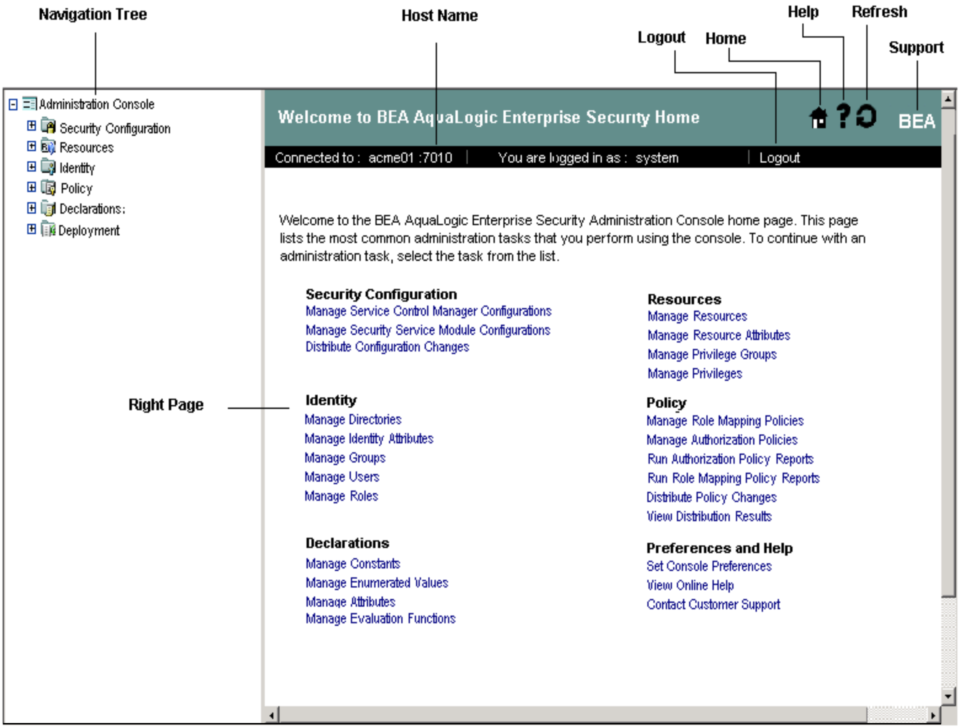
`hostname` –the Domain Name Server (DNS) name or IP address of the machine where the Administration Server running.

`port` –the SSL port number through which the administration server connects (default is 7010).

When the login page appears, enter the username and password (`weblogic`) and click Sign In.

Figure 1-1 shows the Administration Console home page.

Figure 1-1 Administration Console Interface



Welcome to BEA AquaLogic Enterprise Security Home

Connected to : acme01 :7010 | You are logged in as : system | Logout

Welcome to the BEA AquaLogic Enterprise Security Administration Console home page. This page lists the most common administration tasks that you perform using the console. To continue with an administration task, select the task from the list.

Security Configuration
- Manage Service Control Manager Configurations
- Manage Security Service Module Configurations
- Distribute Configuration Changes

Resources
- Manage Resources
- Manage Resource Attributes
- Manage Privilege Groups
- Manage Privileges

Identity
- Manage Directories
- Manage Identity Attributes
- Manage Groups
- Manage Users
- Manage Roles

Policy
- Manage Role Mapping Policies
- Manage Authorization Policies
- Run Authorization Policy Reports
- Run Role Mapping Policy Reports
- Distribute Policy Changes
- View Distribution Results

Declarations
- Manage Constants
- Manage Enumerated Values
- Manage Attributes
- Manage Evaluation Functions

Preferences and Help
- Set Console Preferences
- View Online Help
- Contact Customer Support

Setting Administration Console Preferences

To customize display, access the Preferences tab by clicking on the top node in the navigation tree named `Administration Console`. The page allows you to specify a filter string and page size for each node in the navigation tree.

Table 1-1 Console Display Options

Option	Description
Filter String	The default pattern to search for when retrieving objects in the node. The default setting is an asterisk (*) which displays all objects. Use wildcards as follows: * finds represents zero or more characters. ? retrieves any one character, e.g., J*oe retrieves JDoe, JPoe, JWoe, etc. [ ] represents one character. For example, a search for J?oe will find both JDoe and Jpoe but not JFloe. Brackets [ ] mean match any character inside the brackets. For example, [Aac]* matches Apple, apple, and cat, but does not match Cat.
Page Size	Number of items displayed per page.

Securing ALES Components

ALES is itself secured using the same policy model used to secure any other application. This chapter explains the default policies controlling administrative access to ALES and how to use the Administration Console to customize these defaults to local needs.

Information is provided in the following sections.

Overview

Installing ALES provides a number of database objects that collectively define access to ALES components. This provides rudimentary security at startup and you may use the Administration Console to more completely define administrative access.

The default database objects are listed below and are more fully described in sections that follow.

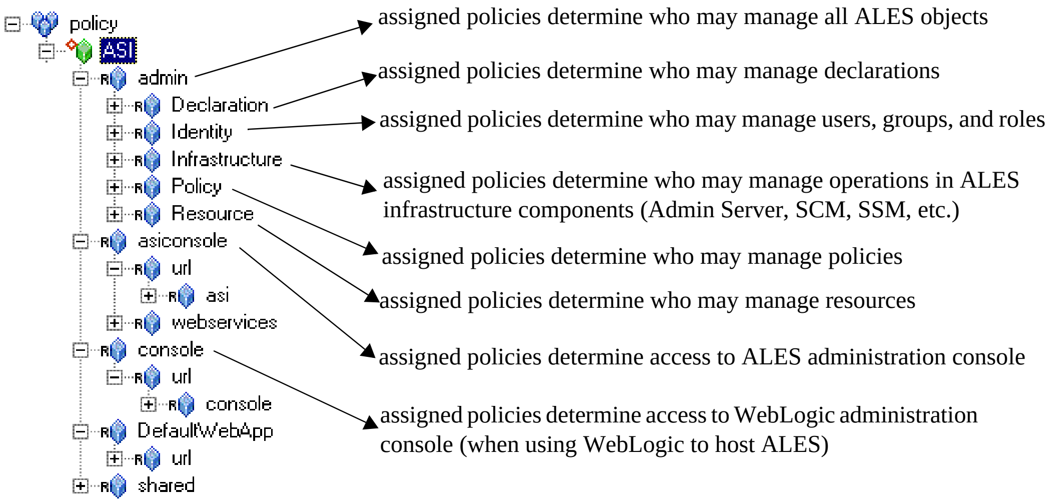
Table 2-1 Default Database Objects Defining Access to ALES

Object Type	Description
Resource	A representation of ALES components is defined in a separate tree under a root resource named ASI. Policies can be assigned to a resource representing an ALES component and thereby define access to that component.
Identity	A number of users, groups, and roles that reflect usage of ALES are provided. In particular, a user named `system` is set up as having complete administrative rights to the database.
Role Mapping Policies	A number of role mapping policies are provided that assign some of the default roles to users/groups.
Authorization Policies	A number of authorization policies are provided that assign privileges to roles/groups/users on specific resources in the ASI resource tree.

ALES Resources

ALES components are represented under the ASI resource tree, as shown in the figure below.

Figure 2-1 Representation of ALES Components



assigned policies determine who may manage all ALES objects

assigned policies determine who may manage declarations

assigned policies determine who may manage users, groups, and roles

assigned policies determine who may manage operations in ALES infrastructure components (Admin Server, SCM, SSM, etc.)

assigned policies determine who may manage policies

assigned policies determine who may manage resources

assigned policies determine access to ALES administration console

assigned policies determine access to WebLogic administration console (when using WebLogic to host ALES)

Administrative Operations

Table 2-2 describes resource objects that define the administrative operations that are performed using the Administration Console. By default, these resources are contained within `//app/policy/ASI/admin`.

Table 2-2 Resources Defining Administrative Operations

Resource Name	Description
admin/Declaration/Attribute	Used to protect operations on attribute declarations.
admin/Declaration/Constant	Used to protect operations on constant declarations.
admin/Declaration/Enumeration	Used to protect operations on enumeration declarations.
admin/Declaration/EvaluationFunction	Used to protect operations on evaluation function declarations.
admin/Identity/Directory/Instance	Used to protect operations on identity directory instances.
admin/Identity/Directory/AttributeMapping/Single	Used to protect operations on what scalar attributes may be assigned to users within a directory.
admin/Identity/Directory/AttributeMapping/List	Used to protect operations on what vector attributes may be assigned to users within a directory.
admin/Identity/Subject/User	Used to protect operations on users.
admin/Identity/Subject/Group	Used to protect operations on groups.
admin/Identity/Subject/Password	Used to protect operations on user passwords.
admin/Identity/Subject/AttributeAssignment/Single	Used to protect operations on scalar subject attribute values.
admin/Identity/Subject/AttributeAssignment/List	Used to protect operations on vector subject attribute values.
admin/Resource/Instance	Used to protect operations on resources.
admin/Resource/AttributeAssignment/Single	Used to protect operations on scalar resource attribute values.
admin/Resource/AttributeAssignment/List	Used to protect operations on vector resource attribute values.
admin/Resource/MetaData/LogicalName	Used to protect operations on setting the "logical name" resource metadata.
admin/Resource/MetaData/IsApplication	Used to protect operations on setting the "is application" resource metadata.
admin/Resource/MetaData/IsDistributionPoint	Used to protect operations on setting the "is distribution point" metadata.
admin/Policy/Grant	Used to protect operations on grant policies.
admin/Policy/Deny	Used to protect operations on deny policies.
admin/Policy/Delegate	Used to protect operations on delegate policies.
admin/Policy/Action/Role/Instance	Used to protect operations on roles (when used as actions).

Table 2-2 Resources Defining Administrative Operations

Resource Name	Description
admin/Policy/Action/Privilege/Instance	Used to protect operations on privileges.
admin/Policy/Action/Privilege/Group	Used to protect operations on privilege groups.
admin/Policy/Analysis/InquiryQuery	Used to protect operations on policy inquiries.
admin/Policy/Analysis/VerificationQuery	Used to protect operations on policy verification.
admin/Infrastructure/Engines/ARME	Used to protect operations on definitions of the Authorization and Role Mapping Engine (ARME).
admin/Infrastructure/Engines/SCM	Used to protect operations on definitions of the Service Control Manager (SCM).
admin/Infrastructure/Management/BulkManager	Used to protect operations on the policy loader.
admin/Policy/Repository	Used to protect operations on the policy repository.

Privileges

Table 2-3 lists and describes the default privileges that may be assigned.

Table 2-3 Privileges

Privilege	Explanation
`create`	Create a policy element, including identities, directories, users, groups, attributes), resources and their attributes, configuration data and their bindings, and privileges and privilege groups.
`view`	View the contents of a policy element, including identities (identity directories, users, groups, identity attributes), resources and their attributes, configuration data and their bindings, privileges and privilege groups.
`delete`	Delete a policy element, including identities (identity directories, users, groups, identity attributes), resources and their attributes, configuration data and their bindings, and privileges and privilege groups.
`cascadeDelete`	Delete an element and its sub-elements (no permission check is made on sub-elements), including identities (identity directories, users, groups, identity attributes), resources and their attributes, configuration data and their bindings, and privileges and privilege groups.
`rename`	Rename a policy element, including identities (identity directories, users, groups, identity attributes), resources and their attributes, configuration data and their bindings, and privileges and privilege groups.

Table 2-3 Privileges (Continued)

Privilege	Explanation
`modify`	Modify the contents of a policy element, including identities (identity directories, users, groups, identity attributes), resources and their attributes, configuration data and their bindings, and privileges and privilege groups.
`listAll`	Filter lists of instances based on a pattern specification.
`addMember`	Add a member to a group.
`removeMember`	Remove a member from a group.
`execute`	Execute a policy analysis query.
`deployUpdate`	Deploy a policy update.
`deployStructuralChange`	Deploy a structural change.
`bind`	Bind a resource to an ASI Authorization and ASI Role Mapping provider.
`unbind`	Unbind a resource from an ASI Authorization and ASI Role Mapping provider.
`login`	Log on to the Administration Application, including the Administration Console, and the Policy Import and Export tools.
`copy`	Copy a policy element, including identities (identity directories, users, groups, identity attributes), resources and their attributes, configuration data and their bindings, and privileges and privilege groups.

Context Attributes

Context attributes can be used to provide fine-grained protection of policy operations. For example, when creating a privilege, the name of the privilege can be supplied as an attribute and used to control access to a single unique privilege.

Table 2-4 describes the default context attributes.

Table 2-4 Context Attributes

Attribute Name	Data Type	Description
declaration	string	Name of a declaration.
data_type	string	The name of a data type, for example, a string, integer, date.
attribute_usage_type	Enumeration (resource_attribute, subject_attribute, dynamic_attribute)	Specifies the type of policy element with which an attribute declaration is associated.

Table 2-4 Context Attributes (Continued)

Attribute Name	Data Type	Description
new_name	string	Generic attribute used when renaming elements.
new_attribute_usage_type	Enumeration (resource_attribute, subject_attribute, dynamic_attribute)	The new value for this item used to modify operations.
value	string	Generic attribute used to represent the value of an element.
values	list of strings	Generic attribute used to represent the value of an element as a list.
directory	string	The name of a directory.
attribute	string	The name of an attribute.
default_value	string	The default value of an attribute.
default_values	list of strings	The default value of a list attribute.
new_default_value	string	Used in modification operations to represent the new default value of an attribute value.
new_default_values	list of strings	Used in modification operations to represent the new default value of a list attribute.
subject_name	string	The name of a subject.
subjects	list of strings	A list of subjects.
groups	list of strings	The group membership of the subject.
subject_type	Enumeration (user_subject, group_subject, role_subject)	The type of subject.
member_subject_type	Enumeration (user_subject, group_subject, role_subject)	The type of the subject group member.
member_subject	string	Name of subject group member.
action	string	Name of the action.
action_type	Enumeration (privilege_action, role_action)	Type of the action.
resource	string	The name of the resource.
resources	list of strings	A list of resources.
constraint	string	The constraint of a policy; this is the portion between the ‘if’ and ‘;’ exclusive.
new_action	string	Name of new action in a modified policy.
new_action_type	string	New action type in a modified policy.
new_resource	string	New resource in a modified policy.

Table 2-4 Context Attributes (Continued)

Attribute Name	Data Type	Description
new_subject_name	string	New subject name.
new_constraint	string	New constraint in a modified policy.
delegator	string	The name of the delegator in a policy.
new_delegator	string	New delegator in a modified policy.
actions	list of strings	A set of actions.
action_groups	list of strings	A list of privilege group names.
action_group	string	The name of a privilege group.
parent_resource	string	The parent of the resource.
meta_data	string	The name of the metadata item.
logical_name	string	The logical name of a resource.
deleted_directories	list of strings	A list of deleted directories.
deleted_engines	list of strings	A list of deleted engines.[1]
deployed_engines	list of strings	A list of deployed engines.
deleted_bindings	list of strings	A list of deleted engine binding node pairs.
deleted_applications	list of strings	A list of deleted applications.
engine	string	The name of an ARME or SCM cluster.
engine_bindings	list of strings	A list of bindable resources bound to the ARME or SCM.
owner	string	The owner of analysis query.
effect_type	Enumeration (grant_effect, deny_effect, delegate_effect)	The type of role mapping and authorization policy effect.
title	string	The title of a analysis query.

1. The term engine refers to an ASI Authorization provider and ASI Role Mapper provider that are configured to operate in conjunction with one another, also referred to as the ARME. This combination of providers are configured to manage your authorization and role mapping policies.

Evaluation Functions

The evaluation functions listed in Table 2-5 are provided for writing custom administration policies. They may be used in the constraint portion of policies to limit the applicability of the policy based on contextual information.

Table 2-5 Evaluation Functions

Function Name	Description
resource_is_child(c,p,[d])	Check if c a child of p. d is a Boolean standing for direct. By default, d is true, meaning check if c is directly a child of p. If false, then c may be a descendant of p at any depth.
subject_in_directory(s,d)	Check if subject s is in directory d. This does not guarantee that either s or d exists, only that based on the name one would be in the other.

subject_is_group(s)
subject_is_user(s)
subject_is_role(s)
	Check if the subject of a user group or role.

action_is_privilege(a)
action_is_role(a)
	Check if the action is a privilege or role

Authorization Queries

Table 2-6 describes when contextual data is used to define administrative access. This data that may be referenced when writing policies to protect the administration console.

Table 2-6 Context Attributes and Administrative Access

Admin Resource	Privilege	Context attributes	Description
Declaration/Attribute	create	declaration	Queried when user attempts to create a new attribute declaration.
	delete	declaration	Queried when user attempts to delete an attribute declaration.
	rename	declaration, new_name	Queried when user attempts to rename an attribute declaration.
	modify	declaration	Queried when user attempts to modify an attribute declaration.

Table 2-6 Context Attributes and Administrative Access

Admin Resource	Privilege	Context attributes	Description
Declaration/ Constant	create	declaration, value	Queried when user attempts to create a new constant.
	delete	declaration, value	Queried when user attempts to delete a constant.
	rename	declaration, value, new_name	Queried when user attempts to rename a constant.
	modify	declaration, value, new_value	Queried when user attempts to modify a constant.
Declaration/ Enumeration	create	declaration, value	Queried when user attempts to create a new enumeration.
	delete	declaration, value	Queried when user attempts to delete an enumeration.
	rename	declaration, value, new_name	Queried when user attempts to rename an enumeration.
	modify	declaration, value, new_value	Queried when user attempts to modify an enumeration.
Declaration/Evaluation Function	create	declaration	Queried when user attempts to create an evaluation function.
	delete	declaration	Queried when user attempts to delete an evaluation function.
	rename	declaration, new_name	Queried when user attempts to rename an evaluation function.
Identity/Directory/Instance	create	directory	Queried when user attempts to create a directory.
	delete	directory	Queried when user attempts to delete a directory.
	cascade Delete	directory	Queried when user attempts to delete a directory and all its users.
	rename	directory, new_name	Queried when user attempts to rename a directory.

Table 2-6 Context Attributes and Administrative Access

Admin Resource	Privilege	Context attributes	Description
Identity/Directory/ AttributeMapping/Single	create	attribute, default_value, directory	Queried when user attempts to add a scalar attribute to an attribute schema of a directory.
	delete	attribute, default_value, directory	Queried when user attempts to delete a scalar attribute from an attribute schema of a directory.
	modify	attribute, default_value, directory, new_default_value	Queried when user attempts to modify a scalar attribute in an attribute schema for a directory.
Identity/Directory/ AttributeMapping/List	create	attribute, default_value, directory	Queried when user attempts to add a vector attribute to an attribute schema of a directory.
	delete	attribute, default_value directory	Queried when user attempts to delete a vector attribute from an attribute schema of a directory.
	modify	attribute, default_value, directory, new_default_value	Queried when user attempts to modify a vector attribute in an attribute schema of a directory.
Identity/Subject/User	create	subject_name	Queried when user attempts to create a new user.
	copy	subject_name, new_subject_name	Queried when user attempts to copy a user.
	delete	subject_name	Queried when user attempts to delete a user.
	cascade Delete	subject_name	Queried when user attempts to cascade a user and all policies associated with the user.
	rename	subject_name, new_subject_name	Queried when user attempts to rename a user.
Identity/Subject/Group	create	subject_name	Queried when user attempts to create a new group.
	delete	subject_name	Queried when user attempts to delete a group.
	rename	subject_name, new_subject_name	Queried when user attempts to rename a group.
	addMember	subject_name, member_subject	Queried when user attempts to add a member to a group.
	remove Member	subject_name, member_subject	Queried when user attempts to remove a member from a group.

Table 2-6 Context Attributes and Administrative Access

Admin Resource	Privilege	Context attributes	Description
Identity/Subject/ AttributeAssignment/Single	create	attribute, value, subject_name	Queried when user attempts to set a value to a currently unset scalar subject attribute.
	delete	attribute, value, subject_name	Queried when user attempts to unset a currently set scalar subject attribute.
	modify	attribute, value, subject_name, new_value	Queried when user attempts to modify the value of a currently set scalar subject attribute.
Identity/Subject/ AttributeAssignment/List	create	attribute, value, subject_name	Queried when user attempts to set a value to a currently unset vector subject attribute.
	delete	attribute, value, subject_name	Queried when user attempts to unset a currently set vector subject attribute.
	modify	attribute, value, subject_name, new_value	Queried when user attempts to modify the value of a currently set vector subject attribute.
Identity/Subject/ Password	modify	subject_name	Queried when user attempts to modify the password for a user. The subject_name attribute contains the name of the user for which the password is associated.
Resource/Instance	create	resource, resource_type	Queried when user attempts to create a new resource.
	delete	resource	Queried when user attempts to delete a resource.
	cascade Delete	resource	Queried when user attempts to cascade delete a resource. This includes deletion of all child resources and associated policies.
	rename	resource, new_name	Queried when user attempts to rename a resource.
Resource/Attribute Assignment/Single	create	attribute, resource, value	Queried when user attempts to set a value to a currently unset scalar resource attribute.
	delete	attribute, resource, value	Queried when user attempts to unset a currently set scalar resource attribute.
	modify	attribute, resource, value, new_value	Queried when user attempts to modify the value of a currently set scalar resource attribute.

Table 2-6 Context Attributes and Administrative Access

Admin Resource	Privilege	Context attributes	Description
Resource/Attribute Assignment/List	create	attribute, resource, value	Queried when user attempts to set a value to a currently unset vector resource attribute.
	delete	attribute, resource, value	Queried when user attempts to unset a currently set vector resource attribute.
	modify	attribute, resource, value, new_value	Queried when user attempts to modify the value of a currently set vector resource attribute.
Resource/MetaData/ IsApplication	modify	resource, value, new_value	Queried when user attempts to toggle the "is application" resource metadata.
Resource/MetaData/ IsDistributionPoint	modify	resource, value, new_value	Queried when user attempts to toggle the "is distribution point" resource metadata.
Resource/MetaData/ Logical Name	create	logical_name, resource	Queried when user attempts to create a logical name for a resource.
	delete	logical_name, resource	Queried when user attempts to delete a logical name for a resource.
	rename	logical_name, resource, new_name	Queried when user attempts to rename a logical name for a resource.
Policy/Grant	create	action, resource, subject_name, constraint	Queried when user attempts to create a new grant policy. "action", "resource", and "subject_name" attributes are lists.
	delete	action, resource, subject_name, constraint	Queried when user attempts to delete a grant policy. The "action", "resource", and "subject_name" attributes are lists.
	modify	action, resource, subject_name, constraint, new_action, new_resource, new_subject_name, new_constraint	Queried when user attempts to modify a grant policies "action", "resource", and "subject_name" attributes are lists.

Table 2-6 Context Attributes and Administrative Access

Admin Resource	Privilege	Context attributes	Description
Policy/Deny	create	action, resource, subject_name, constraint	Queried when user attempts to create a new deny policy. “action”, “resource”, and “subject_name” attributes are lists.
	delete	action, resource, subject_name, constraint	Queried when user attempts to delete a deny policy. The “action”, “resource”, and “subject_name” attributes are lists.
	modify	action, action_type, resource, subject_name, subject_type, constraint, new_effect, new_action, new_action_type, new_resource, new_subject_name, new_subject_type, new_constraint	Queried when user attempts to modify a deny policy. The “action”, “resource”, and “subject_name” attributes are lists.
Policy/Delegate	create	action, resource, subject_name, delegator, constraint	Queried when user attempts to create a new delegate policy. “action”, “resource”, and “subject_name” attributes are lists.
	delete	action, resource, subject_name, delegator, constraint	Queried when user attempts to delete a delegate policy. The “action”, “resource”, and “subject_name” attributes are lists.
	modify	action, resource, subject_name, delegator, constraint, new_action, new_resource, new_subject_name, new_delegator, new_constraint	Queried when user attempts to modify a delegate policy. The “action”, “resource”, and “subject_name” attributes are lists.
Policy/Action/Role/ Instance	create	action	Queried when user attempts to create a new role.
	delete	action	Queried when user attempts to delete a role.
	rename	action, new_name	Queried when user attempts to rename a role.
Policy/Action/ Privilege/Instance	create	action	Queried when user attempts to create a privilege.
	delete	action	Queried when user attempts to delete a privilege.
	rename	action, new_name	Queried when user attempts to rename a privilege.

Table 2-6 Context Attributes and Administrative Access

Admin Resource	Privilege	Context attributes	Description
Policy/Action/ Privilege/Group	create	action_group	Queried when user attempts to create a privilege group.
	delete	action_group	Queried when user attempts to delete a privilege group.
	rename	action_group, new_name	Queried when user attempts to rename a privilege group.
	addMember	action_group, action	Queried when user attempts to add a privilege to a privilege group.
	remove Member	action_group, action	Queried when user attempts to remove a privilege from a privilege group.
Policy/Analysis/ Inquiry Query	create	title, owner, effect_type, subjects, actions, resources, delegator	Queried when user attempts to create a new policy query.
	delete	title, owner	Queried when user attempts to delete a policy query.
	modify	title, owner, effect_type, subjects, actions, resources, delegator	Queried when user attempts to modify a policy query.
	execute	title, owner, effect_type, subjects, actions, resources, delegator	Queried when user attempts to execute a policy query. If this is an unsaved query "title" and "owner" will be set to an empty string.
Policy/Analysis/ Verification Query	create	title, owner, actions, resources	Queried when user attempts to create a new policy verification query.
	delete	title, owner	Queried when user attempts to delete a policy verification query.
	modify	title, owner, actions, resources	Queried when user attempts to modify a policy verification query.
	execute	title, owner, actions, resources	Queried when user attempts to execute a policy verification query. If this is an unsaved query "title" and "owner" will be set to an empty string.

Table 2-6 Context Attributes and Administrative Access

Admin Resource	Privilege	Context attributes	Description
Policy/Repository	deploy Update	resource, directory	Queried when user attempts to deploy a policy update. "resource" is the distribution node and all nodes below it may be effected. This check is made for each chosen distribution point.
	deploy Structural Change	deleted_directories, deployed_engines, deleted_engines, deleted_bindings, deleted_applications	Queried when user attempts to deploy a structural change.
Infrastructure/Engines/ ARME	create	engine	Queried when user attempts to create a new Security Service Module.
	delete	engine	Queried when user attempts to delete a Security Service Module.
	rename	engine, new_name	Queried when user attempts to rename a Security Service Module.
	bind	engine, resource	Queried when user attempts to bind a resource to a Security Service Module.
	unbind	engine, resource	Queried when user attempts to unbind a resource from a Security Service Module.
Infrastructure/Engines/ SCM	create	engine	Queried when user attempts to create a Service Control Manager.
	delete	engine	Queried when user attempts to delete a Service Control Manager.
	rename	engine, new_name	Queried when user attempts to rename a Service Control Manager.
	bind	engine, resource	Queried when user attempts to bind a Security Service Module to a Service Control Manager. The "resource" contains the name of the Security Service Module.
	unbind	engine, resource	Queried when user attempts to unbind a Security Service Module from a Service Control Manager. The "resource" contains the name of the Security Service Module.

Table 2-6 Context Attributes and Administrative Access

Admin Resource	Privilege	Context attributes	Description
Infrastructure/ Management/Console	login		Queried when user attempts to login to the Administration Console.
Infrastructure/ Management/BulkManager	login		Queried when user attempts to login to the Policy Import tool.

Enumerated Types

Table 2-7 lists the name of each enumerated type used in controlling administrative access.

Table 2-7 Enumerated Types

Name	Values	Description
attribute_usage_type_enum	(resource_attribute, subject_attribute, dynamic_attribute)	Specifies the valid usage for attributes.
subject_type_enum	(user_subject, group_subject, role_subject)	Specifies the valid subject types.
action_type_enum	(privilege_action, role_action)	Specifies the valid action types.
resource_type_enum	(organizational_node, binding_node, resource_node)	Specifies the valid resource types.
effect_type_enum	(grant_effect, deny_effect, delegate_effect)	Specifies the valid role mapping and authorization effect types.

ALES Identities

The table below shows the default ALES roles, users, and groups and some of their administrative rights as determined by existing policies.

Table 2-8 Default ALES Role Privileges and Identities

Role	Privileges / Resources	User/ Groups
Admin	Has all privileges, including creating and managing resources, identities, configurations, starting/stopping ALES servers, etc.	`System` (User)
Deployer	Privileges include modifying SCM/SSM configurations, deploying configuration and policy data, and running policy inquiries.	None

Table 2-8 Default ALES Role Privileges and Identities

Role	Privileges / Resources	User/ Groups
Operator	Privileges include managing SCM/SSM configurations, starting /stopping Administration Server, and running policy inquiries.	None
Monitor	This role effectively provides read-only access to the Administration Console. Privileges include monitoring Administration Console activities and viewing SCM/SSM configurations.	None
Everyone	Change password, access the Console login page, access unprotected resources and operations	Allusers(Group)

Anonymous	No privileges. Does not allow access to ASI resources. This role is automatically assigned to all unauthenticated users.	Anonymous(User)
		Allusers(Group)

Default Role Mapping Policies

The default role mapping policies are described in Table 2-9 below. There are two ways they can be viewed in the Administration Console:

- To see role mapping policies assigned to a specific ALES resource, navigate to and select the resource in the ASI resource tree. Then click Role Mapping Policy Inquiry in the lower right page.
- To see role mapping policies assigned to a specific ALES role, expand the Identity node and select the Role node. Then select the role in the right page and click Role Mapping Policy Inquiry.
- To see all role mapping policies, expand the Policy node in the navigation tree and select Role Mapping Policies.

Of particular note, one of the role mapping policies assigns the Admin role to the user named System. This is the only administrative user provided when ALES is installed.

Table 2-9 Default Role Mapping Policies

Policy	Description
grant(//role/Everyone, //app/policy/ASI, //sgrp/asi/allusers/) if true;	Assigns Everyone role to allusers (group).
grant(//role/Admin, //app/policy/ASI, //user/asi/system/) if true;	Assigns Admin role to system (user).
grant(//role/Anonymous, //app/policy/ASI, //user/asi/anonymous/);	Assigns Anonymous role to anonymous (user)

Default Authorization Policies

A number of authorization policies are provided that define access to ALES components. Some of the more important default authorization policies are described in Table 2-10 below.

Table 2-10 Default Authorization Policies

Default Policy	Description
grant(//priv/delete, //app/policy/ASI/admin, //role/Admin) if true;	Allows Admin role to delete policies.
grant(//priv/cascadeDelete, //app/policy/ASI/admin, //role/Admin) if true;	Allows Admin role to perform cascadeDelete on children of ASI/admin.
grant(//priv/rename, //app/policy/ASI/admin, //role/Admin) if true;	Allows Admin role to rename children of ASI/admin.
grant(//priv/deployStructuralChange, //app/policy/ASI/admin/Policy/Repository, //role/Admin) if true;	Allows Admin role to deploy structural changes.
grant(//priv/login, //app/policy/ASI/admin/Infrastructure/Management/BulkManager, //role/Admin) if true;	Allows Admin role to use the policy loader tool.
grant(//priv/copy, //app/policy/ASI/admin/Identity/Subject/User, //role/Admin) if true;	Allows Admin role to copy users.
grant([//priv/bind,//priv/unbind], //app/policy/ASI/admin/Infrastructure/Engines, //role/Admin) if true;	Allows Admin role to bind/unbind resources, and configure authorization and role mapping provider combinations and SCMs.
grant(//priv/deployUpdate, //app/policy/ASI/admin/Policy/Repository, [//role/Admin,//role/Deployer]) if true;	Allows Admin and Deployer roles to deploy policy updates.
grant(//priv/modify, //app/policy/ASI/admin, [//role/Admin,//role/Deployer]) if true;	Allows Admin and Deployer roles to children of ASI/admin (resources, identities, policies, etc.)
grant(//priv/view, //app/policy/ASI/admin, [//role/Admin,//role/Monitor,//role/Operator,//role/Deployer]) if true;	Allows Admin, Monitor, Operator, and Deployer roles to view children of ASI/admin.
grant(//priv/listAll, //app/policy/ASI/admin, [//role/Admin,//role/Monitor,//role/Operator,//role/Deployer]) if true;	Allows Admin, Monitor, Operator, and Deployer roles to perform the listAll on children of ASI/admin.

Default Policy	Description
`grant(//priv/modify,` //app/policy/ASI/admin/Identity/Subject/ Password, //role/Everyone) if subject_name = sys_user_q;	Allows Everyone to modify their own password.

grant(//priv/create, [//app/policy/ASI/admin/Declaration, //app/policy/ASI/admin/Identity, //app/policy/ASI/admin/Infrastructure, //app/policy/ASI/admin/Resource], //role/Admin) if true;
grant(//priv/create, [//app/policy/ASI/admin/Policy/Action, //app/policy/ASI/admin/Policy/Analysis, //app/policy/ASI/admin/Policy/Rule/Delegate, //app/policy/ASI/admin/Policy/Rule/Grant], //role/Admin) if true;
	Allows Admin role to create policies.

grant([//priv/create,//priv/modify, //priv/view], //app/policy/ASI/admin/Policy/Analysis, [//role/Admin,//role/Monitor, //role/Operator,//role/Deployer]) if owner = sys_user_q;	Allows Admin, Monitor, Operator and Deployer roles to query ALES policies they own.
grant(//priv/execute, //app/policy/ASI/admin/Policy/Analysis, [//role/Admin,//role/Monitor,//role/Operator,//role/Deployer]) if owner = sys_user_q or owner = "";	Allows Admin, Monitor, Operator and Deployer roles to query both policies they own and policies with no owner.
grant([//priv/addMember,//priv/ removeMember], //app/policy/ASI/admin, [//role/Deployer]) if true;	Allows Deployer role to add and remove members to subject and privilege groups.

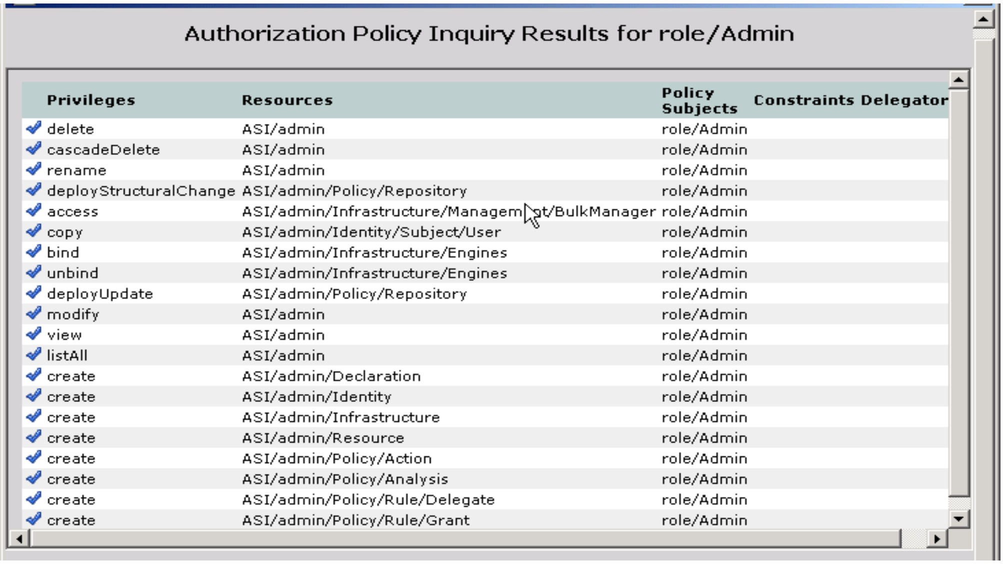
Viewing Authorization Policies

There several ways to view authorization policies in the Administration Console:

- To see authorization policies set on a specific ALES resource, navigate to and select the resource in the ASI resource tree. Then click Authorization Policy Inquiry in the lower right page.
- To see authorization policies set on a specific ALES role, expand the Identity node and select the Role node. Then click Authorization Policy Inquiry in the lower right page.
- To see all authorization policies, expand the Policy node and select Authorization Policies.

Figure 2-2 below shows the results of an authorization policies query on the Admin role.

Figure 2-2 Authorization Policy Inquiry Results Dialog



Authorization Policy Inquiry Results for role/Admin

Privileges	Resources	Policy Subjects	Constraints	Delegator
delete	ASI/admin	role/Admin		
cascadeDelete	ASI/admin	role/Admin		
rename	ASI/admin	role/Admin		
deployStructuralChange	ASI/admin/Policy/Repository	role/Admin		
access	ASI/admin/Infrastructure/Management/BulkManager	role/Admin		
copy	ASI/admin/Identity/Subject/User	role/Admin		
bind	ASI/admin/Infrastructure/Engines	role/Admin		
unbind	ASI/admin/Infrastructure/Engines	role/Admin		
deployUpdate	ASI/admin/Policy/Repository	role/Admin		
modify	ASI/admin	role/Admin		
view	ASI/admin	role/Admin		
listAll	ASI/admin	role/Admin		
create	ASI/admin/Declaration	role/Admin		
create	ASI/admin/Identity	role/Admin		
create	ASI/admin/Infrastructure	role/Admin		
create	ASI/admin/Resource	role/Admin		
create	ASI/admin/Policy/Action	role/Admin		
create	ASI/admin/Policy/Analysis	role/Admin		
create	ASI/admin/Policy/Rule/Delegate	role/Admin		
create	ASI/admin/Policy/Rule/Grant	role/Admin		

Setting Up Application Security Administrators

ALES allows you to set up application-level administrators who are responsible for managing the security for a specific application. The application-level administrator will be able to manage the policies protecting resources belonging to that application, but no others. This chapter describes some basic steps for establishing an application-level security administrators and provisioning them with an initial framework for protecting applications. This section provides information on the following topics:

- "Overview" on page 3-1
- "Create Administrative Users" on page 3-2

Overview

Although the design of the administrative model will vary by use, it is presumed that the task of defining policies to secure an application will be assigned to application-level administrator who have complete rights only for the specific application.

The basic procedure described here for setting up an application-level administrator is to create a parent application resource that will contain a representation of the application in the resource tree, create administrator user accounts and groups as needed, and then use policies that will allow the administrators to manage the application's security.

Establishing a Resource Parent for the Application

To represent an application in ALES, create an ‘binding application’ resource to serve as the application parent. Then give the application security administrator the right to build resources under this parent.

To create an ‘binding application’ resource for an application:

1. Select the Resource node in the navigation tree to display the current resource tree in the right page.
2. Right-click the top parent resource that will contain the application and select Add Resource.
3. Enter a resource name and select Binding in Type field. Then click OK.
4. Right-click the new resource and select Configure Resource.
5. Select Binding Application in the Type field and click OK.

Create Administrative Users

User accounts are needed for the application security administrators. When desired, you may create application-specific directories containing users and groups for the application.

Note: An implicit group named `allusers` is automatically added to all directories.

Identity Directories

To create a separate directory for an application’s users/groups:

1. Select the Identity node in the navigation tree to display the current directories in the right page. After ALES is installed, there is one directory named ASI.
2. Click on New in the lower right page.
3. On the Create Directory dialog, enter the directory name and click OK.

Users and Groups

To add a user or group to a directory:

1. Select the Identity node in the navigation tree to display the current directories in the right page.

2. Click on the directory where you want to add the user or group, then select Edit Users or Edit Groups at the bottom of the page. This displays the directory’s groups or users depending on your selection.
3. Select New at the bottom of the page.
4. On the dialog that displays, enter the user or group name and select OK.

Policies

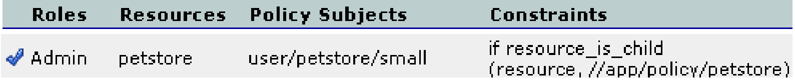
Once the application parent is defined in the resource tree and the necessary identifies have been created, you can use policies to determine administrative access to the application. A number of ways to do this is shown below.

Note: A comprehensive understanding of this process can be obtained by examining the policies already in place for ALES components.

- Using policy constraints allows you to limit administrative rights. For example, the following policy assigns the Admin role to Joe only for managing resources for the Petstore application.

```
grant(//role/Admin, //app/policy/ASI/admin/Resource, //user/asi/Joe/)
if resource_is_child(resource, //app/policy/Petstore, no);
```

Figure 3-1 Using the Resource_is_Child Constraint



Roles	Resources	Policy Subjects	Constraints
✓ Admin	petstore	user/petstore/small	if resource_is_child (resource, //app/policy/petstore)

- Assign Admin role to Bob (user) for the purpose of performing inquiries on policies set on the Petstore application.

```
grant(//role/Admin, //app/policy/ASI/admin, //user/asi/Bob/) if
sys_defined(resource) and resource_is_child(resource,
//app/policy/Petstore, no);
```

Integrating ALES with Applications

This chapter provides information about ALES's built-in support for integration with specific environments.

- "Overview" on page 4-1
- "Security Service Modules" on page 4-2
- "Web Server SSMs" on page 4-5
- "WebLogic Server SSMs" on page 4-19

Overview

ALES provides a number built-in solutions for integration with the following environments:

- – Microsoft Internet Information Server
- – Apache HTTP Server
- – BEA WebLogic Server
- – BEA WebLogic Portal
- – AquaLogic Data Services Platform (ALDSP) 2.1 (formally Liquid Data).

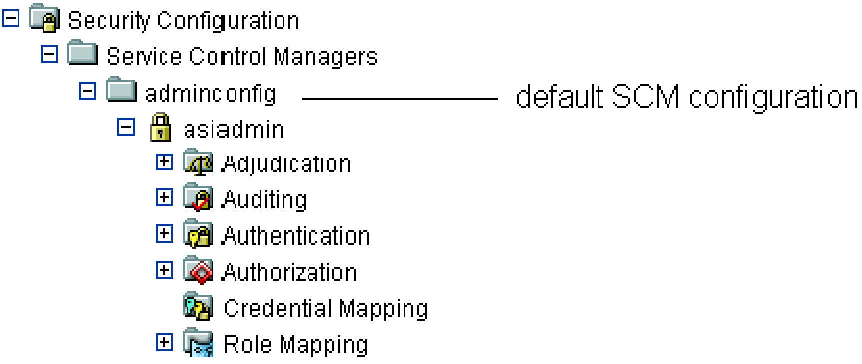
Security Service Modules

Before a SSM can be integrated with a server, a SSM configuration that specifies the security providers must be created and the configuration must be bound to the SCM running on the same machine.

As shown in Figure 4-1, installation of ALES creates a default SCM configuration named `adminconfig` that contains a SSM configuration and security providers used by the Administration Server itself.

If the SSM instance will be located on the same machine, you can use the SCM and create a SSM configuration under it. If on a separate machine, you must create a new SCM. For step-by-step instructions on managing SCM and SSM configurations, see the Management Console help system.

Figure 4-1 Default SCM



To create a SSM configuration:

1. Open the Security Configuration folder.
2. Select Unbound Configurations in the navigation tree and click on `Create a new Security Service Module Configuration` in the right page.
3. On the General tab, complete the following fields and click Create.

Table 4-1 SSM Configuration ID

Field	Description
Configuration ID	This entry must match the SSM config ID that is specified when the SSM instance is created on the server machine. The configuration ID is the means by which the SSM receives it configuration from the SCM.
Description	(Optional) A brief description of the SSM.

SSM Security Providers

The security providers needed depend on the requirements of the application. This section describes the providers included with ALES 2.1. For specific uses of providers with the Web Server SSM, see "Security Providers" on page 4-9. For step-by-step instructions on managing providers, see the Management Console help system.

Table 4-2 Authentication Providers

Provider	Description
Weblogic Authenticator	Authenticate users with WebLogic's embedded LDAP directory.
ALES Identity Asserter	Supports web server authentication and single sign-on between web server SSMs. Use this provider in conjunction with the ALES Credential Mapper.
Database Authenticator	Authenticates users using the ALES relational database provider.
Single Pass Negotiate Identity Asserter	Supports identity assertion using HTTP authentication tokens from the SPNEGO protocol. For more information, see Chapter 6, "Enabling SPNEGO-based Single Sign-on."
SAML Identity Asserter	Accepts SAML assertions sent using the Browser POST Profile and returns the corresponding user. For more information, see Chapter 5, "Enabling SAML-based Single Sign-On."
Open LDAPAuthenticator	Authenticates users using an Open LDAP directory.
Active Directory Authenticator	Authenticates users using Active Directory.
NTAuthenticator	Authenticates users using Windows NT authentication.
iPlanet Authenticator	Authenticates users using an iPlanet LDAP directory.
Novell Authenticator	Authenticates users using a Novell LDAP directory.

Table 4-2 Authentication Providers

Provider	Description
X509 Identity Asserter	Supports identity assertion through an X.509 digital certificate, supporting ASN.1 encoding and decoding
X509 Identity Asserter	Supports identity assertion through an X.509 digital certificate, supporting ASN.1 encoding and decoding

Table 4-3 describes Authorization providers.

Table 4-3 Authorization Providers

Provider	Description
Weblogic Authorizer	Authorizes access to resources based on WebLogic security policy.
ASI Authorization Provider	Authorizes access to resources based on ALES security policy.

Table 4-4 describes Credential Mapping providers.

Table 4-4 Credential Mapping Providers

Provider	Description
Database Credential Mapper	Returns authentication credentials for a user (username and password) from a database.
SAML Credential Mapper	Returns a SAML assertion for an authenticated user. For more information, see Chapter 5, "Enabling SAML-based Single Sign-On."
ALES Identity Credential Mapper	Supports web server authentication and single sign-on between web server SSMs. Returns a ALES assertion for an authenticated user.
Weblogic Credential Mapper	Returns authentication credentials for a user (username and password) from the Weblogic LDAP directory.

Table 4-5 describes Role Mapping providers.

Table 4-5 Role Mapping Providers

Provider	Description
ASI Role Mapper	Returns a set or roles granted to a user on a protected resource based on ALES security policies.
Weblogic Role Mapper	Returns a set or roles granted to a user on a protected resource based on WebLogic security policies.

Web Server SSMs

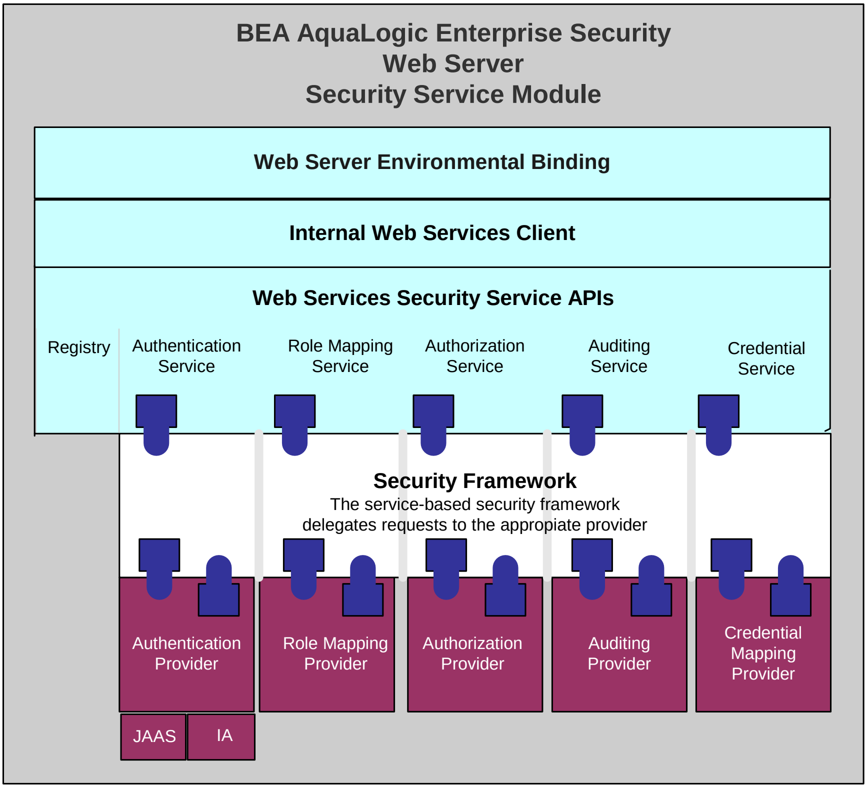
This section covers Web Server SSMs in the following sections:

- "Web Server SSM Overview" on page 4-6
- "Web Server SSM Features" on page 4-10
- "Web Server Constraints and Limitations" on page 4-16
- "Web Server SSM Integration Tasks" on page 4-17

Web Server SSM Overview

An ALES Web Server SSM provides the environmental bindings between the ALES and a web server. It can provide six distinct services: Registry, Authentication, Authorization, Auditing, Role Mapping, and Credential Mapping.

Figure 4-2 Web Server SSM Components

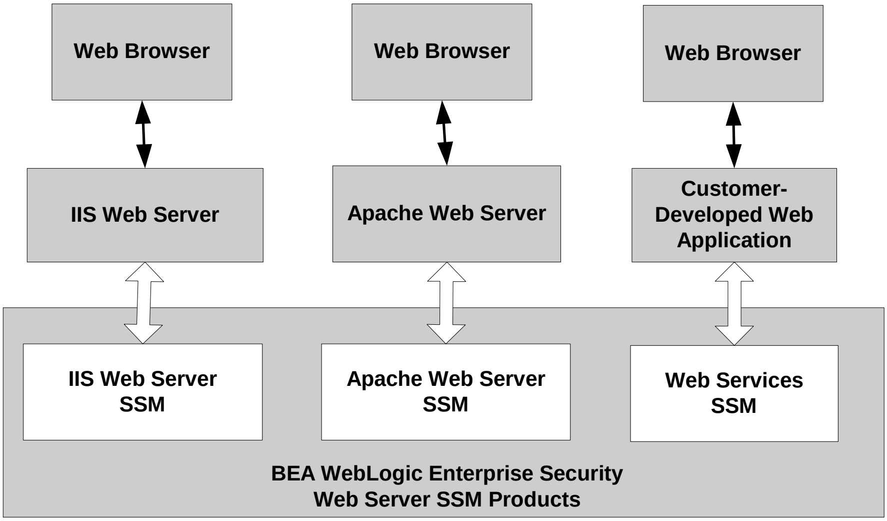


A Web Server SSM makes access decisions for the web server to which it is bound. The security configuration on which the access control decisions are based is defined and deployed by the Administration Server via the Security Control Module.

A Web Server SSM can be tailored to specific needs. Using templates provided as part of the product, security developers can customize the look and feel of authentication pages and configure parameters that allow fine tuning for a particular installation. Web applications can have information added to the HTTP request by the security framework, such as roles and response attributes.

ALES provides three Web Server SSMs: IIS Web Server SSM (SSM), Apache Web Server SSM, and Web Services SSM (see Figure 4-3).

Figure 4-3 Web Server SSM Components



Web Server Environmental Binding

The environmental binding is used to bind to and interact with web servers. Binding a web server SSM to the server projects the ALES subsystem into the web server environment. The SSM accepts HTTPS requests from the web server and presents them to the ALES security framework.

Bindings are provided for two types of web servers: ASF Apache and Microsoft IIS. The second function is ultimately for enforcing access control and providing a means of implementing the SAML Browser/POST profile.

Additionally, the Web Server SSM implements the server-side includes (SSIs) that process SAML Browser/POST profile.

Web Services Security Service APIs

The Web Services Security Service APIs enable access to the ALES security framework. These APIs provide the following security services:

Note: The following topics provide a very brief description of these APIs. For more information, see *Programming Security for Web Services*.

- "Authentication Service" on page 4-8
- "Authorization Service" on page 4-8
- "Auditing Service" on page 4-9
- "Role Mapping Service" on page 4-9
- "Credential Service" on page 4-9

Authentication Service

There are two variations of authentication, JAAS-based and identity assertion. JAAS-based authentication collects evidence (credentials) from a user in order to establish user identity.

Note: For more information on JAAS, see Java Authentication and Authorization Service (JAAS) on the Web at http://java.sun.com/products/jaas/.

Identity assertion authentication consumes a trusted token object to establish identity. The Web Services SSM supports both types of authentication.

- JAAS authentication is highly variable and is dependent upon the configured authentication provider. An authentication provider can use several different types of questions (modeled as callbacks) to collect information from the user.

  Note: The Web Services SSM does not support custom callback types.

- Identity assertion authentication is linked to a specific protocol. For example, X.509 certificate assertion is only valid within the context of a 2-way SSL handshake, and SAML identity assertion is only valid in the context of an Oasis SAML profile. The Web Services SSM implements the Oasis SAML Browser/POST profile and consumes or produces a SAML Identity Assertion.

Authorization Service

In addition to providing a simple permit or denied decision on a URL, the authorization service also has the ability to return attributes into the request as determined by the access control policy implemented. Because the inclusion of coding in the application to handle these attributes creates an undue coupling between the application and security infrastructure, the SSM inserts these returned attributes into the HTTP request header. Depending upon the technology used (ASP, CGI, ISAPI), these headers can be extracted and used by the application.

Auditing Service

The auditing service audits all transactions through the security subsystem. Every URL accessed is sent through the auditing infrastructure.

Role Mapping Service

Although roles are primarily used in authorization, some applications may wish to have access to the roles to which a user is mapped for the purposes of role-based personalization. In order to provide this information to the running applications, the Web Services SSM adds a list of roles to the HTTP request header. Depending upon the technology used (ASP, CGI, ISAPI), the application can extract this list of roles from the header and use it.

Credential Service

The credential service returns sensitive credentials to an application so that the application can use systems that require a secondary (or tertiary) layer of authentication. The Web Services SSM extracts mapped credentials from the security system and makes them available in the HTTP header for use by the application. Depending upon the technology used (ASP, CGI, ISAPI), the application can extract the credential headers and use them to authenticate to other back-end systems.

Security Providers

Installing a SSM deploys a JAR file that contains all ALES security providers. However, before any of the security providers can be used, you must use the Administration Console to configure them. You have the option of configuring either the security providers that ship with the product or custom security providers, which you may develop yourself or purchase from third-party security vendors.

Note: To use security providers with the SSM, you must deploy the security provider MBean JAR file (MJF) to the providers directory on both the machine on which you install the SSM and on the machine on which you install the Administration Console.

The Web Server SSMs use the following security providers:

- ALES Credential Mapping provider
- ALES Identity Assertion provider

  - ASI Adjudication provider
  - ASI Authorization provider
  - ASI Role Mapping provider
  - Database Authentication provider
  - Log4j Auditing provider

For more information on security providers, see "SSM Security Providers" on page 4-3. For more information on how to develop custom security providers, see *Developing Security Providers for BEA AquaLogic Enterprise Security*.

Web Server SSM Features

This section describes the Web Server SSM features in the following sections

- "Web Single Sign-on Capabilities" on page 4-10
- "Authentication Service Features" on page 4-13
- "Authorization Service Features" on page 4-14
- "Auditing Service Features" on page 4-14
- "Role Mapping Features" on page 4-15
- "Credential Mapping Features" on page 4-15
- "Administration Features" on page 4-15
- "Session Management Features" on page 4-16
- "Configuration Features" on page 4-16

Web Single Sign-on Capabilities

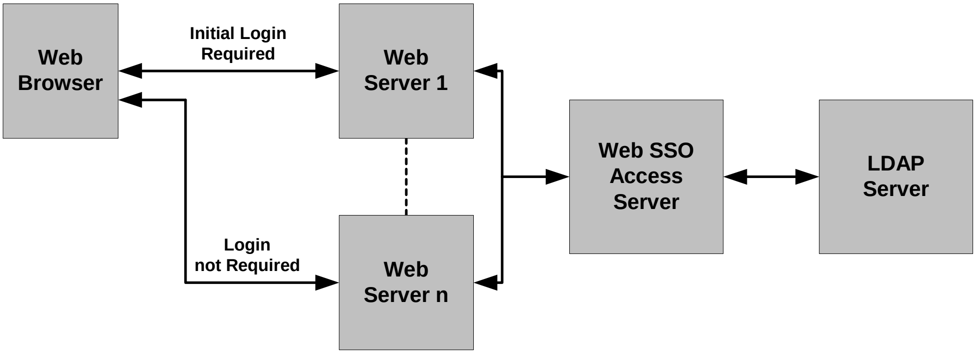
This section covers the following topics:

- "What is Web Single Sign-On?" on page 4-11
- "Single Sign-On Use Cases" on page 4-11
- "Single Sign-On with ALES Identity Assertion" on page 4-12

What is Web Single Sign-On?

Web single sign-on enables users to log on to one web server and gain access to other web servers in the same domain without supplying login credentials again, even if the other web servers have different authentication schemes or requirements. Figure 4-4 shows the basic components of a web single sign-on service.

Figure 4-4 Web Single Sign-on



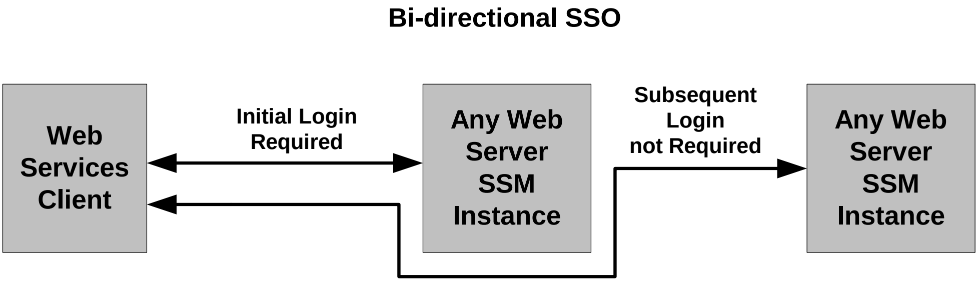
While web single sign-on facilitates access and ease of use, it does not improve security. In fact, security requirements should be considered when implementing a web single sign-on solution.

Single Sign-On Use Cases

The Web Server SSM supports the following single sign-on (SSO) use cases.

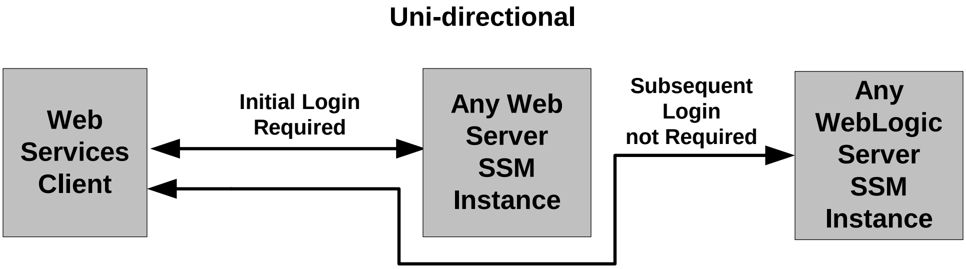
- **Bi-directional SSO among Web Server SSMs**—Once users are authenticated by a Web Server SSM, they are given a identity assertion token that they can use to access other Web Server SSM instances without being required to authenticate again (see Figure 4-5).

Figure 4-5 Web Server SSM to Web Server SSM Single Sing-on


Bi-directional SSO
Web Services Client
Initial Login Required
Any Web Server SSM Instance
Subsequent Login not Required
Any Web Server SSM Instance

- **Uni-directional SSO from Web Server SSM to WebLogic Server 8.1 SSM instances**—Once users are authenticated by a Web Server SSM, they are given an identity assertion token that they can use to access WebLogic Server 8.1 SSM instances without being required to authenticate again (see Figure 4-6).

Figure 4-6 Web Server SSM to WebLogic Server SSM Single Sign-On


Uni-directional
Web Services Client
Initial Login Required
Any Web Server SSM Instance
Subsequent Login not Required
Any WebLogic Server SSM Instance

Single Sign-On with ALES Identity Assertion

The Web Server and WebLogic Server 8.1 SSMs support single sign-on using the ALES Identity Assertion provider. For instructions on how to implement Single Sign-On, see *"Configuring Web Single Sign-on with ALES Identity Assertion"* in *Installing the Web Server and Web Services Security Service Modules*.

Authentication Service Features

The authentication service supports the following features:

- **Conversion of JAAS callbacks to asynchronous**—Although the authentication service is JAAS based, the Web Server SSM masks the JAAS synchronous callback protocol from the web server. When form-based authentication is configured, the credentials are initially gathered and used for authentication. In most cases, an initial form gathers all credentials necessary for authentication.
- **All standard JAAS callbacks**—Sun Microsystems defines seven types of callbacks: NameCallback, PasswordCallback, ChoiceCallback, ConfirmationCallback, LanguageCallback, TextInputCallback, TextOutputCallback.

  Note: If a new callback type is encountered during authentication, the Web Server SSM ignores it.
- **Multiple authentication phases**—JAAS may ask several series of questions using callbacks. Therefore, the SSM does not assume that answering one set of callbacks is sufficient.
- **Web farms**—The following features are supported for web farms:
  - **Redirect during credential gathering**—Within a web farm it is possible for a series of questions (on a form) to start on one machine and be transparently redirected to another machine within that web farm.
  - **Single sign-on**—Within a web farm it is possible that a user is authenticated on one machine and transparently redirected to another. However, a mechanism must be available by which the second machine can accept the identity from the first without having to re-authenticate the user. The identity is only shared within the same cookie domain.
- **Single sign-on with other SSMs**—Can share identity with a custom application that is protected by the Java SSM or another client of the Web Server SSM. The Java SSM and the Web Server SSM use the ALES Credential Mapper and ALES Identity Assertion providers. Single sign-on is limited to a cookie domain.
- **SAML Browser/POST profile**—Consumes an identity assertion from a SAML 1.1 Browser/POST transaction and provides an identity transfer service to serve SAML 1.1 identities to remote systems.
- **Custom, form-based authentication**—Allows for the editing and customizing the forms used in form-based authentication.

Authorization Service Features

The authorization service supports the following features:

- **Resource form**—De-references any use of `".."` and decodes URL encoding. The resource is presented as the path element of a URL and the file or application name. For example, `http://www.bea.com/framework.jsp?CNT=index.htm&FP=/products/aqualogic/` is presented as `/framework.jsp`. The query arguments CNT and FP and associated values are made available in the application context.
- **Allows for unprotected URLs**—Always uses the `isAuthenticationRequired()` method to check if a resource is protected by a security system. This feature is important because you may want to leave some web server resources unprotected.
- **Checks for resource authorization**—When checking for resource authorization, the following features are supported:
  - **Includes rich web service request context**—Because a web application cannot know what elements may be required for enforcement of security at the time of its authoring, it is important that information about the web services request be available in the context given to the security subsystem. The Web Server SSM provides HTTP headers, cookies, query arguments, and form values to the security subsystem. The SSM also decodes all URL encoded context elements before presenting them to the security subsystem.
  - **Works with existing unmodified web applications**—Does not require modification or special code to work with existing web applications running on the web server.
- **Retrieves response attributes during authorization**—Retrieves response attributes during the authorization process and provides them in a form that a layered web application can use.

Auditing Service Features

The auditing service has the following capabilities:

- **Audits all transactions**—All authentications, identity assertions, authorizations, role mappings, credential mappings and audit failures are automatically made available to the auditing infrastructure by the ALES security framework.
- **Audits session cleanup activity**—The occurrence of idle and absolute timeouts are audited.

Role Mapping Features

The role mapping service supports hard-coded roles in applications. Generally hard-coding behavior into an application based on roles is not recommended. It is possible, however, that some customers may need to replace an existing system that uses this mechanism or may want to use roles for user interface personalization. Support for this feature requires that a list of mapped roles available from a security provider for a particular request be provided in a usable form by applications running within the web server.

Note: It is important to note that roles are not global in ALES but can change depending upon the resource and various elements of the context.

Credential Mapping Features

ALES defines two types of credential objects: username/password credentials and generic credentials; however, there is no limitation as to the format of objects that can be used. Credentials can be "mapped" and associated with a resource and identity or an alias.

The credential mapping service has the following features:

- **Provides mapped username/password credential**s—Extracts mapped username/password credentials to the application running within the SSM. This username/password can be used for legacy SSO to log into a database or other system. The Web Server SSM does not use these credentials itself; it will make them available to the web server application.
- **Supports unknown credential types**—Provides a way to inject other credential formats. Since these other formats are unknown to the SSM, they must be converted to a string before being presented to the application.

Administration Features

Administering the security configuration involves writing policies for users, groups, roles, and the web application resources that the SSM protects. The Web Server SSM has the following features:

- **Presents the full URL**—The full URL (including the protocol, server name, port, full path, and query string) is presented to the ALES security framework as part of the context to allow its use in access control policy. Note that the resource presented to the system is in the canonical form. For example, for a web server with the names www.bea.com, www.beasys.com, www.web.internal.bea.com, and 204.236.43.12, the canonical name is www.bea.com.

- **Uses the HTTP method as the action**—The HTTP method (GET, POST, HEAD, PUT) is presented by the Web Server SSM as the action for authorization. In the administration system the privilege must match the action for a policy so this feature allows for separate security policies to be applied to POSTs, GETs, and other methods.
- **Passes in an application context—**The application context is passed through to the SSM's authorization and role mapping security providers and is associated with any audit records logged. This context contains values relevant to the request environment at the time the security provider processes the call.

Session Management Features

To manage session behavior, the Web Server SSM supports the following capabilities:

- **Inactivity timeout**—terminates a session after it has been inactive for a configurable period of time
- **Absolute timeout—**terminates a session after a certain (usually large) period of time, thereby preventing a client from staying perpetually connected, which can be a security risk.
- User logoff—allows for user initiated logoff.
- **Forced logoff**—forces immediate logoff by terminating the session for a single DNS domain.
- **Session cookie**—uses session cookie, not persistent cookies.

Configuration Features

The web server is configured to use the filter component of the Web Server SSM. Local configuration of the web server should only be necessary once and should be static. The Web Server SSM has the following configuration capabilities:

- **Logging channel**—to support configuration and debugging.
- **Configurable flag to control logging and debug mode**—to determine what messages are logged.

Web Server Constraints and Limitations

The Web Server SSM has the following constraints and limitations:

- Does not support cookie-based cross domain single sign-on except through SAML Browser/POST profile.
- Does not support cookie-based cross domain forced logoff.
- To support web farms, local configurations on each web server machine must be manually synchronized.
- Does not support special handling of third-party cookies.
- Requires use of cookies to maintain session state.
- The web server SSM must be manually configured into the web server.
- Does not support load-balancing or failover in accessing the Web Services SSM.
- Must be installed on the same machine as the web server to which it is bound.
- Does not support SAML Browser/Artifact profile.
- Does not preserve the original POST data during redirection to an authentication form.
- Does not save and transfer credentials between machines when more than one machine is involved in an attempt to authenticate a user. Within a web farm, it is possible for a series of questions (on a form) to start on one machine and be transparently redirected to another machine within that web farm. The SSM does not save credentials that have already been entered in the same authentication attempt. Therefore, users are forced to re-enter credential information when more than one machine is involved.

Web Server SSM Integration Tasks

This section provides an overview of integration tasks. Integration tasks center on managing SSM configurations (including the security providers) and configuring the web server to use the web filters. For step-by-step instructions, see the *Installing Web Server and Web Services Security Service Modules*.

The major tasks performed are:

1. Create a SCM and a SSM configuration using the Administration Console. This includes specifying the security providers.
2. Create a parent resource for the application. This will contain ALES's representation of the application.

3. Create the SSM instance on the web server machine and enroll it in the ALES trust environment. The instance will use the security providers defined in step 1 above.

   For detailed instructions on setting up Web Server SSM instances, see the *Installing Web Server and Web Services Security Service Modules*.

   During the instance creation process, the `default.properties` configuration file is created. This file contains the connection information for the ALES services.

4. Configure the web server environmental binding as described in the next section. This loads the web filter on the server and establishes the connection between the web server and ALES.

Web Server Environmental Bindings

This section describes how the IIS and Apache SSMs bind to the web server.

Internet Information Server

To load the web server SSM into IIS:

1. Use the IIS management tool to call the ALES web server filter (wles_isapi.dll).

2. Set authentication to Anonymous and insure that Anonymous user has Read and Read/Execute permissions on the lib, ssl, and config directories where the SSM is installed.

3. Configure server's NamePasswordForm.acc file for:

   ```
   <FORM METHOD=POST ACTION="/<directory>/NamePasswordForm.acc">
   ```

   where <directory> is the location of the acc file.

Apache HTTP Server

To load the web server SSM into Apache HTTP server:

1. Add LoadModule, WLESConfigDir, ServerName, Alias, and Group directives to the server's httpd.conf file to load the SSM when the server is booted. The web filter file name is as follows:

   `mod_wles.dll` (Windows)
   `mod_wles.so` (UNIX)

2. Modify the LD-LIBRARY_PATH statement in the envvars file (in ServerRoot/bin) to ensure that the server loads the dependency libraries for mod_wles.so.

3. Set the Apache ctl script to start or restart the server in the ServerRoot/bin directory.

4. Configure the NamePasswordForm.html file for the Apache Web Server as follows:

   `<FORM METHOD=POST ACTION="/test/NamePasswordForm.html">`

WebLogic Server SSMs

This section provides an overview of integration tasks required for using the WebLogic Server SSM. For step by step instructions, see *Installing WebLogic Server v8.1 Security Service Module*. That guide provides specific steps for integration tasks associated with WebLogic Server, WebLogic Portal, and Aqualogic Data Services.

The WebLogic Server SSM basically provides a means of replacing WebLogic Server's security framework with ALES.

Warning: The WebLogic security framework controls access to it's own administration console as well as the applications it is hosting. When replacing this framework with ALES, you must configure ALES to secure the server's administration console when the WebLogic Server SSM is deployed. If this is not done, the WebLogic administration console will not be accessible.

WebLogic Server SSM Integration Tasks

1. Using the ALES Administration Console, create a SCM and a SSM configuration using, including the necessary security providers.
2. Create a parent Resource node that contains two branches: one for application resources and one for the WebLogic administration console. You can create the branch for the WebLogic administration console by copying the `//asi/console` tree using the Clone function.
3. Create the SSM instance on the web server machine and enroll it in the ALES trust environment. The instance will use the security providers defined in step 1 above.

   For detailed instructions on setting up SSM instances, see the SSM installation guides.

   During the instance creation process, the default.properties configuration file is created for the web server SSM. This file contains the connection information for the ALES services.

4. Modify the WebLogic domain's startup script as described in *Installing WebLogic Server v8.1 Security Service Module*.

WebLogic Server SSM Environmental Bindings

For instructions, see *Installing the WebLogic Server v8.1 Security Service Module*.

Enabling SAML-based Single Sign-On

The ALES provides support for the producing and consuming SAML 1.1 assertions, and for sending/receiving them using the Browser/POST Profile.

This section covers the following topics:

Overview

The use of SAML assertions allows servers in different domains to operate in a federation of trusted servers and grant access to users based on a single login to one of the servers. In a given federation, there are SAML 'producers' and SAML 'consumers'. The SAML providers authenticate users and generate assertions attesting to the user's identity. The SAML assertion can then be included in user requests to other servers in the federation, making additional logins unnecessary.

ALES SSMs allows IIS and Apache HTTP servers to operate as a SAML producer or a SAML consumer (or both) and send/receive SAML assertions using the Browser POST Profile. Note that Browser Artifact Profile is not currently supported.

When set up as a SAML consumer, the SSM running on an IIS or Apache HTTP server will accept requests containing assertions and then use its SAML Identity Asserter to validate the assertions.

Configuring ALES as a SAML Assertion Consumer

When serving as a SAML consumer, the SSM receives requests specifying a protected resource and SAML assertion attesting to the user's validity. The SSM's SAML Identity Asserter accepts the SAML token and returns the corresponding user. ALES will grant access to the resource based on any policies associated with the resource and/or users role.

To configure a IIS or Apache SSM as a SAML consumer:

Note: It is assumed that the necessary Resources and Policies governing access to protected resources have been established.

1. Using the Administration Console, create or use an existing SSM configuration defining a SAML Identity Asserter. The SAML Identity Asserter consumes SAML assertions and returns the corresponding authenticated subjects.

Note: The trusted keystore configured for the SAML Identity Asserter must contain the certificate used to sign the Assertion and the certificate that signed that certificate up to the trust anchor. If the trust anchor is a well known CA such as Verisign, the keystore does not have to contain the trust anchor certificate.

2. Install the ALES SCM and SSM on the IIS or Apache web server.
3. Create instances of the Web Service SSM and the Web Server SSM on the web server and configure the web server to call the SSM. Set the SSMs to use the certificate authority keystore. Set the password for the SSM to use when logging in to the ASI database.
4. Set up a file to serve as the target POST URL. The file can be copied to the web server or referred to using a virtual server. It serves as a placeholder that alerts the SSM to receive a SAML assertion. It must be a valid HTML file, but requires nothing more than empty <HTML> and <BODY> tags.

   Note: The SSM provides a template file named `SAMLIn.acc` (IIS) or `SAMLIn.html` (Apache) in the `templates` directory.

5. In SSM's `default.properties` file, enable the `set saml.incoming.enable` parameter to 'true'.

   Example: `set saml.incoming.enable=true`

6. In SSM's `default.properties` file, set the `saml.incoming.url parameter` to the POST URL you established on the server (see step 4).

   Example: `saml.incoming.url=http://<server>/<dir>/SAMLIn.acc`

Note: Make sure you create a policy that allows POST to the SAML consumer URL.

Configuring ALES as a SAML Assertion Producer

When operating a SAML producer, the SSM will receive requests for a SAML assertion. The SSM's Authentication Provider authenticates the user and its SAML Credential Mapper returns a SAML assertion. The SSM then sends a response contain the SAML assertion using the Browser POST Profile.

By default, SAML assertions produced by ALES are 64-base encoded tokens identifying the principals. They include an XML Signature proving that the assertion has not been tampered with in transit from the sender to the provider, and will contain group information about the user if that information is available. Note that these assertions do not contain the certificate chain used for signing the assertion. It is up to the SAML consumer to notify the recipient of the certificate that can be used to verify the XML signature.

To configure a IIS or Apache SSM as a SAML Producer:

1. Using the Administration Console, create a SSM configuration defining an a Authentication Provider and a SAML Credential Mapper.
2. Install the ALES SCM and SSM on the IIS or Apache web server.
3. Create instances of the Web Service SSM and Web Server SSM on the web server. Set the SSMs to use the certificate authority keystore. Set the password for the SSM to use when logging in to the ASI database.
4. Configure the IIS or Apache web server to integrate with the SSM.
5. Configure the script for handling the Browser POST to the SAML consumer.

Notes: The SSM's `template` directory contains a file named `SAMLXfer.acc` (IIS) or `SAMLXfer.dhtml` (Apache) that can be used.

Make sure you create a policy allowing Everyone access to the script file.

Enabling SPNEGO-based Single Sign-on

This section covers the following topics:

- "Configuring Single Sign-On with Microsoft Clients" on page 6-1
- "Configuring Active Directory Authentication" on page 6-5

Configuring Single Sign-On with Microsoft Clients

Using a Single Pass Negotiate Identity Asserter shipped with AquaLogic Enterprise Security, you can achieve cross-platform authentication, single sign-on (SSO), integration with Microsoft Internet Explorer browser clients and Microsoft .NET web services.

Cross-platform authentication is achieved by emulating the negotiate behavior of native Windows-to-Windows authentication services. The servlet container in this release was modified to handle the necessary header manipulation required by the Windows negotiate protocol, also known as Simple and Protected Negotiate (SPNEGO).

The negotiate identity asserter is an implementation of the Security Service Provider Interface (SSPI) as defined by the WebLogic Security Framework and provides the necessary logic to authenticate a client based on the client's SPNEGO token.

For more information, see the following topics:

- "Requirements" on page 6-2
- "Enabling a Web Service or Web Application" on page 6-3

Requirements

The environment for cross-platform authentication requires the following components.

- Domain controller back-end system
  - Windows 2000 or greater Active Directory
  - Service accounts for mapping Kerberos services
  - Service Principal Names (SPNs) properly configured
  - Keytab files created and inserted, based on platform environment of your server system
- AquaLogic Enterprise Security application server
  - AquaLogic Enterprise Security installed
  - WebLogic 8.1 Security Service Module installed and an instance created
  - WebLogic application server configured to use Active Directory realm
  - Keytab import and configuration
  - MIT Kerberos V5 Generic Security Service API (GSS-API)
- Client systems
  - .NET Framework 1.1
  - Windows 2000 Professional SP2 or greater
  - Internet Explorer 5.01, Internet Explorer v5.5 SP2, Internet Explorer 6.0 SP1
  - Proper configuration of the Internet Explorer browser
  - Proper configuration of web services clients

Note: Client users must be logged on to a Windows 2000 domain or realm, having acquired Kerberos credentials from the Active Directory domain. Local logons do not work.

The following sections describe how to configure the SPNEGO provider and how to set up the necessary components.

- "Enabling a Web Service or Web Application" on page 6-3
- "Configure the Client .NET Web Service" on page 6-7
- "Configure the Internet Explorer Client Browser" on page 6-8

Enabling a Web Service or Web Application

To enable a particular web service, web application, or other protected resource for single sign-on, you must use the Single Pass Negotiate Identity Asserter provider in conjunction with client certification set as the login configuration in your standard J2EE `web.xml` descriptor file.

For configuration instructions, see the following topics:

- "Configuring the SPNEGO Security Provider" on page 6-3
- "Editing the Descriptor File" on page 6-3

Configuring the SPNEGO Security Provider

Configure the Single Pass Negotiate Identity Assertion provider using the Administration Console. To configure the provider, create an instance of the SPNEGO provider for the WebLogic Server 8.1 Security Service Module.

Editing the Descriptor File

Listing 6-1 shows a sample `web.xml file` for a protected WebLogic Web Service resource (Conversation) with the login configuration set to `CLIENT-CERT`.

Listing 6-1 Sample Web.xml File

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
<security-constraint>
       <display-name>Security Constraint on Conversation</display-name>
       <web-resource-collection>
           <web-resource-name>Conversation web service</web-resource-name>
            <description>Only those granted the ConversationUsers role may
             access the Conversation web service.</description>
            <url-pattern>/async/Conversation.jws/*</url-pattern>
            <http-method>GET</http-method>
            <http-method>POST</http-method>
        </web-resource-collection>
        <auth-constraint>
            <role-name>ConversationUsers</role-name>
        </auth-constraint>
    </security-constraint>

    <login-config>
      <auth-method>CLIENT-CERT</auth-method>
    </login-config>

    <security-role>
        <description>Role description</description>
        <role-name>ConversationUsers</role-name>
    </security-role>
</web-app>
```

You can use any role to protect your web resource collection. You want to make the corresponding security and run-as role assignments in your `weblogic.xml` descriptor as needed. Continuing the example, Listing 6-2 shows a sample `weblogic.xml` file.

Listing 6-2 Sample weblogic.xml File

```
<?xml version="1.0" encoding="UTF-8"?>

  <!DOCTYPE weblogic-web-app
    PUBLIC "-//BEA Systems, Inc.//DTD Web Application 7.0//EN"
    "http://www.bea.com/servers/wls700/dtd/weblogic700-web-jar.dtd" >

<weblogic-web-app>
    <security-role-assignment>
        <role-name>ConversationUsers</role-name>
        <principal-name>weblogic</principal-name>
    </security-role-assignment>
</weblogic-web-app>
```

For more information on configuring protected resources for WebLogic Server, see Securing WebLogic Resources in the BEA WebLogic Server 8.1 documentation set available on the Web at `http://e-docs.bea.com/wls/docs81/`.

Configuring Active Directory Authentication

To configure Active Directory authentication, perform the following tasks:

- "Utility Requirements" on page 6-5
- "Configuring and Verifying Active Directive Authentication" on page 6-5
- "Configure the Active Directory Authentication Provider" on page 6-7
- "Configure the Client .NET Web Service" on page 6-7
- "Configure the Internet Explorer Client Browser" on page 6-8

Utility Requirements

This procedure requires the use of the following Active Directory utilities:

- `setspn` – found on the Windows 2000 Resource Kit
- `ktpass` – found on the Windows 2000 distribution CD in `\Program Files\Support Tools`

Configuring and Verifying Active Directive Authentication

The first three steps of this procedure assume that you have two domains: one represents the WebLogic application server domain (`bea.com`) and one controlled by Active Directory (`magellan.corp`).

1. Create a user account for the hostname of the web server machine in Active Directory, by using the Active Directory Users and Computers Snap-in.

   Click Start->Programs->Administrative Tools->Active Directory Users and Computers.

   Use the simple name of the WebLogic server host. For example, if the host you are running the WebLogic application on is called `myhost.bea.com`, create a new user in Active Directory called `myhost`. Do not select "User must change password at next logon." Make a note of the password for use in step 3.

2. Create the Service Principal Names (SPNs) for this account:

```
setspn -A host/myhost.bea.com myhost
setspn -A HTTP/myhost.bea.com myhost
```

3. Create your user mapping and export the keytab files using the `ktpass` utility:

```
ktpass -princ host/myhost@MAGELLAN.CORP -pass <password> -mapuser myhost
-out c:\temp\myhost.host.keytab
```

```
ktpass -princ HTTP/myhost@MAGELLAN.CORP -pass <password> -mapuser myhost
-out c:\temp\myhost.HTTP.keytab
```

Note: If you generated the keytab files for a WebLogic server on a Unix host, copy the keytab files securely to the Unix host. Login as root and then merge them into a single keytab using the ktutil utility:

```
ktutil: "rkt myhost.host.keytab"
ktutil: "rkt myhost.HTTP.keytab"
ktutil: "wkt mykeytab"
ktutil: "q"
```

If your WebLogic server is running on a Windows platform, generate the keytab from that machine using the ktab.

4. Verify that authentication works.
   - To verify that Kerberos authentication is working on the Unix system, run the kinit utility:

     ```
     kinit -t mykeytab myhost
     ```

     You are prompted for the password and if authentication succeeds, the command prompt returns without an error message.
   - To verify that Kerberos authentication is working on a Windows system, use the `ktab` utility locally on the WebLogic server host to create the keytab file in the WebLogic server domain directory:

     ```
     setEnv
     ktab -k mykeytab -a myhost@MAGELLAN.CORP <password>
     ```

5. Create a JAAS login configuration file.

   For either a Windows or a Unix server host, you need a JAAS login configuration file. You also need to set some system properties to direct WebLogic server to allow the proper Kerberos authentication to occur. A sample login configuration file called `krb5Login.conf` looks like this:

```
com.sun.security.jgss.initiate
{
com.sun.security.auth.module.Krb5LoginModule required principal=
"myhost@MAGELLAN.CORP" useKeyTab=true keyTab=mykeytab storeKey=true;
};
com.sun.security.jgss.accept
{
com.sun.security.auth.module.Krb5LoginModule required principal=
```

```
"myhost@MAGELLAN.CORP" useKeyTab=true keyTab=mykeytab storeKey=true;
};
```

6. Add the following system properties to the start line of your WebLogic server:

```
-Djava.security.krb5.realm=MAGELLAN.CORP
-Djava.security.krb5.kdc=ADhostname
-Djava.security.auth.login.config=krb5Login.conf
-Djavax.security.auth.useSubjectCredsOnly=false
```

For a web service client, complete the steps described in "Configure the Client .NET Web Service" on page 6-7. For Internet Explorer configuration, complete the steps described in "Configure the Internet Explorer Client Browser" on page 6-8.

Configure the Active Directory Authentication Provider

To populate groups properly in the authenticated subject and to use the keystores, you must configure the Active Directory Authentication Provider.

Configure the Client .NET Web Service

If you are configuring the client .NET web service, perform the following steps:

1. Open the `web.config` file for the client web service.
2. Set the authentication mode to Windows for IIS and ASP.NET. This is usually the default.

   ```
   <authentication mode="Windows" />
   ```

3. Add the statement needed for the web services client to pass to the proxy web service object so that the credentials are sent through SOAP.

   For example, if you have a web service client for the conversation web service represented by the proxy object `conv`, then setting the web services client credentials in C# looks like this:

   ```
   /*
   * Explicitly pass credentials to the Web Service
   */
   conv.Credentials = System.Net.CredentialCache.DefaultCredentials;
   ```

Configure the Internet Explorer Client Browser

If you are configuring Internet Explorer, perform the following steps:

- "Configure the Sites" on page 6-8
- "Configure Intranet Authentication" on page 6-8
- "Verify the Proxy Settings" on page 6-9
- "Set the Internet Explorer 6.0 Configuration Settings" on page 6-9

Configure the Sites

To configure the sites:

1. In Internet Explorer, click Tools, and then click Internet Options.
2. Click the Security tab.
3. Click Local intranet.
4. Click Sites.
5. Ensure that the Include all sites that bypass the proxy server check box is checked, and then click Advanced.
6. In the Local intranet (Advanced) dialog box, enter all relative domain names that will be used on the intranet (e.g. myhost.bea.com).
7. Click OK to close the dialog boxes.

Configure Intranet Authentication

To configure Intranet Authentication:

1. Click the Security tab, click Local intranet, and then click Custom Level.
2. In the Security Settings dialog box, scroll down to the User Authentication section of the list.
3. Select Automatic logon only in Intranet zone. This setting prevents users from having to re-enter logon credentials; a key piece to this solution.
4. Click OK to close the Security Settings dialog box.

Verify the Proxy Settings

To verify the Proxy Settings:

1. In Internet Explorer, click Tools, and then click Internet Options.
2. Click the Connections tab.
3. Click LAN Settings.
4. Verify that the proxy server address and port number are correct.
5. Click Advanced.
6. In the Proxy Settings dialog box, ensure that all desired domain names are entered in the Exceptions field.
7. Click OK to close the Proxy Settings dialog box.

Set the Internet Explorer 6.0 Configuration Settings

In addition to the previous settings, one additional setting is required if you are running Internet Explorer 6.0.

1. In Internet Explorer, click Tools, and then click Internet Options.
2. Click the Advanced tab.
3. Scroll down to the Security section.
4. Make sure that Enable Integrated Windows Authentication (requires restart) is checked, and then click OK.
5. If this box was not checked, restart the browser.

CHAPTER 7

Configuring Metadirectories

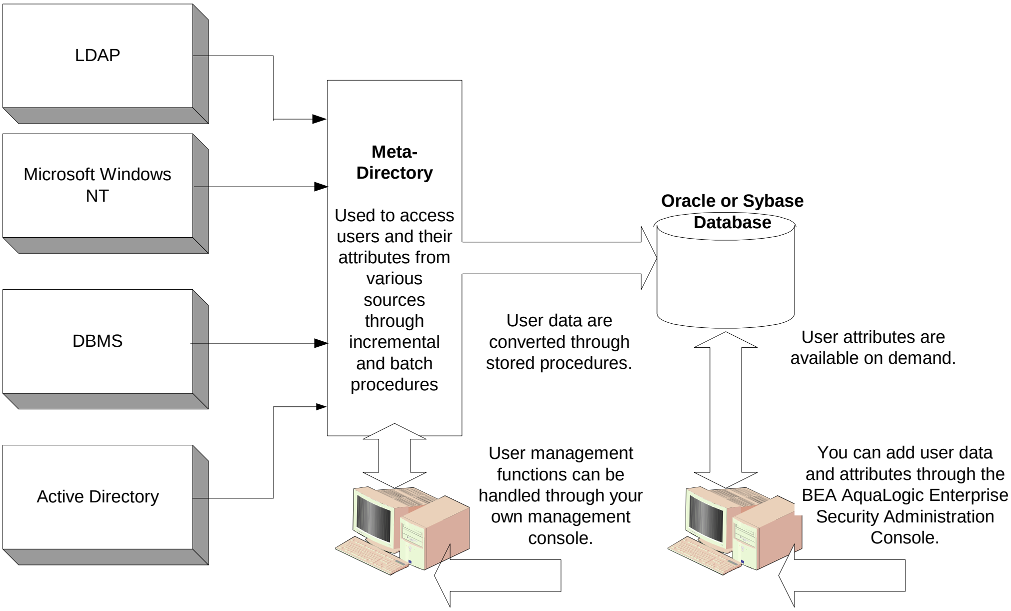
This section describes how to configure a metadirectory to extract user data from your user repository (for example, an LDAP server, an Active Directory, a database server, or NT Domain directory) and import that data into the policy database. As a result, the user, group and attribute data (referred to simply as attributes) are available and synchronized, and can be used to enforce dynamic security policies in your applications through the ASI Authorization and ASI Role Mapping services.

This section covers the following topics:

Why Use Metadirectories?

ALES requires that all user and policy data be stored in either an Oracle or Sybase database. Using metadirectories you can import user identity information from external repositories and thus achieve a unified view of all identity information. This capability enables you to solve important business issues that result from having identity information stored in multiple, disparate data repositories throughout an organization. Thus, through the use of metadirectories, the maintenance cost of sharing information is reduced and the accuracy and the overall security of application resources is improved (see Figure 7-1).

Figure 7-1 Metadirectory Architecture



LDAP
Microsoft Windows NT
DBMS
Active Directory
Meta-Directory
Used to access users and their attributes from various sources through incremental and batch procedures
Oracle or Sybase Database
User data are converted through stored procedures.
User attributes are available on demand.
User management functions can be handled through your own management console.
You can add user data and attributes through the BEA AquaLogic Enterprise Security Administration Console.

In ALES, the term directory applies to any collection of user data, stored in a database, LDAP directory server, or other type of repository. These directories form the core of any identity management solution because every user repository has its own approach to the storage of information.

An identity directory refers to any user repository configured for use with ALES. In the Administration Console, an identity directory defines a logical collection of users, groups, roles, and attributes that can be used to design your role mapping and authorization policies. An identity directory typically represents roles and groups of users of a particular application or resource, users in a specific location, or users imported from an external user repository.

A metadirectory can be used to store attributes replicated and synchronized from your user repository into the policy database. Following replication, the user data are available as attributes through the Administration Console identity directories for use in your role mapping and authorization policies. Any changes that you make to the replicated user attributes using the Administration Console are not propagated back to your user repository.

Metadirectory Configuration Overview

To configure metadirectories, you use several different components (see Figure 7-2). In Figure 7-2, each task is represented by a number that is positioned next to the component that you use to perform the task.

Figure 7-2 Metadirectory Configuration Components

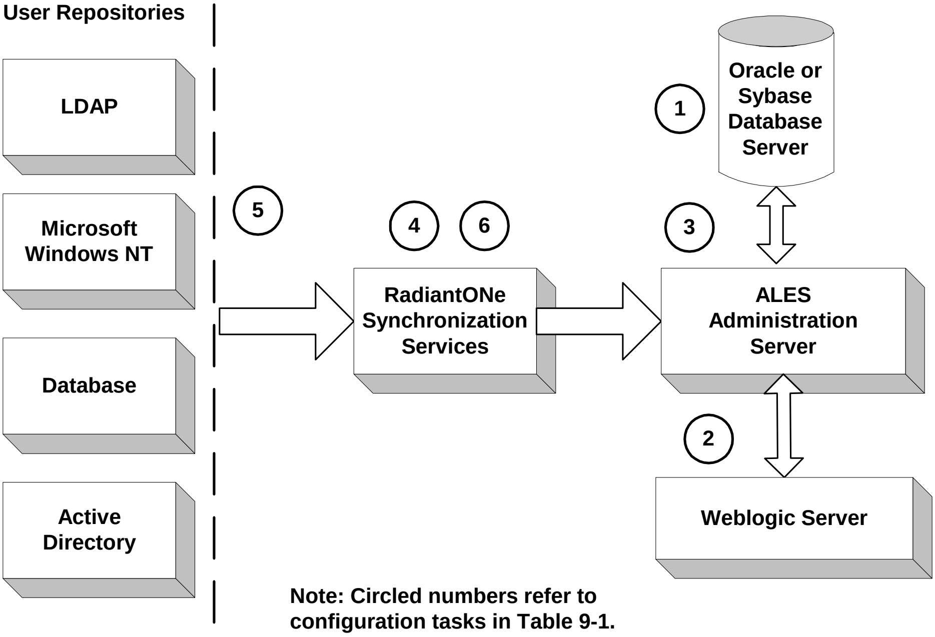

User Repositories
LDAP
Microsoft Windows NT
Database
Active Directory
5
4
6
RadiantONe Synchronization Services
1
Oracle or Sybase Database Server
3
ALES Administration Server
2
Weblogic Server
Note: Circled numbers refer to configuration tasks in Table 9-1.

Table 7-1 summarizes the tasks and links each task to a circled number in Figure 7-2.

Table 7-1 Metadirectory Configuration Tasks

Use this component:	To perform these tasks:
Database Server (Oracle or Sybase)	1. Create the destination tables.
WebLogic Server v8.1 Administration Console	2. Configure a JDBC connection pool and the Java Message Service.
ALES Administration Console	3. Configure database triggers.
RadiantOne Synchronization Services	4. Configure metadirectory schemas and the synchronization hub.
User Repository Server	5. Configure the directory connector. **Note:** This task is required only if you are using Sun ONE Active Directory as your user repository server.
RadiantOne Synchronization Services	6. Configure the connectors and start the synchronization hub and the connectors.

For detailed instructions for installing the metadirectory software on performing each of the tasks listed in Table 7-1, see the following sections:

- "Installing the Metadirectory Software" on page 7-4
- "Configuring Metadirectory Tables and Database Triggers" on page 7-5
- "Configuring Metadirectory Schemas" on page 7-13
- "Configuring Metadirectory Synchronization" on page 7-27

Installing the Metadirectory Software

Before you can configure and use metadirectories, you must install and start the RadiantOne Synchronization Services. RadiantOne Synchronization Services use the JDBC and Java Message Service (JMS) features of the servlet container that hosting ALES to update the metadirectory.

The RadiantOne Synchronization Services tool provides connectors for the master identity repositories that send XML formatted messages whenever information in a user repository is updated. Through the use of the JMS, this tool ensures that all updates are delivered and processed.

The RadiantOne Synchronization Services software is available as a separate installation in the ALES product.

To install the RadiantOne Synchronization software:

- On a Microsoft Windows platform, run: `ales21metadir_win32.exe`.
- On a Sun Solaris platform, run: `ales21metadir_solaris32.bin`.
- On a Linux platform, run: `ales21metadir_rhas21_IA32.bin`.

After installation, access the applications from the Start>Programs>RadiantOne menu.

Note: A second installation package is available for installing the RadiantOne Connectors. These are included in the RadiantOne Synchronization Services install package and do not need to be installed separately. However, you can install the connector package separately on another machine. For instructions for installing the connectors separately, refer to the RadiantOne Synchronization Services installation documentation. When you install the RadiantOne software, the online documentation is installed in the RadiantOne directory

Configuring Metadirectory Tables and Database Triggers

This section covers the tasks that you must perform to create destination tables in the ALES policy database and install triggers on those tables.

To configure metadirectory tables and database triggers, perform the following tasks:

- "Creating Metadirectory Destination Tables" on page 7-5
- "Configuring a JDBC Connection Pool and JMS" on page 7-8
- "Configuring Metadirectory Database Triggers" on page 7-11

Creating Metadirectory Destination Tables

There are two tables that you have to create in the policy database for the synchronization of users and groups to work properly: `ASI_USERS` and `ASI_GROUPS`. After you create these tables, you configure them through the ALES Administration Console. You must create these tables before you perform the remaining tasks in this section.

The following sections provide guidelines and restrictions for the tables and instructions for creating them:

- "Metadirectory Destination Table Guidelines and Restrictions" on page 7-6

- "Creating Metadirectory Destination Tables Using Oracle or Sybase" on page 7-8

Metadirectory Destination Table Guidelines and Restrictions

Two table are required in the policy database for the synchronization of user repositories: `ASI_USERS` and `ASI_GROUPS`. The following sections provide guidelines and restrictions for these tables:

- "User Synchronization Table Guidelines" on page 7-6
- "User Synchronization Table Guidelines" on page 7-6
- "User and Group Attributes Character Set Restrictions" on page 7-7

User Synchronization Table Guidelines

The user synchronization table is used by the partner tool to stage user and user attribute information for import into the policy database. The name of the table used for user synchronization is configurable.

While the names of the columns in the table are configurable, the following restrictions apply:

- One column in the table must serve as the unique identifier for the user. The UID may contain any character, but the '/' character must be escaped. For example, "John\Doe" must be entered as "John\/Doe".
- The Primary Key for the table should be the column used as the UID. For performance and data consistency, the user synchronization table should include the primary key in its definition.

The user synchronization table accommodates source repositories that store group memberships as user attributes. Managing group memberships as user attributes does not impact managing group memberships explicitly through the group synchronization table—both ways can be used.

You must adhere the following restrictions and requirements when setting up group memberships through the user synchronization table:

- Only one column may be used for storing the group memberships.
- The group column needs to be a character string (typically in Oracle: `varchar2`).
- Membership in multiple groups is possible and is stored as a delimited text string. The choice of delimiter is configurable but should be sufficiently uncommon so that parsing of the group list may be done correctly.

- If the group name contains the '/' character, it should be escaped.

Any number of columns in the user synchronization table may be used for passing attributes into the Administration Server. The columns used for all attributes in the User Synchronization table must be of variable length character (for example, in Oracle: varchar2). For purposes of importing from the user synchronization table, you may map attributes to any of the following ALES policy data types: `string`, `integer`, `boolean`, `date`, `time`, `dayofweek_type`, `month_type`, and `object_type`. Attributes are also defined as either `list` or `single`. Multiple attribute values of type `list` are stored as a delimited text string. The delimiter used for attributes of type `list` must be the same as the delimiter used for groups.

Group Synchronization Table Guidelines

In addition to the user-attribute-based group membership discussed above, group memberships may also be defined by using the group synchronization table. Unlike the `User Synchronization table`, the schema for the group synchronization table is fixed, that is, it must adhere to the structure shown in Table 7-2.

Table 7-2 Group Synchronization Table

Column Name	Type	Description
CN	varchar2	The name of the group
UNIQUEMEMBER	varchar2	The name of the user belonging to the group.

You must adhere the following restrictions and requirements when setting up the group synchronization table:

- The Primary Key for the table should consist of both columns.
- A forward slash (/) in the value for either of the columns must be escaped using a back slash (\).

User and Group Attributes Character Set Restrictions

The following requirements and restrictions apply to user and group attributes

- The name of the attribute cannot be longer than 1000 character (580 characters for some Sybase 12.5 configurations, depending on the page size)
- Each value of a user attribute cannot be longer than 1000 characters. (580 characters for some Sybase 12.5 configuration, depending on the page size)

- The length of the value of all user attributes combined cannot be longer than the lesser of 16,000 characters or the `varchar2` column-size limit for the database.
- Attribute names in ALES may only consist of alphanumeric characters (a-z, A-Z, 0-9) and the underscore (_) character.
- The column name of the user synchronization table is limited by any database character set limitations.
- Attribute names must start with an alphabetic character or an underscore.
- Any printable characters are allowed except double quote (“) and back slash (\).

Creating Metadirectory Destination Tables Using Oracle or Sybase

To create the `ASI_USERS` and `ASI_GROUPS` destination tables using an Oracle or a Sybase database server,

1. To log into the policy database, open a command window and type:

   ```
   sqlplus username/password@asi
   ```

   where: *`username`* and *`password`* are the username and password you defined when you created the database user account and *`asi`* is the database instance name.

2. To create the ASI_USERS destination table, enter the following SQL command:

   ```
   SQL> CREATE TABLE ASI_USERS(DisplayName VARCHAR(255) NULL,
          COMMONDOMAIN VARCHAR(255) NOT NULL, PRIMARY KEY (COMMONDOMAIN))
   ```

3. To create the ASI_GROUPS destination table, enter the following SQL command:

   ```
   SQL> CREATE TABLE ASI_GROUPS(CN VARCHAR(255) NOT NULL,
          UNIQUEMEMBER VARCHAR(255) NOT NULL, PRIMARY KEY
           (CN,UNIQUEMEMBER))
   ```

Note: In addition to `DisplayName`, you can add more columns to the destination tables to be used as user attributes, such as street address, zip code, `email`, `phone`, and so on.

Configuring a JDBC Connection Pool and JMS

To connect to the RadiantOne Synchronization Services to the ALES asiDomain, you need to configure a persistent Java Message Service (JMS). To accomplish this, you use the WebLogic Server Administration Console to configure a JDBC connection Pool and the JMS.

To configure the JDBC connection pool and JMS, perform the following steps:

1. To start the WebLogic Server Administration Console, open a browser and go to `https://`*`hostname`*`:7010/console`,

   where:

   `hostname` in the name of the machine that is hosting the ALES Administration Server

   7010 is the port on which the Administration Console is running

2. In the left pane of the Administration Console, open the Services and JDBC folders and click Connection Pools. The asiDomain> JDBC Connection Pools page is displayed in the right pane.
3. Click Configure a new JDBC Connection Pool. The Configure a JDBC Connection Pool: Choose database page is displayed.
4. Select the Database Type and the Database Driver as specified in Table 7-3 and click Continue. The Configure a JDBC Connection Pool: Define connection properties page is displayed.

Table 7-3 Database Type and Database Driver Parameter Settings

JDBC Connection Pool Parameter	Setting
Database Type	Oracle or Sybase
Database Driver	For Oracle 9i, select Oracle’s Driver (Thin) Version: ,9.2.0 For Oracle 10g Release 1, select Oracle’s Driver (Thin) Version: 10g 10.1.0.4 For Sybase, select BEA’s Sybase Driver (Type 4) Versions: 11.X,12.X

5. Refer to Table 7-4, enter the appropriate values in the Configure a JDBC Connection Pool: Define connection properties page, and click Continue. The Configure a JDBC Connection Pool: Test database connection page is displayed.

Table 7-4 JDBC Connection Pool Configuration Parameters

Parameter	Description
Name	The JDBC connection pool name that you specify, for example, ConsolePool.
Database name	The name assigned to the instance of the database when it was created, for example, ASI.

Table 7-4 JDBC Connection Pool Configuration Parameters

Parameter	Description
Hostname	The name of the machine on which the database server is installed, for example, ASI_host.
Port	The port used for the connection to the database server (default: 1521).
Database User Name	The username of the database account, for example, ales.
Password/Confirm Password	The password assigned with the database account was create for the user (any alphanumeric string).

6. Click Test Driver Configuration. A “Connection successful” message and the Configure a JDBC Connection Pool: Create and deploy page is displayed.
7. Click Create and Deploy. The connection pool is deployed.
8. To configure a JMS template, perform these steps:
    a. In the left pane, open the Services and JMS folders and click Templates. The asiDomain> JMS Templates configuration page is displayed in the right pane.
    b. Click Configure a new JMS Template, name the template `RLI_JMS`, and then click Create.
9. To configure a JMS JDBC store, perform these steps:
    a. In the left pane, click Stores.
    b. Click Configure a new JMS JDBC Store.
    c. Name the store `RLI_JDBC_STORE`.
    d. Set the Connection Pool to the name of the connection pool created previously (`ConsolePool`) and select Create.
10. To configure a JMS server, perform these steps:
    a. In the left pane, click Servers.
    b. Click Configure a new JMS Server.
    c. Name the server `RLI_JMS_SERVER`.
    d. Set Persistent Store to the JDBC store that was created previously (`RLI_JDBC_STORE`).

   e. Set Temporary Template to the template that was created previously (`RLI_JMS`), and click Create.
   f. Click the Target an Deploy tab and set Target to asiAdminServer and click Apply.

11. To configure a JMS Connection factory, perform these steps:
   a. In the left pane, click Connection Factories.
   b. Click Configure a new JMS Connection Factory,
   c. Set Name to `RLI_JMS_CONNECTION`,
   d. Set JNDI Name to `weblogic.asiAdminServer.jms.TopicConnectionFactory`, and click Create.
   e. Click the Target an Deploy tab and set Target to `asiAdminServer` and click Apply.
12. To restart the WebLogic Server so that the change takes effect, close the WebLogic Server command window or, if WebLogic Server is setup to run as a Windows service, restart the service.
13. This completes the configuration of the JDBC connection pool and JMS.

Configuring Metadirectory Database Triggers

You must configure database triggers for the user and group synchronization tables in the policy database. A database trigger provides a necessary link between the metadirectory database and the policy database. A trigger enables the user attributes to be received by the Administration Server and put into the identity directory (that you define) whenever a change occurs in the underlying metadirectory database.

Note: Any modifications that you make to the existing data records in the synchronization tables must be made with an `UPDATE` command, not through a series of `DELETE` and `INSERT` commands. Use `INSERT` only for new records and `DELETE` only for removing records. Also, do not use the “truncate table” command to clean either the user or group synchronization tables because that command does not activate the triggers.

To configure metadirectory database triggers, perform the following steps:

1. To start the Administration Console, open a browser and go to to `https://`*`hostname`*`:7010/asi`,

   where:

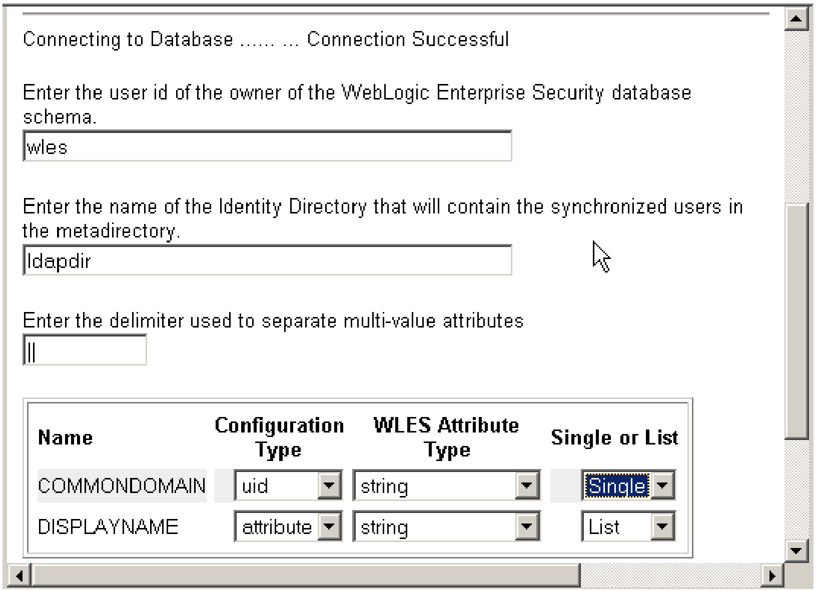
   `hostname` is the name of the machine that is hosting the ALES Administration Server

`7010` is port on which the Administration Console is running

`asi` is the domain name

2. In the left pane, open the Identity folder and click Metadirectory Configuration. The Metadirectory Configuration page is displayed in the right pane.
3. In the Metadirectory Configuration page, select the database type (either Oracle or Sybase), enter the name of the JDBC connection pool (for example: ConsolePool) and the name of the synchronization tables (ASI_USERS and ASI_GROUPS), and click Connect. A "Successful Connection" message is displayed along with additional fields that require input.
4. Refer to Figure 7-3, and fill in the additional fields. Set `user id` to the ALES schema owner, which is the same as the database account username. Set the `identity directory` name field to any directory name that is unique in the asiDomain. This identity directory is the directory in the policy database into which users are populated. Set the `COMMONDOMAIN` and `DISPLAYNAME` parameters as shown in Figure 7-3.
5. Click Install Trigger. A "Trigger successfully installed" message is displayed.
6. This completes the configuration of the metadirectory database triggers.

Figure 7-3 Metadirectory Triggers Configuration



Connecting to Database .... Connection Successful

Enter the user id of the owner of the WebLogic Enterprise Security database schema.

wles

Enter the name of the Identity Directory that will contain the synchronized users in the metadirectory.

ldapdir

Enter the delimiter used to separate multi-value attributes

||

Name	Configuration Type	WLES Attribute Type	Single or List
COMMONDOMAIN	uid	string	Single
DISPLAYNAME	attribute	string	List

Configuring Metadirectory Schemas

ALES uses a comprehensive schema for tracking and updating all policy data. You use RadiantOne Synchronization Services to configure the schemas required to to upload user and groups information from the user repository to the policy database.

To configure the required metadirectory schemas, perform following tasks:

- "Extracting the Source Schemas" on page 7-13
- "Loading the Source Schemas" on page 7-16
- "Extracting the Destination Schemas" on page 7-18
- "Loading the Destination Schemas" on page 7-21
- "Configuring the Source-to-Destination Topology" on page 7-23
- "Configuring the Topology Transformations" on page 7-24
- "UpLoading User and Group Data" on page 7-26

Extracting the Source Schemas

This section describes how to extract the source schemas for the user repository.

To extract the source schemas from the user repository, perform the following steps:

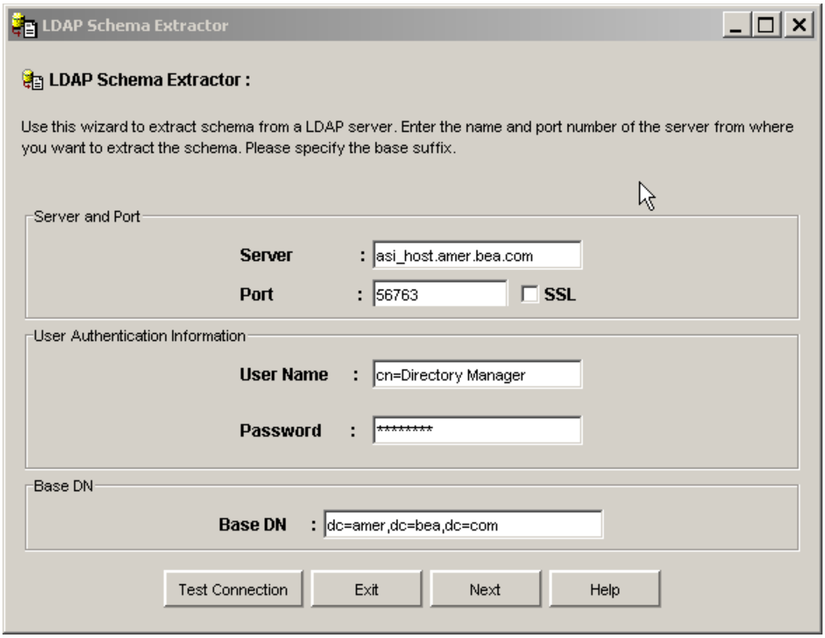
1. Copy the `WL_Home\server\lib\weblogic.jar` file to the `RadiantOne\r1syncsvcs\bea_lib` directory.
2. To start the RadiantOne Synchronization Services tool:

   On Windows: click Start>Programs>RadiantOne>RadiantOne Synchronization Services> Synchronization Services Administrator.

   On Sun Solaris: From the `RadiantOne/r1syncsvcs/bin` directory, run: `runSSC.sh`.
3. To start the Schema Extraction Wizard is displayed, select New from the Datastore drop-down menu.
4. Select LDAP Schema Extraction radio button and click Next. The LDAP Schema Extractor page is displayed.

5. Refer to Table 7-4 and Table 7-5 and enter the directory server information. To determined the complete directory server name and port number, refer to the directory server console (see Figure 7-5) and check the values. For example, in Figure 7-5, the complete name is `asi_host.amer.bea.com` and the port number is `56763`.

Figure 7-4 LDAP Schema Extractor Page



LDAP Schema Extractor :

Use this wizard to extract schema from a LDAP server. Enter the name and port number of the server from where you want to extract the schema. Please specify the base suffix.

Server and Port

Server : asi_host.amer.bea.com

Port : 56763 ☐ **SSL**

User Authentication Information

User Name : cn=Directory Manager

Password : ********

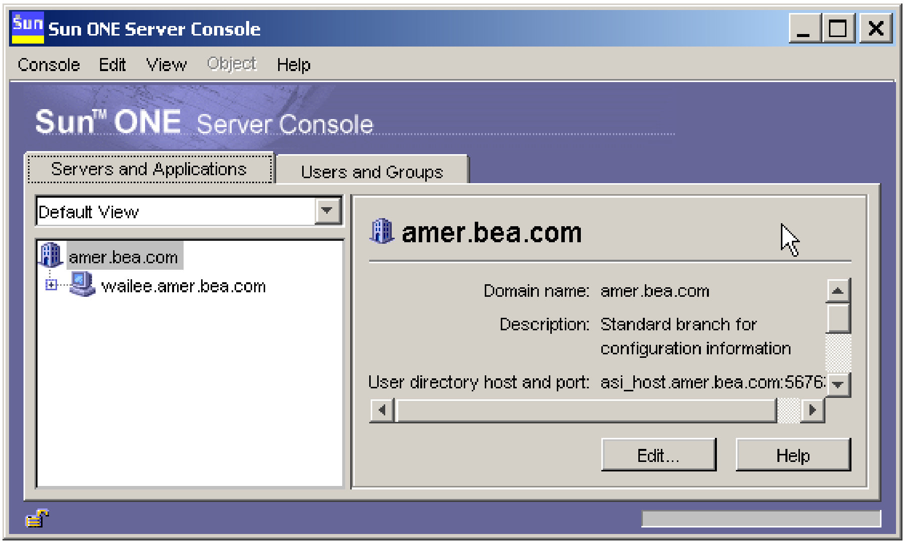
Base DN

Base DN : dc=amer,dc=bea,dc=com

Table 7-5 LDAP Schema Extractor Parameters

Parameter	Description
Server	The name of the LDAP Directory server, for example, `asi_host.amer.bea.com`
Port	The port number of the LDAP Directory server, for example, `56763`
Username	The username you enter to access the LDAP Directory server (default: Directory Manager).
Password	The password you enter to access the LDAP Directory server.
Base DN	The base domain name, for example, `dc=amer, dc=bea, dc=com`.

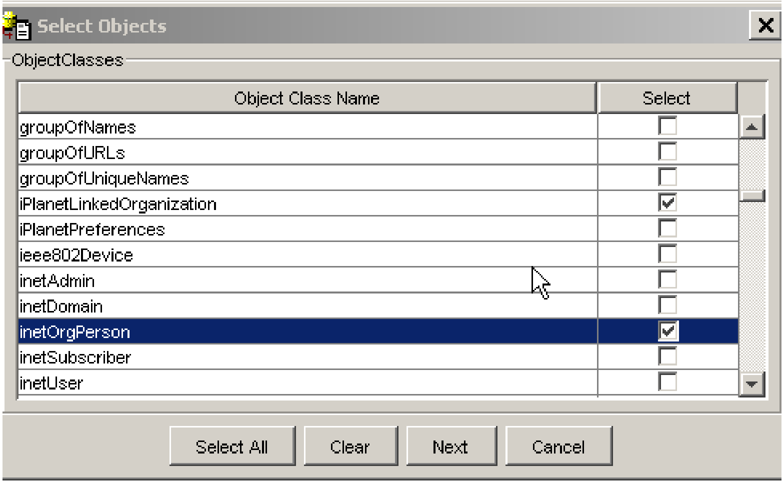
Figure 7-5 Sun ONE Server Console



Domain name: amer.bea.com
Description: Standard branch for configuration information
User directory host and port: asi_host.amer.bea.com:5676:

6. Click Test Connection. A “Connection Successful” message dialog box is displayed.
7. To close the message dialog box, click Ok and then click Next. The Select Objects page is displayed (see Figure 7-6).

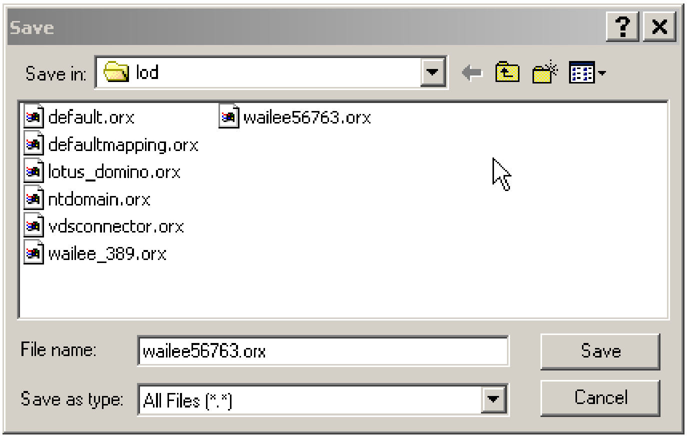
Figure 7-6 RadiantOne Select Objects Page



Object Class Name	Select
groupOfNames	☐
groupOfURLs	☐
groupOfUniqueNames	☐
iPlanetLinkedOrganization	☑
iPlanetPreferences	☐
ieee802Device	☐
inetAdmin	☐
inetDomain	☐
inetOrgPerson	☑
inetSubscriber	☐
inetUser	☐

8. Select the `groupOfUniqeNames` and `inetOrgPerson` object classes and click Next. The Save windows is displayed (see Figure 7-7).

Figure 7-7 RadiantOne Select Object Save Window



Save in: lod

File name: wailee56763.orx

Save as type: All Files (*.*)

9. In the Save window, edit the Filename field to remove all but the directory server name, the port number, and the `.orx` filename extension as shown in Figure 7-7, and click Save. A “Schema Extraction Completed” message dialog box is displayed.

10. . Click Ok and then click Exit.

11. This completes the extraction of the source schemas.

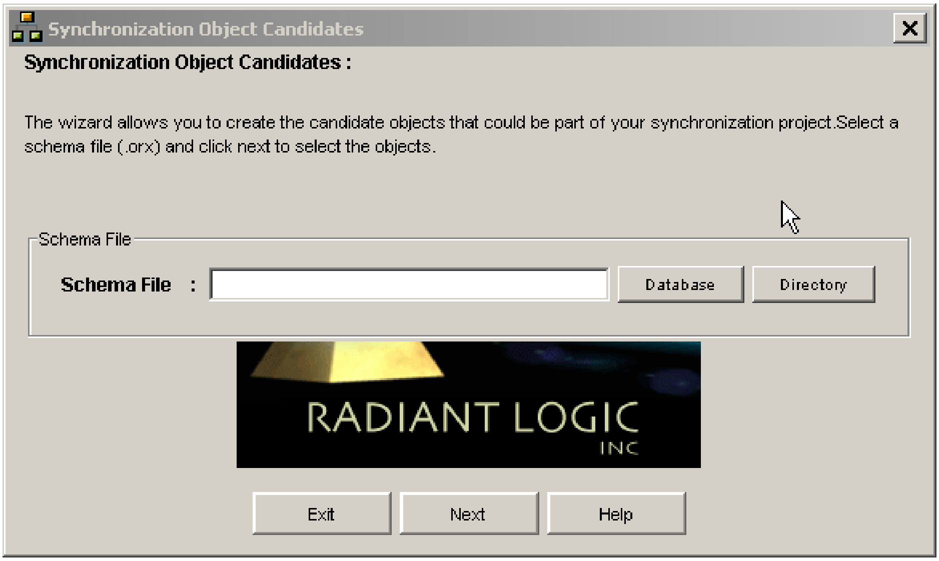
Loading the Source Schemas

This section describes how to load the source schemas for the user repository.

To load the source schemas from the user repository, perform the following steps:

1. On the RadiantOne Synchronization Services Administration console, from the Datastore drop-down menu, select Add. The Synchronization Object Candidates page is displayed (see Figure 7-8).

Figure 7-8 RadiantOne Synchronization Object Candidates Page



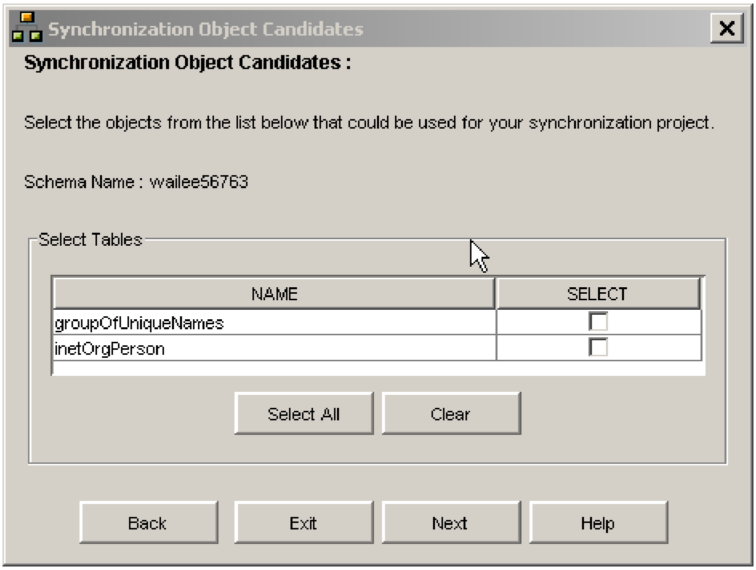
Synchronization Object Candidates :

The wizard allows you to create the candidate objects that could be part of your synchronization project.Select a schema file (.orx) and click next to select the objects.

Schema File :

2. Click Directory. The Open window is displayed.
3. Select schema extraction file that you created in "Extracting the Source Schemas" on page 7-13 (for example: `asi56763.orx`) and click Open. The Schema File field is populated with the filename, including the path.
4. Click Next. The Synchronization Objects Candidates page is displayed (see Figure 7-9)

Figure 7-9 Synchronization Select Objects Page



Synchronization Object Candidates :

Select the objects from the list below that could be used for your synchronization project.

Schema Name : wailee56763

NAME	SELECT
groupOfUniqueNames	☐
inetOrgPerson	☐

5. Click Select All and click Next. A “Successfully generated the datastore” message is displayed.
6. To exit, click Finish.
7. This completes the loading of the source schemas.

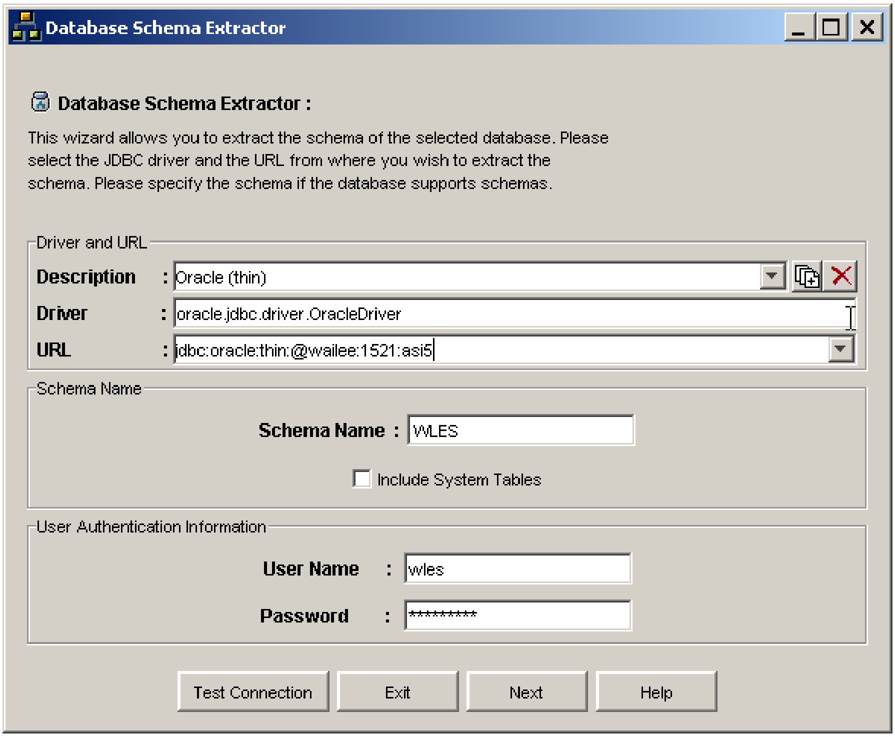
Extracting the Destination Schemas

This section describes how to extract the policy database destination schemas for the database server.

To extract the destination schemas from the database server, perform the following steps:

1. On the RadiantOne Synchronization Services Administration console, to start the Schema Extraction Wizard is displayed, select New from the Datastore drop-down menu.
2. Select Database Schema Extraction radio button and click Next. The Database Schema Extractor page is displayed (see Figure 7-10).

Figure 7-10 Database Schema Extractor Page



Database Schema Extractor

Database Schema Extractor :

This wizard allows you to extract the schema of the selected database. Please select the JDBC driver and the URL from where you wish to extract the schema. Please specify the schema if the database supports schemas.

Driver and URL

Description : Oracle (thin)

Driver : oracle.jdbc.driver.OracleDriver

URL : jdbc:oracle:thin:@wailee:1521:asi5

Schema Name

Schema Name : WLES

☐ Include System Tables

User Authentication Information

User Name : wles

Password : **********

3. Refer to Table 7-6 and set the database server parameters.

Table 7-6 Database Server Parameters

Parameters	Description
Description	The type of database used for the database server.
Driver	The database JDBC driver. **Note:** Better performance may be achieved by configuring a Type II JDBC driver. For Oracle (OCI), this is the same driver but uses a different URL syntax. Please refer to your Oracle documentation for the correct syntax and configuration.

Table 7-6 Database Server Parameters (Continued)

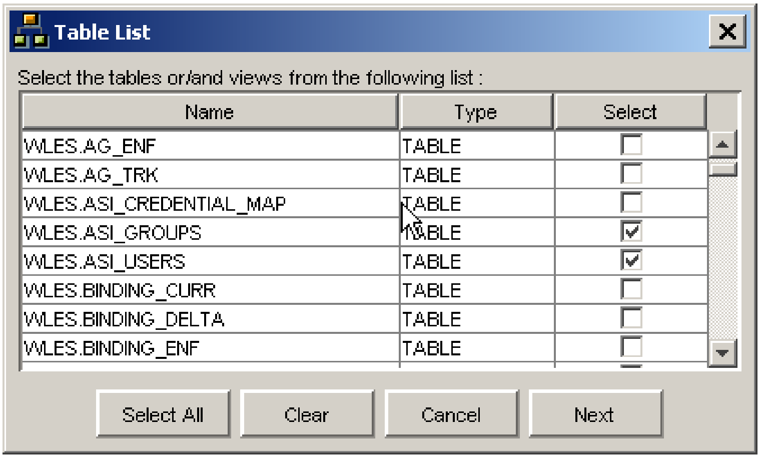
Parameters	Description
URL	The URL of the database server. You are only required to supply the name of the database host machine and database name, for example, jdbc:oracle:thin@asi_host:1521:asi5.
Schema Name	The schema name. This name must be unique in the database server, for example, `WLES`. **Note:** If you are using an Oracle database server, you must type the schema name in uppercase.
Include System Tables	Determines whether system files are included. Check this box to on.
User Name	The username you enter to access the database server.
Password	The password you enter to access the database server

4. To verify that the schema extractor can connect to the database server, click Test Connection. A "Connection succeeded" dialog box is displayed.

Note: If the connection fails, make sure that all the database server parameters are set correctly.

5. Click Ok and click Next. The Table List dialog box is displayed (see Figure 7-11).

Figure 7-11 Table List Dialog Box



Select the tables or/and views from the following list :

Name	Type	Select
WLES.AG_ENF	TABLE	☐
WLES.AG_TRK	TABLE	☐
WLES.ASI_CREDENTIAL_MAP	TABLE	☐
WLES.ASI_GROUPS	TABLE	☑
WLES.ASI_USERS	TABLE	☑
WLES.BINDING_CURR	TABLE	☐
WLES.BINDING_DELTA	TABLE	☐
WLES.BINDING_ENF	TABLE	☐

6. Select the `WLES.ASI_GROUPS` and `WLES.ASI_USERS` tables and click Next. The Save window is displayed.
7. Accept the default filename (the default filename should match the database name) and click Open. A “Schema Extraction Completed” message dialog box is displayed.

   Note: Be sure to save the filename in lowercase. Also, the filename cannot contain any periods (for example, this filename is correct: `asi5.orx`).
8. Click Ok and then click Exit. By default this schema is saved to `../RLI_HOME/data/org`.
9. This completes the extraction of the destination schemas.

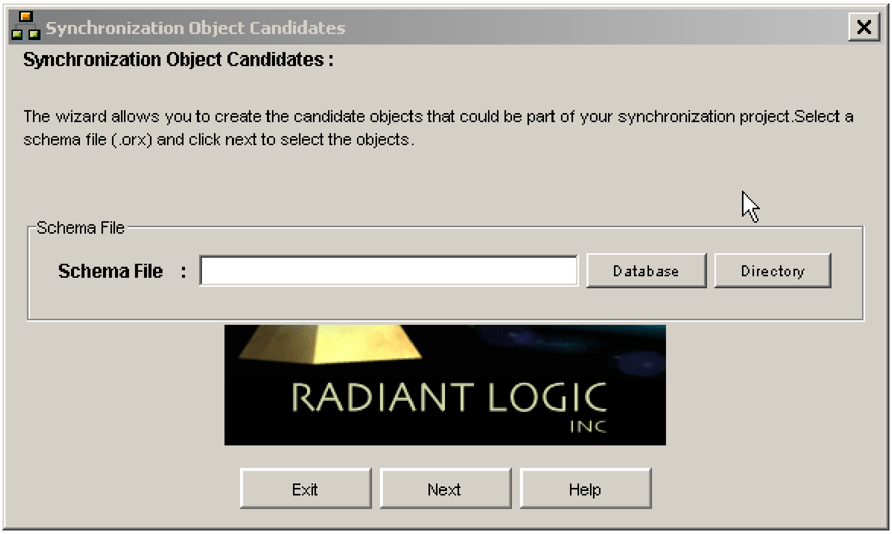
Loading the Destination Schemas

This section describes how to load the policy database destination schemas for the database server.

To load the destination schemas from the database server, perform the following steps:

1. On the RadiantOne Synchronization Services Administration console, from the Datastore drop-down menu, select Add. The Synchronization Object Candidates page is displayed (see Figure 7-12).

Figure 7-12 RadiantOne Synchronization Object Candidates Page



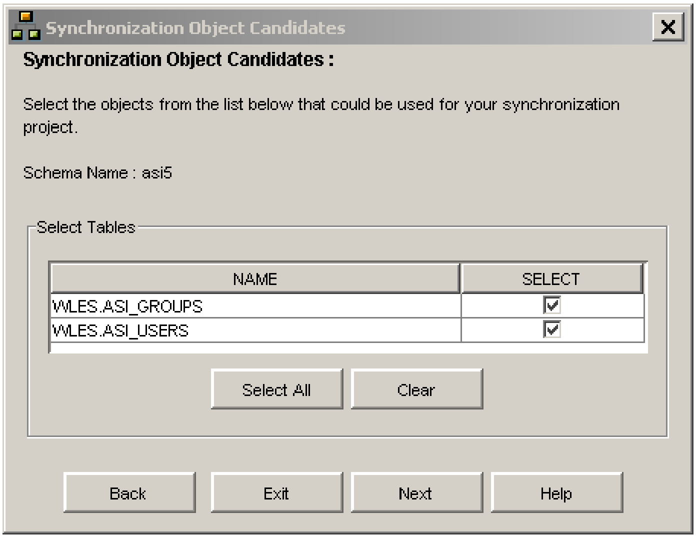
Synchronization Object Candidates

Synchronization Object Candidates :

The wizard allows you to create the candidate objects that could be part of your synchronization project.Select a schema file (.orx) and click next to select the objects.

2. Click Database. The Open window is displayed.
3. Select schema extraction file that you created in "Extracting the Destination Schemas" on page 7-18 (for example: `asi5.orx`) and click Open. The Schema File field is populated with the filename, including the path.
4. Click Next. The Synchronization Objects Candidates page is displayed (see Figure 7-13).

Figure 7-13 Synchronization Select Objects Page (Database Server Objects)



Synchronization Object Candidates :

Select the objects from the list below that could be used for your synchronization project.

Schema Name : asi5

Select Tables

NAME	SELECT
WLES.ASI_GROUPS	☑
WLES.ASI_USERS	☑

5. Click Select All and click Next. A “Successfully generated the datastore” message is displayed.
6. To exit, click Finish.
7. This completes the loading of the database server schemas for the policy database.

Configuring the Source-to-Destination Topology

This section describes how to configure the RadiantOne topology that serves to link the source user repository to the policy database. The topology shows all of the objects involved in the synchronization process and the data flow. You use the topology to define and connect all of the data objects that are involved in a particular synchronization process.

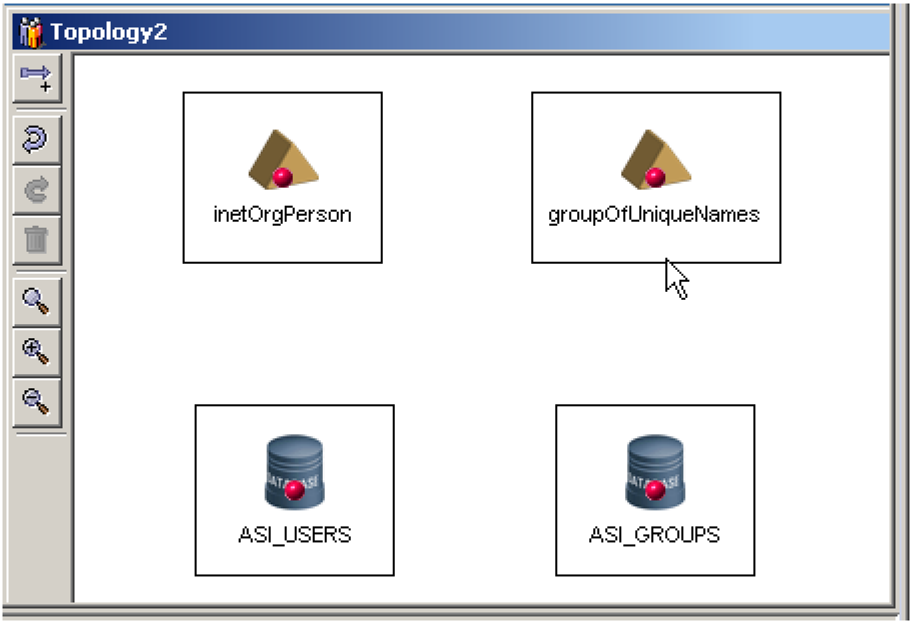
To configure the topology, perform the following steps:

1. On the RadiantOne Synchronization Services Administration console, from the RadiantOne Synchronization Services Topology drop-down menu, select New. The Topology window is displayed in the right pane.
2. Expand the nodes in the left pane, click the `groupOfUniqueNames` node and drag and drop it into the right pane.

3. Repeat step 3 for the `initOrgPerson`, `ASI_GROUPS`, and `ASI_USERS` nodes.

4. Click the red dot for each of the publishing objects (`groupOfUniqueNames` and `initOrgPerson`) and drag it to the red dot of the subscribing object. The tool draws lines connecting to the objects and labels them Transformation1 an Transformation2 (see Figure 7-14).

Figure 7-14 Topology Layout



Configuring the Topology Transformations

This section describes how to configure the RadiantOne topology transformation scripts. These scripts determine how the source data in the user repository is transformed before it is written to the policy database.

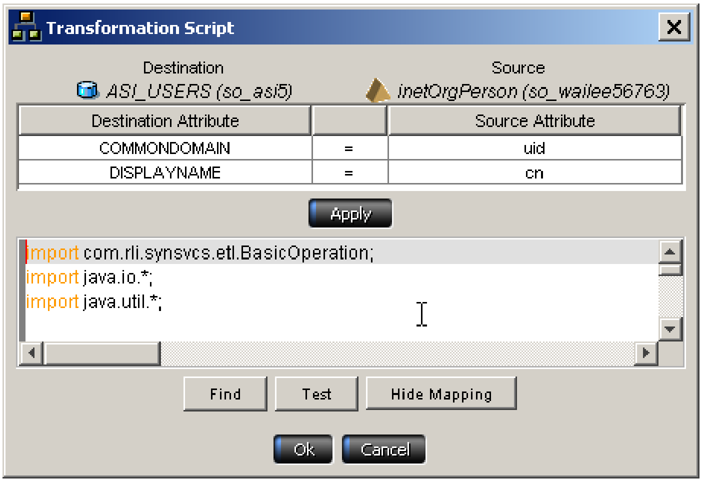
Note: The RadiantOne Synchronization Services tool is capable of very sophisticated transformations, including creating a unified identity from multiple sources. The transformation scripting language is Java. If you want to explore more sophisticated transformations or merging of identity data, refer to the RadiantOne documentation available in the RadiantOne installation directory.

ALES ships with sample user and group transformation scripts located in `BEA_Home\ales21-admin\examples\r1syncservice`. The file names are `asi_users.djava` and `asi_groups.djava`. The samples are dynamic java scripts that are compiled and run by the RadiantOne Synchronization Services as part of its transformation runtime. The following procedure uses the `asi_groups.djava` sample script.

To configure the topology transformation scripts, perform the following steps:

1. On the RadiantOne Synchronization Services Administration console, right click Transformation1 and select Edit Script. The Transformation Script window is displayed (see Figure 7-15).

Figure 7-15 Topology Script for Transformation1



Destination: *ASI_USERS (so_asi5)*

Source: *inetOrgPerson (so_wailee56763)*

Destination Attribute		Source Attribute
COMMONDOMAIN	=	uid
DISPLAYNAME	=	cn

```
import com.rli.synsvcs.etl.BasicOperation;
import java.io.*;
import java.util.*;
```

2. Set the `COMMONDOMAIN` and `DISPLAYNAME` attributes to `uid` and `cn` respectively as shown in Figure 7-15, click Apply, and click Ok.
3. Right click Transformation2 and select Edit Script. The Transformation Script window is displayed.
4. Position your cursor in the bottom region of the window, right click and select load. The Open window is displayed.
5. Locate the `asi_groups.djava` file in the `BEA_Home\ales21-admin\examples\r1syncservice` directory, select it and click Open, click Apply, and click Ok.

6. Click the Topology drop-down menu and click Save to save the topology.
7. This completes the configuration of the transformation topology.

UpLoading User and Group Data

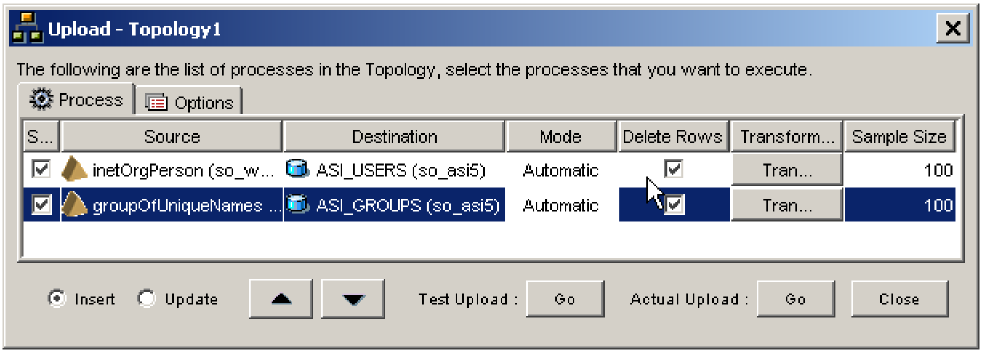
This section describes how to use the transformation topology to upload the user and group data from the source user repository to the policy database.

Note: This task serves to test all that all the configuration tasks you have performed up to this point have been performed correctly and that you can proceed to the next section, "Configuring Metadirectory Synchronization" on page 7-27, and perform the synchronization tasks. If the upload fails, check the previous tasks to ensure that they were performed correctly. Also, verify that you used the correct passwords in each of the previous configuration tasks.

To upload the user and group data, perform the following steps:

1. On the RadiantOne Synchronization Services Administration console, in the right pane, click the Deployment tab, click the folder icon, and open the topology that you just saved. The topology is displayed.
2. From the Deployment drop-down menu, select Upload. The upload topology page is displayed (see Figure 7-16).

Figure 7-16 Upload Topology Page

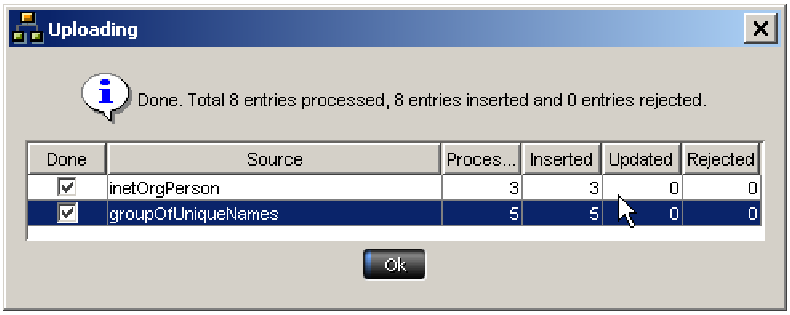


The following are the list of processes in the Topology, select the processes that you want to execute.

S...	Source	Destination	Mode	Delete Rows	Transform...	Sample Size
☑	inetOrgPerson (so_w...	ASI_USERS (so_asi5)	Automatic	☑	Tran...	100
☑	groupOfUniqueNames ...	ASI_GROUPS (so_asi5)	Automatic	☑	Tran...	100

3. Check on both Delete Rows check boxes and click Test Upload. The Uploading page is displayed and indicates that the upload is successful (see Figure 7-17).

Figure 7-17 Topology Uploading Page



Done. Total 8 entries processed, 8 entries inserted and 0 entries rejected.

Done	Source	Proces...	Inserted	Updated	Rejected
☑	inetOrgPerson	3	3	0	0
☑	groupOfUniqueNames	5	5	0	0

4. Click Ok
5. On the upload topology page, click Actual Upload (see Figure 7-16). The Uploading page is displayed again and indicates whether all entries were processed successfully.
6. Click Ok and on the Upload - Topology page, click close.
7. To verify that the upload actually moved user data into the designated ALES identity directory, perform the following steps:
    a. In the Administration Console's navigation tree, open the Identity folder.
    b. Click Groups and Users. The user data is displayed in the console.
8. This completes the user and group data upload.

Configuring Metadirectory Synchronization

This section describes how to configure the metadirectory components for automatic updates to the policy database whenever changes are made to the user repository.

The RadiantOne Synchronization Services provides connectors and a synchronization hub that work together to synchronize data between various data sources. Connectors interface with the data sources. Data flows to and from the connectors asynchronously in the form of XML messages. All messages flow through the synchronization hub, which is a server that transforms the messages and routes them to the connectors that are subscribed to the changes. The BEA WebLogic JMS messaging broker manages the topics and provides guaranteed message delivery.

The role of the connectors is two fold. First, the connectors capture changes in the data source, translate the changes into a common XML format, and send them to the synchronization hub through the messaging server. Secondly, the connectors receive XML messages, translate them, and apply the changes to the data source.

To configure metadirectory synchronization, perform the following tasks:

- "Configuring the Synchronization Hub" on page 7-28
- "Configuring the Directory Connector" on page 7-29
- "Configuring the Policy Database Connectors" on page 7-30
- "Starting the Synchronization Hub" on page 7-31
- "Starting the Source and Destination Connectors" on page 7-31

Configuring the Synchronization Hub

To configure the synchronization hub, set the parameters in `RadiantOne_Home\r1syncsvcs\rlicon.ini` to match the settings on the WebLogic Server. The required settings are shown in Listing 7-1. This information is used by the RadiantOne Synchronization Services tool to contact the JMS server running on the Administration Server.

Listing 7-1 Synchronization Hub Settings

```
...
[BEA]
JMS SERVER NAME=RLI_JMS_SERVER
JNDI CLASS NAME=weblogic.jndi.WLInitialContextFactory
CONNECTION FACTORY NAME=weblogic.asiAdminServer.jms.TopicConnectionFactory
URL=t3://localhost:7000
USER_NAME=system
USER_PASSWORD=weblogic
Msg Time-To-Live=3600000
RETRY PERIOD=12
MAXIMUM ATTEMPTS = 11
```

Configuring the Directory Connector

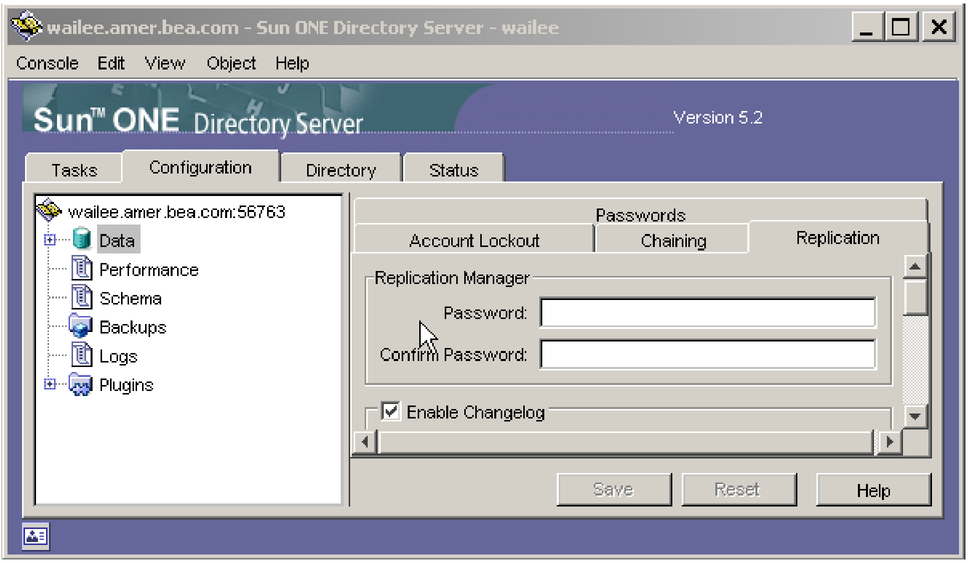
If you are using the Sun ONE Directory Server, you must configure the directory connector for so that changes to the user repository are automatically updated in the policy database.

Note: If you are using Active Directory, Windows NT, or other repository type, skip this section and go "Configuring the Policy Database Connectors" on page 7-30. For more information on LDAP connector configuration requirements and procedures, see the *RadiantOne Synchronization Services Guide* located in the RadiantOne installation directory.

To configure the directory connector, perform the following steps:

1. Click Start>Programs>Sun ONE Server Products>Sun ONE Server Console 5.2 and log into the Directory Server Console using the `admin` User ID.
2. In the left pane, expand the Domain and Server Group nodes, and double-click the Directory Server for your directory server. The Directory Server Tasks page is displayed.
3. Click the Configuration tab, select the Data node in the left pane, and select the Replication tab. The Replication page is displayed (see Figure 7-18).

Figure 7-18 Directory Server Replication Page



☑ Enable Changelog

4. Check the Enable Changelog check box and click Save.

5. In the left pane, open the Plugins folder, click Retro Changelog Plugin, check the Enable plug check box, and click Save.

Note: If you are using an LDAP server other than iPlanet or you do not want to use the iPlanet changelog, Radiant Synchronization Services also has a polling connector. For information on configuring the other LDAP connectors, refer to the RadiantOne online documentation located in the RadiantOne installation directory.

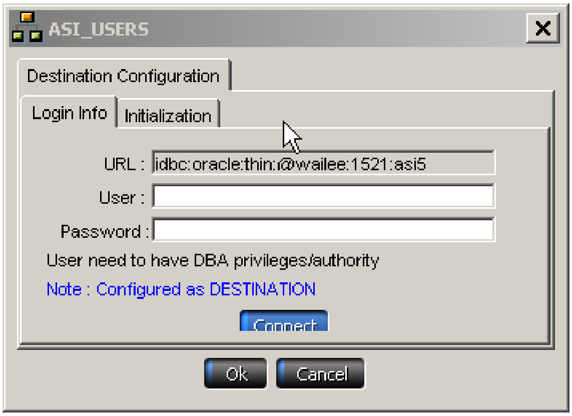
6. Click the Tasks tab and click Restart Directory Server to restart the server.
7. The completes configuration of the directory connector.

Configuring the Policy Database Connectors

To configure the policy database connector, perform the following steps:

1. On the RadiantOne Synchronization Services Administration console, click the topology tab and open the topology.
2. Right-click the ASI_USERS database object, and select Configure. The ASI_USERS Destination Configuration page is displayed (see Figure 7-19).

Figure 7-19 ASI_USERS Destination Configuration Page



URL : jdbc:oracle:thin:@wailee:1521:asi5

User :

Password :

User need to have DBA privileges/authority

Note : Configured as DESTINATION

Note: If the database object that the connector is listening to changes or is modified (for example, one of the data types changes or you add or remove columns), you can reconfigure the connector by right-clicking on the database object in the topology, and then choosing Configure. A note is displayed on the Login Info tab that specifies how the connector is configured. Enter the user and password information and, on the Initialization tab, reconfigure the connector by either Applying the Script or Saving and executing it later.

3. Enter the user and password to connect to the database as the database administrator and click Connect. A “Connection Successful” dialog is displayed. Click Ok. A script is generated to create the rli_con user, the needed log tables, and triggers.
4. Click the Initialization tab, select the Apply Now radio button to generate log tables and triggers for the ASI_USERS database object, and click Apply. A “Configuration completed” dialog box is displayed. Click Ok.
5. Click Ok.
6. Repeat steps 2 through 5 for the `ASI_GROUPS` database object.
7. This completes the configuration of the policy database connectors.

Starting the Synchronization Hub

To start the Synchronization Hub, perform the following steps:

1. On the RadiantOne Synchronization Services Administration console, click the Deployment drop-down menu, select Synchronization Hub, and click Start. The JMS Connection Username/Password dialog is displayed.
2. Enter the JMS connection username and password (if necessary) and click Ok. Use the same username/password that you used to log into the WebLogic Server Administration Console. A small window for the hub opens and indicates that the hub is running.

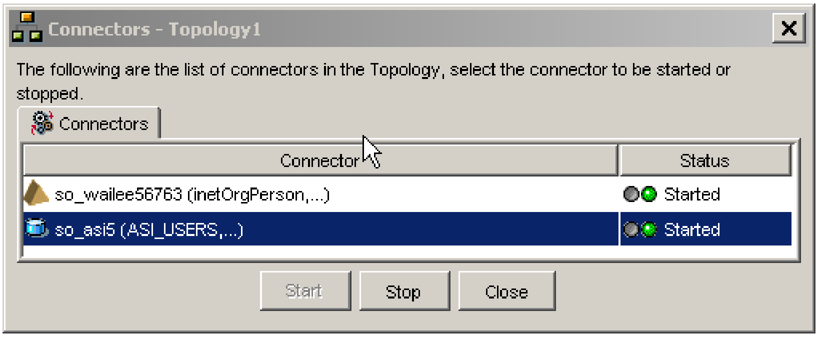
Note: You can also use the Start Hub icon on the Deployment Tab to start the hub.

Starting the Source and Destination Connectors

To start the source and destination connectors, make sure the Synchronization Hub is running. Then perform the following steps:

1. On the RadiantOne Synchronization Services Administration console, select the Deployment tab, click the folder icon, and select and open the Topology. The topology is displayed.
2. From the Deployment drop-down menu, select Connectors. The Connectors - Topology dialog is displayed (see Figure 7-20).

Figure 7-20 Connectors - Topology Page


Connectors - Topology1
The following are the list of connectors in the Topology, select the connector to be started or stopped.
Connectors
Connector
Status
so_wailee56763 (inetOrgPerson,...)
Started
so_asi5 (ASI_USERS,...)
Started
Start
Stop
Close

3. Select each connector and click Start. The connectors start. A small window for each connector opens and indicates that the connector is running.

Note: The connectors can also be started and stopped by clicking the Start Connector and Stop Connectors icons under the Deployment tab.

4. This completes configuration of metadirectory Synchronization.

Verifying that Metadirectory Synchronization Works

This section describes how to verify that metadirectory synchronization is properly configured such that changes to user and group entries in the user repository are reflected in the policy database.

To verify that metadirectory synchronization is properly configured, perform the following steps:

1. Use the user repository server to create a new user and add that user to the source group.
2. Open the Administration Console, open the Identity folder, the list of identity directories is displayed in the right pane.
3. Select the identity directory that you configured for automatic updates (for example, `ldapdir`), and click Users in the left pane. The new user is displayed in the list of users in the right pane.

CHAPTER 8

Authorization Caching

This section covers the following topics:

Understanding Authorization Caching

Authorization caching allows the ASI Authorization and ASI Role Mapper providers to cache the result of an authorization call, and use that result if future calls are made by the same caller. The cache match is based on a combination of the following:

- Subject
- Resource
- Privilege
- Roles
- Applicable Context (the portion used in making the original decision)

Additionally, the authorization cache automatically invalidates itself if there is a policy or user profile change.

Configuring Authorization Caching

Authorization caching is on by default. It may be configured from within the Administration Console through the ASI Authorization and ASI Role Mapper provider configuration. Table 8-1 lists the switches affect the authorization cache.

Table 8-1 Authorization Caching

Setting	Default Value	Description
AccessAllowedCaching	True	Enables/disables caching of authorization decisions.
GetRolesCaching	True	Enables/disables caching of role mapping decisions.
SessionExpiration	60	Specifies the number of seconds that authorization decisions for a user will be cached in memory. Upon expiration, the cached information is cleared and then updated if the user makes a subsequent request. While increasing this value can improve performance, it may also reduce security by making authorization decisions based on outdated information.
SubjectDataCacheExpiration	60	Defines how long user profile data will be cached. Cached authorization decisions are reset each time this data cache expires. You can increase this value to improve performance.

The properties listed in Table 8-2 can be entered as advanced configuration properties to further tune the cache.

Table 8-2 Advanced Configuration Properties

Setting	Default Value	Description
ASI.AuthorizationCacheLimit	1000	Determines the maximum number of cached decisions per user session. Once exceeded, old cached values are overwritten.

Table 8-2 Advanced Configuration Properties (Continued)

Setting	Default Value	Description
ASI.AuthorizationCacheDynamicAttribute Limit	10	Determines the maximum number of context attributes a decision may use and still be stored in the cache.
ASI.PolicyCacheInvalidatorPollingInterval	1000	Determines how often the cache checks for policy distributions. The value is in milliseconds

Authorization Caching Expiration Functions

There is a small subset of data that may change without the knowledge of the cache. This includes internally computed time values, as well as custom evaluation plug-ins. Because the cache is not aware of changes in these values, it does not automatically invalidate a cached decision when they change. For this reason a series of evaluation functions is provided to control the period of cache validity. These functions are only needed in policies that make explicit use of internally computed time values or custom evaluation plug-ins.

Table 8-3 lists the internally computed time values. If these values are referenced in a policy, you should also explicitly set the cache validity for the policy.

Table 8-3 Time Values Used with Expiration Functions

Credential	Value	Range or Format
time24	integer	0–2359
time24gmt	integer	0–2359
dayofweek	Dayofweek_type	Sunday–Saturday
dayofweekgmt	Dayofweek_type	Sunday–Saturday
dayofmonth	integer	1–31
dayofmonthgmt	integer	1–31
dayofyear	integer	1–366
dayofyeargmt	integer	1–366

Table 8-3 Time Values Used with Expiration Functions (Continued)

Credential	Value	Range or Format
daysinmonth	integer	28–31
daysinyear	integer	365–366
minute	integer	0–59
minutegmt	integer	0–59
month	month_type	January–December
monthgmt	month_type	January–December
year	integer	0–9999
yeargmt	integer	0–9999
timeofday	time	HH:MMAM” or “HH:MMPM”
timeofdaygmt	time	HH:MMAM” or “HH:MMPM”
hour	integer	0–23
hourgmt	integer	0–23
date	Date	MM/DD/YYYY”
dategmt	Date	MM/DD/YYYY”

Table 8-4 lists the expiration functions for the authorization cache. You can use these functions to set an expiration time for the decision. This way you can instruct the cache to only hold the value for a given period of time, or to not hold it at all. These functions correspond roughly to each of the internally computed time types.

Table 8-4 Expiration Functions

Function	Argument	Description
valid_for_mseconds	integer	Valid for a given number of milliseconds
valid_for_seconds	integer	Valid for a given number of seconds
valid_for_minutes	integer	Valid for a given number of minutes

Table 8-4 Expiration Functions (Continued)

Function	Argument	Description
valid_for_hours	integer	Valid for a given number of hours
valid_until_timeofday	time	Valid until the specified time on the date the evaluation is performed
valid_until_time24	integer	Valid until the specified time on the date the evaluation is performed
valid_until_hour	integer	Valid until the specified hour on the date the evaluation is performed
valid_until_minute	integer	Valid until the specified minute of the hour the evaluation is performed
valid_until_date	Date	Valid until the specified date
valid_until_year	integer	Valid until the specified year
valid_until_month	month_type	Valid until the specified month of the year the evaluation is performed
valid_until_dayofyear	integer	Valid until the specified day of the year the evaluation is performed
valid_until_dayofmonth	integer	Valid until the specified day of the month the evaluation is performed
valid_until_dayofweek	Dayofweek_ type	Valid until the specified day of the week the evaluation is performed
valid_until_timeofday_gmt	time	Valid until the specified time on the date the evaluation is performed in GMT time.
valid_until_time24_gmt	integer	Valid until the specified time on the date the evaluation is performed in GMT time.
valid_until_hour_gmt	integer	Valid until the specified minute of the hour the evaluation is performed in GMT time
valid_until_minute_gmt	integer	Valid until the specified minute of the hour the evaluation is performed in GMT time.
valid_until_date_gmt	Date	Valid until the specified date in GMT time.

Table 8-4 Expiration Functions (Continued)

Function	Argument	Description
valid_until_year_gmt	integer	Valid until the specified year in GMT time.
valid_until_month_gmt	month_type	Valid until the specified month of the year the evaluation is performed in GMT time.
valid_until_dayofyear_gmt	integer	Valid until the specified day of the year the evaluation is performed in GMT time.
valid_until_dayofmonth_gmt	integer	Valid until the specified day of the month the evaluation is performed in GMT time.
valid_until_dayofweek_gmt	Dayofweek_ type	Valid until the specified day of the week the evaluation is performed in GMT time.

For example, if you had the following policy:

```
GRANT(//priv/order,//app/resturant/breakfast,//sgrp/customers/allusers/
) if hour < 11;
```

When authorization caching is enabled, you write the policy as:

```
GRANT(//priv/order,//app/resturant/breakfast,//sgrp/customers/allusers/
) if hour < 11 and valid_until_hour(11);
```

With authorization caching, the result of this policy is cached in the provider until 11:00 AM, at which time, it expires. Not calling `valid_until_hour` argument results in this policy being cached until the next policy distribution. Therefore, if you are using authorization caching, it is important to update your time dependent policies appropriately.